



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

RATE CONTROL FOR OBJECT-BASED VIDEO CODING

Paulo Jorge Lourenço Nunes

(Mestre)

**DISSERTAÇÃO PARA OBTENÇÃO DO GRAU DE DOUTOR
EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES**

Orientador: Doutor Fernando Manuel Bernardo Pereira

Júri

Presidente: Reitor da Universidade Técnica de Lisboa

Vogais: Doutor Fernando Manuel Bernardo Pereira
Doutor Mário Alexandre Teles de Figueiredo
Doutor Leonel Augusto Pires Seabra de Sousa
Doutor Pedro António Amado de Assunção
Doutor António José Nunes Navarro Rodrigues

JULHO DE 2007

To my wife Dina and my daughter Carolina.

Acknowledgements

This Thesis would not have been possible without the support and friendship of many people. To all of them I am indebted and I would like to express my gratefulness here.

The first person that I would like thank is Prof. Fernando Pereira, supervisor of this Thesis. Without his permanent support and guidance throughout these years, this work would never have reached an end. It is an honor to have worked with such an outstanding professional and human being. I have to thank him also for his friendship.

I would also like to thank Luís Ducla Soares and Paulo Lobato Correia, for their helpful suggestions and support, and, not least, also for their friendship. My former colleagues João Valentim and Manuel Menezes de Sequeira deserve a particular mention here as well for their collaboration in the development of some software tools. Thanks also go to all my other colleagues of the Image Group of IST for such a friendly and cooperative working environment.

It was for me a privilege and a pleasure to be able to participate in various international projects that promoted my contact with the international research community from all over the world. I would like to thank, in particular, the participants of the MoMuSys ACTS project and the members of the MPEG Video group for the fruitful discussions and effective collaboration.

I would also like to express my thankfulness to Fundação para a Ciência e a Tecnologia for the PRAXIS XXI scholarship, a valuable financial support without which this work would not have been possible.

I could not finish these acknowledgments without mentioning my family and my friends. To them goes my great recognition for their support.

To my wife Dina and my daughter Carolina, to whom I dedicate this work, for their patience, understanding, and everlasting affection.

To my parents, Maria and Joaquim, for stressing my education since the very early days and for their valuable teachings.

To my uncle Manuel and my friend Tó, who are not any longer with us, for their testimony of life.

To Rui, my companion since I met myself, for all the endless talks and deep friendship.

To all my friends, who walk with me the road of life, simply for being there.

Abstract

This Thesis considers the novel problem of video coding rate control fostered by the object-based video coding architecture such as the one adopted by the MPEG-4 Visual standard, notably by developing methods for controlling a video encoder in order to produce a set of bitstreams which efficiently represent the scene in the sense that the quality of the video experience is maximized.

Because object-based video opens new dimensions and, consequently, new strategies to the rate control problem, a new framework for object-based video coding rate control is proposed in this Thesis, where this important function of any video encoder is performed by using two levels: i) scene-level rate control, responsible for allocating the available resources between the objects in the scene; and ii) object-level rate control, responsible for computing the best encoding parameters for each object.

Also a critical analysis of the MPEG-4 Video Buffering Verifier mechanism is provided in this Thesis, discussing its major features and drawbacks, notably in comparison with alternative solutions. As a consequence, a model for the integration of this mechanism into a generic video encoder rate controller in order to produce bitstreams that comply with a chosen MPEG-4 video profile@level is proposed.

Since a major goal of the rate control mechanism is to achieve an optimal rate-distortion trade-off given some pre-defined restrictions, rate-distortion modeling is also considered in this Thesis. In this field, a set of rate-distortion models for Intra and Inter coding are proposed, aiming at relating either the rate or the distortion with the quantization parameters in order to characterize the behavior of MPEG-4 video encoders, allowing to predict the encoder behavior before encoding the video data.

Finally, a robust and efficient rate-control algorithm for single and multiple video objects encoding is proposed in this Thesis. This algorithm provides adequate mechanisms to deal with deviations between theoretical models and actual coding results, notably: i) compensation mechanisms (e.g., rate control decisions and actions) that are able to track these deviations and compensate them in order to allow a stable and efficient operation of the encoder, and ii) adaptation mechanisms (e.g., estimation of model parameters) that are able to instantaneously represent the actual behavior of the encoder and its rate controller.

KEYWORDS

Object-based video coding; rate control; rate-distortion models; MPEG-4 video; compliant video encoding; video buffering verifier; profiles and levels.

Resumo

Esta Tese aborda o problema do controlo do débito binário em codificadores de vídeo com uma arquitectura de codificação de vídeo baseada em objectos como a adoptada pela norma MPEG-4 Visual. O objectivo central é, assim, desenvolver métodos para controlo de codificadores de vídeo deste tipo de modo a produzir fluxos binários que representem eficientemente a cena, maximizando a qualidade da experiência visual.

Uma vez que a codificação de vídeo baseada em objectos abre novas oportunidades e, consequentemente, novas estratégias no controlo de débito, propõe-se nesta Tese uma nova abordagem para este problema, em que esta importante função de qualquer codificador de vídeo é executada através de dois níveis: i) nível de cena, responsável por atribuir os recursos disponíveis aos vários objectos da cena, ii) nível de objecto, responsável por determinar os melhores parâmetros de codificação para cada objecto.

É também efectuada nesta Tese uma análise crítica do mecanismo de verificação de vídeo MPEG-4, discutindo as suas principais características e limitações, nomeadamente em comparação com soluções alternativas. Neste sentido, é proposto um modelo para a sua integração num mecanismo genérico de controlo de débito de modo a gerar fluxos binários em conformidade com um perfil e nível MPEG-4 Visual.

Uma vez que o principal objectivo do mecanismo do controle de débito é obter uma relação óptima débito-distorção, dadas algumas restrições, é também abordado nesta Tese a modelização débito-distorção. Nesta área, é proposto um conjunto de modelos para codificação Intra e Inter que relacionam o débito ou a distorção com os parâmetros de quantificação, caracterizando e permitindo prever antecipadamente o comportamento dos codificadores de vídeo MPEG-4 antes de codificar os dados de vídeo.

Finalmente, é proposto nesta Tese um algoritmo de controlo de débito, robusto e eficiente, para codificação de cenas com um ou vários objectos de vídeo. Este algoritmo fornece mecanismos adequados para lidar com os desvios entre os modelos teóricos e os resultados reais da codificação, nomeadamente: i) mecanismos de compensação (e.g., decisões e acções do controlador de débito) que permitem o seguimento e compensação destes desvios, originando um funcionamento estável e eficiente do codificador, e ii) mecanismos de adaptação (e.g., estimação de parâmetros dos modelos) que possibilitam a representação instantânea do comportamento do codificador e do seu controlador.

PALAVRAS-CHAVE

Codificação de vídeo baseada em objectos; controlo de débito; vídeo MPEG-4; codificação de vídeo conforme; mecanismo de verificação de vídeo; perfis e níveis.

Contents

Chapter 1	Introduction.....	1
1.1	Context and Motivation	1
1.2	Main Objectives of this Thesis	6
1.3	Summary of Original Contributions	7
1.4	Outline of this Thesis.....	10
Chapter 2	MPEG-4 Standard: An Overview	13
2.1	Introduction.....	13
2.2	MPEG-4 Context and Objectives	14
2.2.1	The Object-based Representation.....	14
2.2.2	Functionalities	17
2.2.3	Applications.....	19
2.2.4	Organization of the MPEG-4 Standard	21
2.3	MPEG-4 Visual Coding Architecture.....	23
2.3.1	Hierarchical Syntactic Structure.....	24
2.3.2	Video Object Coding Scheme	26
2.4	MPEG-4 Video Coding Tools	29
2.4.1	Shape Coding.....	29
2.4.2	Motion Estimation and Compensation	33
2.4.3	Texture Coding.....	37
2.4.4	Sprite Coding.....	39
2.4.5	Still Texture Coding	40
2.4.6	Error Resilience	40
2.4.7	Scalability	43
2.4.8	Reduced Resolution Video Coding	46

2.4.9	Interlaced Video Coding.....	46
2.4.10	Short Video Header	46
2.5	MPEG-4 Video Rate Control.....	47
2.5.1	Frame Rate Control	47
2.5.2	Multiple Video Object Rate Control	50
2.5.3	Macroblock Rate Control	54
2.6	MPEG-4 Profiling.....	56
2.6.1	Profiling Concepts	56
2.6.2	Version Management.....	58
2.6.3	Visual Object Types	59
2.6.4	Visual Profiles	62
2.6.5	Video Profile@Level Definitions.....	65
2.6.6	Performance Evaluation of Video Profiles	68
2.7	Final Remarks	69
Chapter 3	Object-based Video Coding Rate Control: A Review	71
3.1	Introduction.....	71
3.2	Video Coding Rate Control Basic Objectives	72
3.3	Rate Control Constraints.....	74
3.3.1	Delay Constraints	74
3.3.2	Channel Constraints.....	77
3.3.3	Complexity Constraints	80
3.4	Frame-based Rate Control: The YUV Dimensions	80
3.4.1	Rate Control Dimensions.....	80
3.4.2	Rate Control Strategies	81
3.4.3	Rate Control Architecture.....	82
3.5	Object-based Rate Control: The Semantic Dimension	85
3.5.1	Rate Control Dimensions.....	86
3.5.2	Rate Control Strategies	86
3.5.3	Rate Control Architecture.....	89
3.6	Review of Object-based Bit Rate Control Methods	90
3.6.1	Telenor SVO Rate Control	91
3.6.2	Sarnoff SVO and MVO Rate Control.....	92
3.6.3	Mitsubishi MVO Rate Control and Related Work	93
3.6.4	Sharp MB Rate Control	94
3.6.5	Universidad Politécnica de Madrid MVO Rate Control	95

3.6.6	University of California at Santa Barbara SVO and MVO Rate Control via ρ -Domain Source Modeling	100
3.6.7	University of Texas at Arlington SVO and MVO Rate Control	103
3.6.8	Other Related Work	109
3.7	Final Remarks	111
Chapter 4	MPEG-4 Video Buffering Verifier Mechanism: Analysis and Alternatives	113
4.1	Introduction	113
4.2	The MPEG-4 Video Buffering Verifier Mechanism	115
4.2.1	Video Rate Buffer Verifier (VBV) Definition	116
4.2.2	Video Complexity Verifier (VCV) Definition	123
4.2.3	Video Reference Memory Verifier (VMV) Definition	126
4.2.4	Interaction between the VBV, VCV, and VMV Models	129
4.3	MPEG-2 and H.263 Video Verification Mechanisms	130
4.3.1	MPEG-2 Video Buffering Verifier	130
4.3.2	H.263 Hypothetical Reference Decoder	133
4.4	MPEG-4 Video Buffering Verifier Integration Architecture	134
4.5	Analysis of the Video Reference Memory Verifier	136
4.5.1	Decoder Picture Memory Modeling	136
4.5.2	VMV Model Approaches	137
4.5.3	VMV Encoder Implementation	142
4.6	Analysis of the Video Complexity Verifier	143
4.6.1	Encoded Data Complexity Modeling	144
4.6.2	VCV Model Approaches	146
4.6.3	VCV Encoder Implementation	161
4.7	Analysis of the Video Rate Buffer Verifier	162
4.7.1	VBV Model Approaches	162
4.7.2	VBV Encoder Implementation	164
4.8	Final Remarks	165
Chapter 5	Rate-Distortion Modeling for Low-Delay Video Encoding	167
5.1	Introduction	167
5.2	Rate-Distortion Modeling	169
5.2.1	Fundamentals of Rate-Distortion Theory	170
5.2.2	Operational Rate-Distortion	174
5.2.3	The Constrained Bit Rate Control Problem	175
5.2.4	Rate-Distortion Modeling for DCT-based Video Coding	176

5.2.5	Review of Rate-Distortion Modeling	183
5.3	Proposal of Rate and Distortion Models for Low-Delay Video Encoding.....	194
5.3.1	Rate and Distortion Models for Intra Coding.....	200
5.3.2	Stationary Rate and Distortion Models for Inter Coding	222
5.3.3	Delta Rate and Distortion Models for Inter Coding	226
5.4	Final Remarks	233
Chapter 6	Rate Control Algorithm for Low-Delay Video Encoding	235
6.1	Introduction.....	235
6.2	Control Approaches for the Bit Rate Control Problem.....	236
6.2.1	Feedback versus Feedforward Control	236
6.2.2	Linear Feedback Control	238
6.2.3	Adaptive Control	239
6.3	Reviewing Major Rate Control Solutions from a Compensation and Adaptation Perspective	242
6.3.1	H.261 RM8 Rate Control.....	242
6.3.2	MPEG-4 VM4 Rate Control.....	244
6.3.3	MPEG-2 Video TM5 Rate Control	246
6.3.4	MPEG-4 Visual Annex L Rate Control.....	251
6.4	Proposal for a Low-Delay Rate Control Algorithm.....	258
6.4.1	Application Scenario Requirements	259
6.4.2	Scene Analysis for Resource Allocation	262
6.4.3	Spatio-Temporal Resolution Control.....	263
6.4.4	Rate-Distortion Modeling.....	264
6.4.5	Bit Allocation	271
6.4.6	Video Buffering Verifier Control	283
6.4.7	Coding Mode Control.....	289
6.4.8	Summary of the Proposed Rate Control Algorithm	292
6.5	Scene-level and Object-level Rate Control Breakdown	295
6.5.1	Scene-level Rate Control	295
6.5.2	Object-level Rate Control.....	298
6.6	Quality Control in the Proposed Rate Control Algorithm	299
6.6.1	Temporal Inter SP Quality Control	300
6.6.2	Spatial Intra SP Quality Control.....	300
6.6.3	Spatial Intra VOP Quality control	300
6.7	Test Conditions and Performance Analysis.....	301

6.7.1	Single Video Object Performance Analysis	301
6.7.2	Multiple Video Objects Performance Analysis	315
6.8	Final Remarks	333
Chapter 7	Achievements and Future Directions	337
7.1	Achievements.....	337
7.2	Future Directions	338
Annex A	Additional Rate and Distortion Modeling Results for Intra Coding ..	343
A.1	Rate-Quantization Model Parameters	344
A.2	Rate-Quantization Model Parameters with a Reduced Number of Model Parameters.....	348
A.3	Distortion-Quantization Model Parameters	352
A.4	Rate-Distortion Model Parameters	355
References	361

List of Figures

Figure 1.1 – Evolution of the compression ratio for the main international video coding standards.....	4
Figure 1.2 – News video sequence (top) and its corresponding video objects (bottom): Background, Dancers, Speakers, and Logo	5
Figure 2.1 – MPEG-4 object-based representation architecture [51].....	16
Figure 2.2 – Hierarchical structure of MPEG-4 video bitstreams.....	25
Figure 2.3 – Example video object obtained through segmentation: a) sample image of the Stefan video sequence; b) video object (Player); c) shape information.....	25
Figure 2.4 – Example video object obtained through chroma-keying: a) sample image of kids shot in front of a green screen; b) video object (Kids); c) shape information.....	26
Figure 2.5 – MB types within the VOP bounding box.....	27
Figure 2.6 – MPEG-4 VOP coding modes: Intra (I), Predicted (P), and Bidirectional (B)	27
Figure 2.7 – MPEG-4 VOP encoder block diagram	29
Figure 2.8 – Template for computing the BAB type context for I-VOPs	31
Figure 2.9 – CAE templates for context computation (‘?’ represents the pixel being encoded): a) Intra-coded BABs; b) Inter-coded BABs.....	33
Figure 2.10 – Sample interpolation for half-pixel MVs.....	34
Figure 2.11 – Motion vector prediction for the 1MV mode.....	35
Figure 2.12 – Motion vector prediction for the 4MV mode.....	36
Figure 2.13 – Padding process for motion estimation and compensation.....	36
Figure 2.14 – Global motion compensation	37
Figure 2.15 – Reconstruction of a scene using its background sprite [51]	39
Figure 2.16 – Decoder resynchronization following error detection [51].....	41
Figure 2.17 – Data recovery with RVLC [51].....	42
Figure 2.18 – MPEG-4 video packet with header extension code	43

Figure 2.19 – MPEG-4 temporal scalability	44
Figure 2.20 – MPEG-4 spatial scalability	45
Figure 2.21 – Basic FGS decoder structure [51]	45
Figure 2.22 – Relation between MPEG-4 versions [54]	59
Figure 3.1 – Frame-based rate control architecture.....	83
Figure 3.2 – Object-based rate control levels.....	87
Figure 3.3 – Object-based rate control framework.....	90
Figure 4.1 – Dynamics of the VBV occupancy for one VOL.....	119
Figure 4.2 – B-VOP and corresponding forward and backward predictions	120
Figure 4.3 – Dynamics of the VCV occupancy.....	126
Figure 4.4 – Dynamics of the VMV occupancy.....	128
Figure 4.5 – Relation between VCV and VMV occupancies.....	128
Figure 4.6 – Integration of the video buffering verifier mechanism in a MPEG-4 video encoder.....	135
Figure 4.7 – Relation between the VOP decoding, composition, and release times for the VMV models under study, when no B-VOPs are used.....	139
Figure 4.8 – VMV occupancy for the MPEG-4 and decoding memory approaches for a scene with 1 VO, when B-VOPs are not used	139
Figure 4.9 – VMV occupancy for the MPEG-4 and decoding memory approaches for a scene with 2 VOs, when B-VOPs are not used	140
Figure 4.10 – Relation between the VOP decoding, composition, and release times for the VMV models under study when B-VOPs are used.....	141
Figure 4.11 – VMV occupancy for the VMV models under study for a scene with 1 VO, when B-VOPs are used	141
Figure 4.12 – Decoder memory allocation estimation for the MPEG-4 (I) and the decoding memory (II) VMV models	143
Figure 4.13 – Decoding complexity evaluation approaches for MPEG-4 video	145
Figure 4.14 – Container sequence: object mainly composed by transparent MBs (82 %).....	150
Figure 4.15 – News sequence: MBs inside the rectangles are counted three times (twice as transparent) for the VCV and VMV buffers.....	150
Figure 4.16 – VCV and VMV occupancies for Stefan: (top) CP@L1; (bottom) CP@L2.....	152
Figure 4.17 – VCV and VMV occupancies for Children: (top) CP@L1; (bottom) CP@L2	153
Figure 4.18 – VCV and VMV occupancies for Coastguard: (top) CP@L1; (bottom) CP@L2	153
Figure 4.19 – VCV and VMV occupancies for News: (top) CP@L1; (bottom) CP@L2	154
Figure 4.20 – VCV and VMV occupancies for Container: CP@L2.....	154

Figure 4.21 – MPEG-4 and IST VCV and VMV occupancies for the Simple Profile: a) Akiyo SP@L1 at 64 kbps; b) Stefan SP@L3 at 384 kbps.....	158
Figure 4.22 – MPEG-4 and IST VCV and VMV occupancies for: a) News CP@L1 at 384 kbps; b) Coastguard CP@L1 at 384 kbps; c) News CP@L2 at 2000 kbps.....	160
Figure 4.23 – Number of MBs per shape type for the News sequence.....	161
Figure 4.24 – Independent VBV buffer control with fixed bandwidth allocation	163
Figure 4.25 – Independent VBV buffer control with dynamic bandwidth allocation.....	163
Figure 4.26 – Combined VBV buffer control with shared bandwidth resources.....	164
Figure 5.1 – Typical rate-distortion function for a discrete source	170
Figure 5.2 – Rate-distortion encoder and encoder model [151].....	170
Figure 5.3 – Rate-distortion plots: a) computation of the ORDF from the set of admissible rate-distortion points; b) comparison of coding schemes using the ORDF	174
Figure 5.4 – Example of a convex function	175
Figure 5.5 – Prediction error quantization.....	178
Figure 5.6 – Experimental and model rate-quantization characteristics for a frame of the Foreman sequence: a) Intra-coded; b) and Inter-coded	182
Figure 5.7 – Approximation of the logarithmic function	182
Figure 5.8 – Quadratic rate-distortion models for a frame of the Foreman sequence: a) Intra-coded; b) Inter-coded	187
Figure 5.9 – Estimation of the quadratic and hyperbolic rate-distortion models for a frame of the Foreman sequence using a sub-set of the experimental data points ($8 \leq Q \leq 24$): a) Intra-coded; b) Inter-coded.....	188
Figure 5.10 – Estimation of the simplified MB quadratic rate-distortion model for a frame of the Foreman sequence: a) Intra-coded; b) Inter-coded.....	189
Figure 5.11 – Intra and Inter rate and distortion functions for a frame of the Foreman and Stefan sequences: a) Foreman rate-quantization; b) Foreman distortion-quantization; c) Foreman rate-distortion; d) Stefan rate-quantization; e) Stefan distortion-quantization; f) Stefan rate-distortion ..	197
Figure 5.12 – Rate components for the Foreman and one arbitrarily shaped VO of the Stefan sequences encoded in Intra and Inter mode: a) Foreman Intra; b) Foreman Inter; c) Stefan VO 1 Intra; d) Stefan VO 1 Inter	198
Figure 5.13 – First frame of each test sequence: a) Foreman; b) Stefan; c) News; d) Kayak; e) Mother and Daughter (M&D); f) Football	199
Figure 5.14 – Experimental distortion-quantization function and model approximation for Intra coding: a) Foreman sequence; b) Stefan sequence	211
Figure 5.15 – Experimental rate-distortion function for Intra coding: a) Foreman sequence; b) Stefan sequence.....	216
Figure 5.16 – Dependency of the distortion-quantization on the reference quantization parameter for Inter coding	225

Figure 5.17 – Piecewise approximation of the distortion-quantization function	225
Figure 5.18 – Modification of the rate-quantization function when $Q_{ref} \neq Q_0$	228
Figure 5.19 – Delta rate and distortion functions for the Foreman sequence: (a) $R(Q, Q_{ref})$; (b) $\Delta R(Q)$ for $\Delta Q_{ref} = 4$; (c) $\Delta R(Q)$ for $\Delta Q_{ref} = 8$; (d) $D(Q, Q_{ref})$; (e) $\Delta D(Q)$ for $\Delta Q_{ref} = 4$; (f) $\Delta D(Q)$ for $\Delta Q_{ref} = 8$	230
Figure 6.1 – Generic, linear feedback control model	238
Figure 6.2 – Block diagram of a system with gain scheduling [172]	240
Figure 6.3 – Block diagram of a model-reference adaptive system [172]	240
Figure 6.4 – Block diagram of a self-tuning regulator [172]	241
Figure 6.5 – Architecture of the proposed rate control algorithm	259
Figure 6.6 – Early decoding start leading to VBV underflows	261
Figure 6.7 – VOP-level encoder model	265
Figure 6.8 – Evolution of parameter a for the Kayak sequence [QCIF@15Hz; 160 kbit/s]	269
Figure 6.9 – Estimation of parameter a for two VOPs of the Kayak sequence [QCIF@15Hz; 160 kbit/s]: a) VOP 60 Intra-coded; b) VOP 100 Inter-coded	270
Figure 6.10 – Multiple video objects encoding with different VOP rates	272
Figure 6.11 – Evolution of α_l along the sequence with and without distortion correction: a) Foreman; b) Stefan	277
Figure 6.12 – PSNR for various VOP coding type weight adaptations for three GOS of the Football sequence	277
Figure 6.13 – VO complexity weight estimation for the Stefan sequence: a) VO complexity weights; b) VO VOP PSNR; c) Scene PSNR	280
Figure 6.14 – Number of texture bits per MB as a function of the MB MAD for the Kayak sequence encoded with SP@L2: a) Intra-coded MBs; b) Inter-coded MBs	282
Figure 6.15 – Number of texture bits per MB as a function of the MB MAD for the Stefan sequence encoded with SP@L2: a) Intra-coded MBs; b) Inter-coded MBs	282
Figure 6.16 – Target and actual VBV buffer occupancy for each SP after VOP removing: a) SVO Stefan sequence; b) News sequence with 4 VOs encoded at different frames rates	285
Figure 6.17 – VBV buffer occupancy deviation leading to imminent VBV underflow ..	286
Figure 6.18 – Control limits to prevent violation of the VBV buffer	287
Figure 6.19 – Sample frames for the test sequences used for SVO encoding: a) Football; b) Kayak; c) Stefan; d) Foreman; e) Mother & Daughter; f) News	301
Figure 6.20 – Football SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation...	304

Figure 6.21 – Football SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation.	304
Figure 6.22 – VOP luminance PSNR for the Football sequence encoded at 192 kbit/s: a) Intra period 1s; b) Intra period 10s	305
Figure 6.23 – Kayak SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation.....	306
Figure 6.24 – Kayak SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation....	306
Figure 6.25 – VOP luminance PSNR for the Kayak sequence encoded at 1024 kbit/s: a) Intra period 1s; b) Intra period 10s	306
Figure 6.26 – Stefan SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation.....	307
Figure 6.27 – Stefan SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation	308
Figure 6.28 – VOP luminance PSNR for the Stefan sequence encoded at 768 kbit/s: a) Intra period 1s; b) Intra period 10s	308
Figure 6.29 – Foreman SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation..	309
Figure 6.30 – Foreman SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation	309
Figure 6.31 – VOP luminance PSNR for the Foreman sequence encoded at 192 kbit/s: a) Intra period 1s; b) Intra period 10s	310
Figure 6.32 – Mother & Daughter SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation	311
Figure 6.33 – Mother & Daughter SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation	311
Figure 6.34 – VOP luminance PSNR for the Mother & Daughter sequence encoded at 512 kbit/s: a) Intra period 1s; b) Intra period 10s	311
Figure 6.35 – News SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation	313
Figure 6.36 – News SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation	313
Figure 6.37 – VOP luminance PSNR for the News sequence encoded at 512 kbit/s: a) Intra period 1s; b) Intra period 10s	313
Figure 6.38 – MVO Stefan sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Player)	316
Figure 6.39 – MVO Coastguard sequence (frame 100): a) Composed Scene; b) VO 0 (Water); c) VO 1 (Large Boat); d) VO 2 (Small Boat); e) VO 3 (Shore)..	316
Figure 6.40 – MVO Bream sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Fish).....	316
Figure 6.41 – MVO News sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Dancers); d) VO 2 (Speakers); e) VO 3 (Logo) .	317
Figure 6.42 – Stefan MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation	321
Figure 6.43 – Stefan MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation	321
Figure 6.44 – Stefan MVO Scene PSNR (Intra period 1s): a) QCIF@7.5Hz 128 kbit/s; b) QCIF@15Hz 256 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@30Hz 768 kbit/s.....	322

Figure 6.45 – Stefan MVO Scene PSNR (Intra period 10s): a) QCIF@7.5Hz 128 kbit/s; b) QCIF@15Hz 256 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@30Hz 768 kbit/s.....	322
Figure 6.46 – Coastguard MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation	324
Figure 6.47 – Coastguard MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation	324
Figure 6.48 – Coastguard MVO Scene PSNR (Intra period 1s): a) QCIF@7.5Hz 96 kbits/s; b) QCIF@15Hz 256 kbits/s.....	324
Figure 6.49 – Coastguard MVO Scene and VOs PSNR (Intra period 1s): a) Scene PSNR; b) VO 0 PSNR; c) VO 1 PSNR; d) VO 2 PSNR; e) VO 3 PSNR.	325
Figure 6.50 – Bream MVO (Intra period 1s) ($\gamma_D = 0.2$): a) Average Scene PSNR; b) Scene PSNR Variation.....	327
Figure 6.51 – Bream MVO (Intra period 10s) ($\gamma_D = 0.2$): a) Average Scene PSNR; b) Scene PSNR Variation.....	327
Figure 6.52 – Bream MVO (Intra period 10s) ($\gamma_T = 0.5$): a) Scene Average PSNR; b) Scene PSNR Variation.....	327
Figure 6.53 – Scene and VOs PSNR for the Bream sequence: a) $\gamma_D = 0.2$; b) $\gamma_D = 0.5$	328
Figure 6.54 – News MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation	330
Figure 6.55 – News MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation	330
Figure 6.56 – News MVO Scene PSNR (Intra period 1s): a) CIF@15Hz 256 kbit/s; b) CIF@15Hz 320 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@15Hz 448 kbit/s.....	331
Figure 6.57 – News MVO Scene PSNR (Intra period 10s): a) QCIF@7.5Hz 64 kbit/s; b) QCIF@15Hz 128 kbit/s; c) CIF@15Hz 256 kbit/s; d) CIF@30Hz 640 kbit/s.....	331

List of Tables

Table 1.1 –	Digital video formats for different applications [5]	3
Table 2.1 –	List of BAB types for CAE.....	30
Table 2.2 –	List of BAB sizes for CAE	32
Table 2.3 –	Quantization step for the Intra DC coefficients	38
Table 2.4 –	Visual tools versus visual object types [51].....	61
Table 2.5 –	Visual profiles versus visual object types [51]	65
Table 2.6 –	Levels for video profiles [51]	67
Table 4.1 –	Values of k for the allowed picture formats.....	123
Table 4.2 –	VOP Size in MB units.....	140
Table 4.3 –	MB decoding complexity classes and relative complexity weights [21]...	151
Table 4.4 –	Test sequences used for each Profile@Level	152
Table 4.5 –	VMV and VCV buffer sizes and decoding rates for the profile@levels used [19].....	152
Table 4.6 –	Relation between the VCV decoder rate and VCV buffer size for the Simple Profile @ Level 1 and Simple Profile @ Level 3.....	157
Table 4.7 –	Comparison between the MPEG-4 and IST VCV models: MB decoding complexity classes and relative complexity weights for the Core Profile.	159
Table 4.8 –	Relation between the VCV decoder rate and VCV buffer size for the Core Profile @ Level 1 and Core Profile @ Level 3.....	160
Table 5.1 –	Rate-quantization model parameters for the Foreman sequence [QCIF] ..	202
Table 5.2 –	Rate-quantization model parameters for the Foreman sequence [CIF]	202
Table 5.3 –	Rate-quantization model parameters for the Stefan sequence [QCIF]	203
Table 5.4 –	Rate-quantization model parameters for the Stefan sequence [CIF]	203
Table 5.5 –	Rate-quantization average model fitting error results.....	204
Table 5.6 –	Rate-quantization model fitting error deviation results	204
Table 5.7 –	Rate-quantization model estimation complexity results ($\varepsilon = 10^{-3}$)	205

Table 5.8 –	Rate-quantization model estimation complexity results ($\varepsilon = 10^{-1}$).....	205
Table 5.9 –	Rate-quantization model parameters for the Foreman sequence [QCIF] with a reduced number of model parameters.....	207
Table 5.10 –	Rate-quantization model parameters for the Foreman sequence [CIF] with a reduced number of model parameters.....	207
Table 5.11 –	Rate-quantization model parameters for the Stefan sequence [QCIF] with a reduced number of model parameters.....	208
Table 5.12 –	Rate-quantization model parameters for the Stefan sequence [CIF] with a reduced number of model parameters.....	208
Table 5.13 –	Rate-quantization average model fitting error results with a reduced number of model parameters.....	209
Table 5.14 –	Rate-quantization model fitting standard deviation error results with a reduced number of model parameters.....	209
Table 5.15 –	Distortion-quantization model parameters for the Foreman sequence [QCIF].....	212
Table 5.16 –	Distortion-quantization model parameters for the Foreman sequence [CIF].....	212
Table 5.17 –	Distortion-quantization model parameters for the Stefan sequence [QCIF].....	213
Table 5.18 –	Distortion-quantization model parameters for the Stefan sequence [CIF]	213
Table 5.19 –	Distortion-quantization average model fitting error results.....	214
Table 5.20 –	Distortion-quantization model fitting standard deviation error results.....	214
Table 5.21 –	Distortion-quantization model estimation complexity results.....	215
Table 5.22 –	Distortion-quantization average model fitting error results with a reduced number of model parameters.....	215
Table 5.23 –	Rate-distortion model parameters for the Foreman sequence [QCIF].....	218
Table 5.24 –	Rate-distortion model parameters for the Foreman sequence [CIF].....	218
Table 5.25 –	Rate-distortion model parameters for the Stefan sequence [QCIF].....	219
Table 5.26 –	Rate-distortion model parameters for the Stefan sequence [CIF].....	219
Table 5.27 –	Rate-distortion average model fitting error results.....	220
Table 5.28 –	Rate-distortion model fitting standard deviation error results.....	220
Table 5.29 –	Rate-distortion model estimation complexity results.....	221
Table 5.30 –	Rate-distortion average model fitting error results with a reduced number of model parameters.....	221
Table 5.31 –	Stationary rate-quantization average model fitting error results.....	223
Table 5.32 –	Stationary distortion-quantization average model fitting error results.....	224
Table 5.33 –	Stationary rate-distortion average model fitting error results.....	226
Table 5.34 –	Delta rate-quantization average model fitting error results.....	232

Table 5.35 –	Delta distortion-quantization average model fitting error results	233
Table 6.1 –	Feedback versus feedforward rate control	237
Table 6.2 –	SVO spatio-temporal resolutions and target bit rates for the high-motion test sequences: Football, Kayak, and Stefan	302
Table 6.3 –	SVO spatio-temporal resolutions and target bit rates for the low-motion test sequences: Foreman, Mother & Daughter, and News	302
Table 6.4 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Football sequence	304
Table 6.5 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Kayak sequence	305
Table 6.6 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Stefan sequence	307
Table 6.7 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Foreman sequence	309
Table 6.8 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Mother & Daughter sequence	310
Table 6.9 –	SVO average PSNR and bit rate gains of the proposed rate control algorithm for the News sequence	312
Table 6.10 –	SVO average PSNR and bit rate gains for QCIF@7.5Hz	314
Table 6.11 –	SVO average PSNR and bit rate gains for QCIF@15Hz	314
Table 6.12 –	SVO average PSNR and bit rate gains for CIF@15Hz	315
Table 6.13 –	SVO average PSNR and bit rate gains for CIF@30Hz	315
Table 6.14 –	MVO spatio-temporal resolutions and target bit rates for the high-motion test sequences: Stefan and Coastguard	318
Table 6.15 –	MVO spatio-temporal resolutions and target bit rates for the low-motion test sequences: Bream and News	318
Table 6.16 –	MVO average PSNR and bit rate gains of the proposed rate control algorithm for the Stefan sequence	320
Table 6.17 –	MVO average PSNR and bit rate gains of the proposed rate control algorithm for the Coastguard sequence	323
Table 6.18 –	MVO average PSNR and bit rate gains of the proposed rate control algorithm for the Bream sequence	326
Table 6.19 –	MVO average PSNR and bit rate gains of the proposed rate control algorithm for the News sequence	329
Table 6.20 –	MVO average PSNR and bit rate gains for QCIF@7.5Hz	332
Table 6.21 –	MVO average PSNR and bit rate gains for QCIF@15Hz	332
Table 6.22 –	MVO average PSNR and bit rate gains for CIF@15Hz	333
Table 6.23 –	MVO average PSNR and bit rate gains for CIF@30Hz	333
Table A.1 –	Rate-quantization model parameters for the News sequence [QCIF]	344

Table A.2 –	Rate-quantization model parameters for the News sequence [CIF]	344
Table A.3 –	Rate-quantization model parameters for the Kayak sequence [QCIF]	345
Table A.4 –	Rate-quantization model parameters for the Kayak sequence [CIF]	345
Table A.5 –	Rate-quantization model parameters for the M&D sequence [QCIF]	346
Table A.6 –	Rate-quantization model parameters for the M&D sequence [CIF]	346
Table A.7 –	Rate-quantization model parameters for the Football sequence [QCIF] ...	347
Table A.8 –	Rate-quantization model parameters for the Football sequence [CIF]	347
Table A.9 –	Rate-quantization model parameters for the News sequence [QCIF] with a reduced number of model parameters	348
Table A.10 –	Rate-quantization model parameters for the News sequence [CIF] with a reduced number of model parameters	348
Table A.11 –	Rate-quantization model parameters for the Kayak sequence [QCIF] with a reduced number of model parameters	349
Table A.12 –	Rate-quantization model parameters for the Kayak sequence [CIF] with a reduced number of model parameters	349
Table A.13 –	Rate-quantization model parameters for the M&D sequence [QCIF] with a reduced number of model parameters	350
Table A.14 –	Rate-quantization model parameters for the M&D sequence [CIF] with a reduced number of model parameters	350
Table A.15 –	Rate-quantization model parameters for the Football sequence [QCIF] with a reduced number of model parameters	351
Table A.16 –	Rate-quantization model parameters for the Football sequence [CIF] with a reduced number of model parameters	351
Table A.17 –	Distortion-quantization model parameters for the News sequence [QCIF]	352
Table A.18 –	Distortion-quantization model parameters for the News sequence [CIF] .	352
Table A.19 –	Distortion-quantization model parameters for the Kayak sequence [QCIF]	353
Table A.20 –	Distortion-quantization model parameters for the Kayak sequence [CIF]	353
Table A.21 –	Distortion-quantization model parameters for the M&D sequence [QCIF]	354
Table A.22 –	Distortion-quantization model parameters for the M&D sequence [CIF] .	354
Table A.23 –	Distortion-quantization model parameters for the Football sequence [QCIF]	355
Table A.24 –	Distortion-quantization model parameters for the Football sequence [CIF]	355
Table A.25 –	Rate-distortion model parameters for the News sequence [QCIF]	356
Table A.26 –	Rate-distortion model parameters for the News sequence [CIF]	356
Table A.27 –	Rate-distortion model parameters for the Kayak sequence [QCIF]	357
Table A.28 –	Rate-distortion model parameters for the Kayak sequence [CIF]	357

Table A.29 – Rate-distortion model parameters for the M&D sequence [QCIF]	358
Table A.30 – Rate-distortion model parameters for the M&D sequence [CIF]	358
Table A.31 – Rate-distortion model parameters for the Football sequence [QCIF].....	359
Table A.32 – Rate-distortion model parameters for the Football sequence [CIF].....	359

List of Acronyms

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
3GPP	Third Generation Partnership Project
ACE	Advanced Coding Efficiency
AFX	Animation Framework Extension
API	Application Programming Interface
ARTS	Advanced Real-Time Simple
ATM	Asynchronous Transfer Mode
AV	Audiovisual
AVC	Advanced Video Coding
BAB	Binary Alpha Block
BAC	Binary Arithmetic Coding
BB	Bounding Box
BIFS	Binary Format for Scenes
CAE	Content-based Arithmetic Encoding
CD	Compact Disc
CIF	Common Intermediate Format
CODEC	Coder/Decoder
CR	Conversion Ratio
DAI	Delivery Multimedia Integration Framework Application Interface
DCT	Discrete Cosine Transform
DMIF	Delivery Multimedia Integration Framework
DP	Data Partitioning
DP	Dynamic Programming
DRC	Dynamic Resolution Conversion
DVB	Digital Video Broadcasting
DVD	Digital Versatile Disc
DWT	Discrete Wavelet Transform
ES	Elementary Stream
FBA	Face and Body Animation
FGS	Fine Granularity Scalability
GFX	Graphics Framework eXtensions
GMC	Global Motion Compensation
GOP	Group of Pictures
GOS	Group of Scene Planes

GOV	Group of Video Object Planes
GSM	<i>Groupe Spécial Mobile</i>
GUI	Graphical User Interface
HDL	Hardware Description Language
HDTV	High-Definition Television
HEC	Header Extension Code
HRD	Hypothetical Reference Decoder
HVS	Human Visual System
IEC	International Electrotechnical Commission
IPMP	Intellectual Property Management and Protection
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
IST	<i>Instituto Superior Técnico</i>
ITU	International Telecommunication Union
ITU-R	Radiocommunication Sector of ITU
ITU-T	Telecommunication Standardization Sector of ITU
JPEG	Joint Photographic Experts Group
JTC	(ISO/IEC) Joint Technical Committee
JVT	(MPEG/VCEG) Joint Video Team
KLT	Karhunen-Loève Transform
LASER	Lightweight Application Scene Representation
LPE	Low Pass Extrapolation
MB	MacroBlock
MDC	Multiple Description Coding
MIME	Multipurpose Internet Mail Extensions
MJPEG	Motion JPEG
MPEG	Motion Picture Experts Group
MRF	Markov Random Field
MSE	Mean Square Error
MV	Motion Vector
MVD	Motion Vector Difference
MVDS	Motion Vector Difference for Shape
MVO	Multiple Video Object
MVS	Motion Vector for Shape
NEWPRED	New Prediction
NTSC	National Television Systems Committee
OBMC	Overlapped Block Motion Compensation
OCI	Object Content Information
OD	Object Descriptor
OFFS	Open Font Format Specification
ORDF	Operational Rate-Distortion Function
PAL	Phase Alternating Line
PC	Personal Computer
PDA	Personal Digital Assistant
PDF	Probability Density Function
PSNR	Peak Signal-to-Noise Ratio
PSTN	Public Switched Telephone Network
QCIF	Quarter Common Intermediate Format
QoS	Quality of Service
RMS	Root Mean Square

RR	Reduced Resolution (VOP)
RTCP	RTP Control Protocol
RTP	Real Time Transport Protocol
RVLC	Reversible Variable Length Coding
SAD	Sum of Absolute Differences
SA-DCT	Shape-Adaptive DCT
SAF	Simple Aggregation Format
SDTV	Standard-Definition Television
SECAM	<i>Séquentiel Couleur avec Memoire</i>
SIF	Source Intermediate Format
SP	Scene Plane
SR	Scene (temporal) Rate
SSE	Sum of Square Errors
SVD	Singular Value Decomposition
SVG	Scalable Vector Graphics
SVO	Single Video Object
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Locator
VBV	Video Buffering Verifier (MPEG-1 and -2 context)
VBV	Video Rate Buffer Verifier (MPEG-4 context)
VCD	Video Compact Disc
VCEG	Video Coding Experts Group
VCV	Video Complexity Verifier
VLC	Variable Length Coding
VM	(Video) Verification Model
VMV	Video Reference Memory Verifier
VO	Video Object
VOL	Video Object Layer
VOP	Video Object Plane
VP	Video Packet
VS	Visual Object Sequence
VTC	Visual Texture Coding
WWW	World Wide Web
XML	Extensible Markup Language
XMT	Extensible MPEG-4 Textual

Chapter 1

Introduction

1.1 Context and Motivation

In the last two decades, the world witnessed a large scientific and technological evolution in the areas of telecommunications, computers, and TV/cinema. This “Digital Age” started in the second-half of the 20th century, when digital computers and related technologies were developed. Elements that have historically belonged to each of the mentioned areas appear now indistinctively in the others. Computers are using audio, video, and communication capabilities; video and interactivity are being added to the telecommunications world; interactivity is coming to TV/cinema. In fact, the creation, exchange, storage, access, and manipulation of audiovisual information play an increasingly growing role in modern societies. A few technological advances have concurrently contributed to this “Digital Revolution” creating the need for new ways of representing, integrating, and exchanging audiovisual information, namely:

- The continuous increase in the available computational capacity with the progress of microelectronics that is providing extremely powerful and programmable processors.
- The evolution of the network infrastructure with the deployment of diverse new delivery systems such as fixed broadband and mobile networks.
- The evolution of the audiovisual information production and consumption paradigms, with the deployment of personal devices with still image and video capabilities, increasing availability of computer generated information, and higher degrees of interactivity in the Internet.

Digital audio and video had an essential role in the continuously changing audiovisual landscape. A popular example dates back to the 1980s with the digitalization of music, with the introduction of the compact disk (CD) to replace analog devices such as vinyl records and

magnetic tapes.

Having all information in digital format allows the use of similar techniques and systems to process, transmit, and store a large range of digital data types, notably audio and video, which is the basic principle behind multimedia [1]. However, the digital representation has its own price – the high bandwidth required to transmit/store the digital signal.

In the field of digital video, the model originally adopted was just a digital representation of the corresponding analog signal. In fact, both analog and digital video consisted of a periodic sequence of (rectangular) frames (progressive video) or fields (interlaced video). The major difference between these two models is the fact that in the analog representation each frame or field is made of a number of (analog) lines, whereas in the digital representation the frames or fields are matrices of picture elements (pixels) [2]. Nowadays, this type of digital video is commonly referred to as “frame-based video”.

In an attempt to standardize the digital video representation, several organizations defined standards for digital video formats for different application scenarios. For representing standard definition television (SDTV) video signals in digital format, the typical format is ITU-R BT.601 [2], developed by the International Telecommunications Union (ITU) – Radiocommunication Sector (ITU-R). For high definition television (HDTV), the Society of Motion Picture and Television Engineers (SMPTE) defined several digital video formats, notably SMPTE 295M [3] and SMPTE 296M [4]. Other formats have been derived from the ITU-R BT.601 basic format for other classes of application: the source intermediate format (SIF) for digital recording on video compact disc (VCD); the common intermediate format (CIF) for videoconference over ISDN¹; and the quarter common intermediate format (QCIF) for videotelephony over wired or wireless telephone networks. Table 1.1 summarizes the main characteristics of these digital video formats, and presents examples of applications where these formats can, typically, be used.

A common feature of all these video formats is the high raw data rate of the digital signals. Notice that digital frame-based video is usually characterized by the frame/field rate (temporal resolution), the number of lines per frame² as well as the number of samples per line³ (spatial resolution), and the number of bits used to represent each pixel (pixel depth)⁴. Together these characteristics define the raw data rate, R , generated by the digitalization process, i.e.,

$$R = (\text{temporal resolution}) \times (\text{spatial resolution}) \times (\text{pixel depth}) [\text{bit/s}].$$

¹ Integrated Digital Services Network (ISDN).

² Also known as vertical spatial resolution.

³ Also known as horizontal spatial resolution.

⁴ For color signals the pixel depth depends on the chroma sampling format. For 8-bit video, i.e., when the luminance and chrominance are digitalized with 8 bit/sample, the following pixel depths are commonly used:

- 1) 24 bit/pixel – The luminance and the two chrominances are sampled at the same sampling rate – 4:4:4 chroma sampling format (e.g., used in high-quality studio video production).
- 2) 16 bit/pixel – The chrominances are horizontally sub-sampled at half the sampling rate – 4:2:2 chroma sampling format (e.g., used in high-quality studio video production).
- 3) 12 bit/pixel – The chrominances are sub-sampled at half the sampling rate in both directions (horizontal and vertical) – 4:2:0 chroma sampling format (e.g., used in video storage and distribution).

Table 1.1 – Digital video formats for different applications [5]

Video Format	Y Spatial Resolution	Color Sub-sampling	Frame Rate P–(frame/s), I–(field/s)	Raw Data Rate (Mbit/s)
<i>HDTV over air, cable, satellite, MPEG-2, 20–45 Mbits/s</i>				
SMPTE 296M	1280×720	4:2:0	24P ⁵ /30P/60P	265/332/664
SMPTE 295M	1920×1080	4:2:0	24P/30P/60I ⁶	597/746/746
<i>Video production, MPEG-2, 15–50 Mbits/s</i>				
ITU-R BT.601	720×480/576	4:4:4	60I/50I	249
ITU-R BT.601	720×480/576	4:2:2	60I/50I	166
<i>High-quality video distribution (DVD, SDTV), MPEG-2, 4–8 Mbits/s</i>				
ITU-R BT.601	720×480/576	4:2:0	60I/50I	124
<i>Intermediate-quality video distribution (VCD, WWW), MPEG-1, 1.5 Mbits/s</i>				
SIF	352×240/288	4:2:0	30P/25P	30
<i>Videoconferencing over ISDN/Internet, H.261/H.263, 128–384 kbits/s</i>				
CIF	352×288	4:2:0	30P	37
<i>Videoconferencing over wired/wireless modem, H.263, 20–64 kbits/s</i>				
QCIF	176×144	4:2:0	30P	9.1

For example, the raw data rate of ITU-R BT.601 considering the digitalization of PAL⁷/SECAM⁸ analog video (720 pixels/line, 576 lines/frame, 50 fields/s, 8 bit/sample, and 4:2:0 chroma sampling format) for SDTV leads to a raw data rate of approximately 124 Mbit/s; the same raw data rate obtained for the NTSC⁹ analog video (720 pixels/line, 480 lines/frame, 60 fields/s, 8 bit/sample, and 4:2:0 chroma sampling format) – see Table 1.1.

In this context, compression assumes a fundamental role in enabling many applications making use of digital video, and has been addressed by video coding standardization efforts, such as H.261, H.263, MPEG-1 Video, and MPEG-2 Video [7, 8, 9, 10]. Frame-based video coding algorithms aim at reducing the number of bits necessary to represent a given video sequence by exploiting the temporal, spatial, and statistical redundancies in the input data. While the exploitation of redundancy does not introduce any losses, many applications accept that the decoded sequence is not precisely equal to the original sequence, provided that the losses are not visually noticeable or, at least, that the losses have a subjective impact that is acceptable for the application in question. This fact led to the exploitation of irrelevancy which is the information present in the original signal but to which the human visual system (HVS) is not sensitive. Elimination of irrelevancy in the coding process, although irreversibly affecting the signal, can provide significant increases in compression efficiency without a any impact on the final subjective quality. If higher compression ratios are requested, distortion

⁵ P denotes Progressive video format.

⁶ I denotes Interlaced video format.

⁷ Phase Alternating Line (PAL) television standard [6] used in most European countries.

⁸ *Séquentiel Couleur avec Mémoire* (SECAM) television standard [6] used in France.

⁹ National Television Systems Committee (NTSC) television standard [6] used in North America and Japan.

that goes beyond irrelevancy may be introduced in a controlled way.

Compression is currently supported by the traditional frame-based video coding standards, H.261, H.263, MPEG-1 Video, and MPEG-2 Video, by using hybrid coding schemes, which mainly consist in the application of transform coding on the motion compensated prediction error, followed by entropy coding (variable length coding or arithmetic coding). Figure 1.1 roughly illustrates the evolution of video coding standardization over the past twenty years in terms of compression ratios achieved. For example, a videoconference service using CIF video and the H.261 video coding standard over a primary ISDN access of 2048 kbit/s (E1 data rate) requires a compression ratio for video of approximately 20:1; a one hour MPEG-1 video stored in a VCD roughly requires a compression ratio of 25–40:1; a H.263 videophone operating over a PSTN¹⁰ requires a compression ratio higher than 50:1, etc. Although, in this time period, the achievements in the area of video compression have been impressive, the time to stop research on video coding has not arrived yet, as recent developments have shown [11, 12].

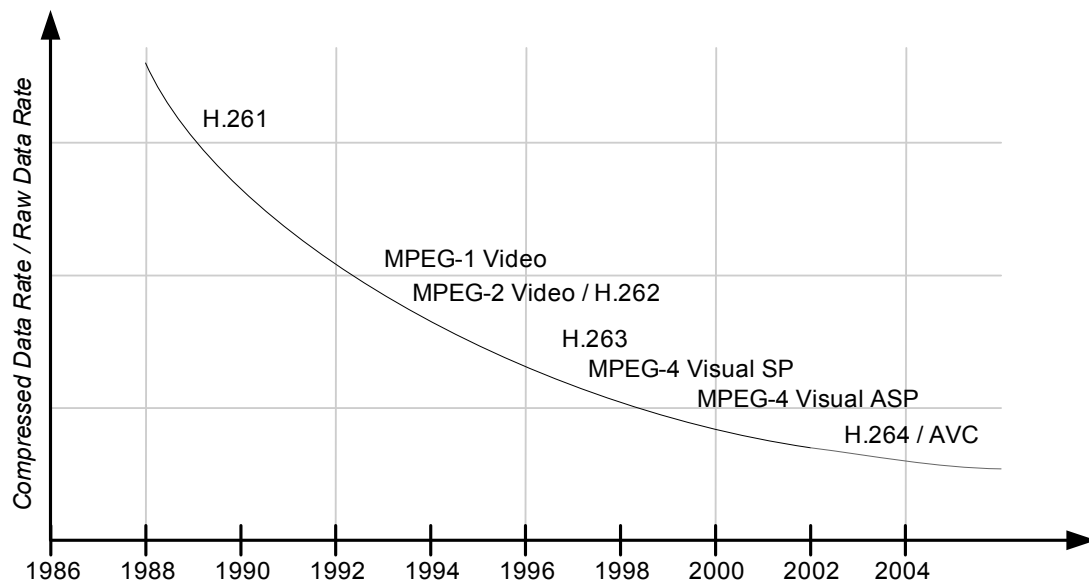


Figure 1.1 – Evolution of the compression ratio for the main international video coding standards

Although compression has been one of the main functionalities of video coding standards, with the widespread of multimedia applications and services, functionalities such as content-based interactivity and improved coding efficiency are now also requested by the users and content providers. This requires new audiovisual representation methods, notably supporting content-based access and manipulation of visual objects. A coding architecture that provides an independent representation of audiovisual objects can provide the requested interactive functionalities. In the case of video, it will be possible to:

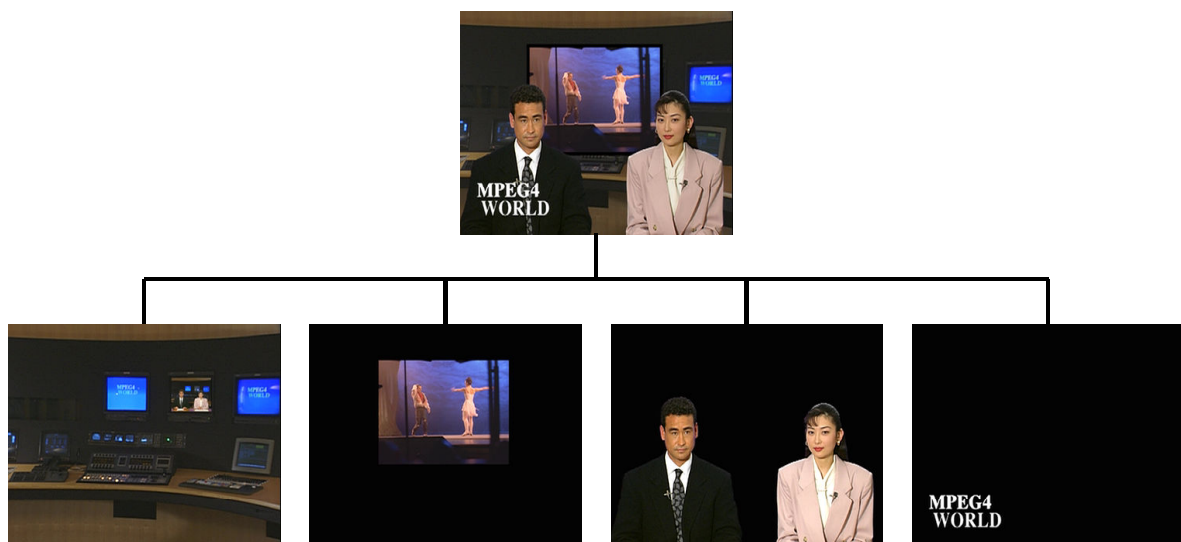
- Focus the attention of the user into a particular video object by coding it with better quality, temporal resolution, or error resilience.
- Allow the user to select, manipulate, and change attributes of semantically meaningful video objects.

¹⁰ Public Switched Telephone Network (PSTN).

- Compose new video scenes with video objects from different sources or reuse objects from one scene into another.

Therefore, more recently, a new video data representation model has been introduced: the object-based model. In this model, video data is no longer seen as a sequence of frames or fields, but consists of several independent (semantically) relevant video objects that together build the video scene. Figure 1.2 illustrates this new video representation model, presenting a video sequence composed of four video objects.

Video coding schemes based on this new data representation model can also be called content-based video coding schemes since the representation entities in the model (the objects) can have semantic value and can, therefore, be subjected to semantically meaningful actions. This new representation approach allows, in addition to the advantages already provided by the digital frame-based representation, new and improved functionalities in terms of interactivity, coding efficiency, and universal access since, for the first time, the content is not only selectively processed but also independently represented, accessed and consumed.



*Figure 1.2 – News video sequence (top) and its corresponding video objects (bottom):
Background, Dancers, Speakers, and Logo*

This video representation model was generalized with the MPEG-4 standard (ISO/IEC 14496). In this context, MPEG-4 emerged as the first international audiovisual representation standard relying on the concept of audiovisual scenes composed by objects, providing standardized ways to perform a series of content-based functionalities:

- Represent audio, visual, and audiovisual media objects of both natural and synthetic origin.
- Represent the composition of audiovisual scenes resulting from the composition of these media objects.
- Multiplex and synchronize the data associated to these media objects in order to enable its transmission over network channels providing an appropriate quality of service (QoS) for the nature of the specific media objects.
- Interact (and hyperlink) with the audiovisual scene content presented at the receiver's end.

Without a standard way to perform some of the above-mentioned operations, interoperability between different multimedia devices/applications would be very difficult, restraining the users' expectations of accessing and exchanging information without technical barriers.

This new audiovisual object-based coding architecture, allowing the users to access semantically meaningful objects in a scene and to interact more “naturally” with the scene content, clearly represents a step forward in terms of video representation. The strength of this approach is more evident if a comparison with the traditional frame-based video coding approach is made. There, the “real world” is represented by a set of rectangular temporally correlated frames, the frame being the smallest unit that can be independently accessed and manipulated by the user (and carrying a more diffuse semantic value).

With the advent of object-based video, new relevant problems have appeared in various fields, since new types of data, such as the shape and the scene composition information, have to be transmitted in addition to the motion and texture already used in previous frame-based coding systems. One of these fields is video coding rate control, understood as the mechanism responsible for efficiently controlling the video encoder in order that it meets relevant constraints of the encoding framework, as, for example, channel and delay constraints. The challenge in terms of video coding rate control is, therefore, to develop new methods that are able to allocate the available coding resources among the several objects in the scene using appropriate allocation criteria and constraining mechanisms.

1.2 Main Objectives of this Thesis

In order to extend the useful lifetime of a video coding standard, standardization bodies specify, usually, the minimum set of tools essential for guaranteeing interoperability between devices or applications of different manufacturers. With this strategy, the standard may evolve continuously through the development and improvement of its non-normative parts. The rate control mechanism is one of the video coding tools that is not normatively specified in any of the currently available and emerging video coding standards, since this is not necessary for interoperability, although providing one of the main degrees of freedom to improve the performance of standard-based systems. Generically, the major objectives of video coding rate control, whatever the coding architecture, can be summarized as:

- Regulation of the video encoder output data rate according to specific constraints.
- Maximization of the subjective impact of the decoded video.

In frame-based coding architectures, the rate control mechanism has, usually, the degrees of freedom to choose the best trade-off in terms of spatial and temporal resolution as well as to introduce (controlled) distortion in the texture data, to maximize the global subjective quality of the decoded video, given the resources and conditions at hand.

In object-based coding architectures, the degrees of freedom of frame-based video coding rate control appear now for each object in the video scene. Additionally, there is the shape data that defines the shape of each object, and the scene description data that specifies which objects are in the scene and the way that the scene is organized. The major novelty here is the semantic dimension of the data model and consequently of the rate control mechanism. This mechanism can decide to perform actions such as not transmitting a less relevant object to save bits for the most semantically relevant objects, or to dynamically allocate the available bits to the various objects depending on their subjective and semantic relevance. Therefore, for object-based coding, the relevant criteria to be used for rate control are related not only to

the texture and shape characteristics of each object but also to their semantic dimension. This later is in turn related to the priority and relevance of each object in the context of the scene and the application in question.

This Thesis aims at studying and developing efficient methods for video coding rate control in the context of object-based video coding architectures such as the one provided by the MPEG-4 standard, notably methods for controlling a video encoder in order to produce standard compliant bitstreams. Therefore, this Thesis concerns the design, specification, and evaluation of bit rate control algorithms for object-based video coding architectures, namely:

- Identify and organize the relevant requirements and constraints of bit rate control in object-based coding environments.
- Investigate and analyze the new bit rate control dimensions and strategies opened by the object-based coding approach.
- Study, propose, and evaluate mechanisms for guaranteeing efficient interoperability among devices/applications targeting compliance with an object-based video coding standard.
- Design an appropriate rate control architecture suitable for the implementation of an object-based rate control algorithm integrating a compliance mechanism.
- Develop suitable mathematical models for the different components of an object-based rate control mechanism.
- Design and evaluate bit rate control solutions for relevant video coding scenarios, notably, in the context of the different MPEG-4 Visual Profiles.

It is worth stressing here that, although the MPEG-4 standard provides the adequate framework for testing and validating the concepts and tools developed in this Thesis, these concepts and tools are, generically, standard independent and could be applied to other object-based coding architectures.

1.3 Summary of Original Contributions

Fulfilling the defined objectives, this Thesis includes several contributions to the field of rate control for object-based video coding, notably, analysis and organization of the problem, specification of an adequate framework and mechanisms for compliant object-based video encoding, rate and distortion modeling, and compensation and adaptation algorithms for efficient low-delay object-based rate control. A more detailed description of these contributions is presented below.

RATE CONTROL FOR OBJECT-BASED VIDEO CODING: PROBLEM DEFINITION

The first contribution of this Thesis concerns the analysis and definition of the rate control problem for object-based coding architectures [13, 14]. The new dimensions and strategies for rate control when an object-based coding architecture is used are defined in [13], notably the amount of content control and the semantic resolution control. A joint rate control methodology for encoding scenes with multiple arbitrarily shaped video objects is proposed in [14]. This methodology defines a set of tasks targeting the achievement of as much as possible near constant quality within the scene and along time in the context of constant bit rate (CBR) coding environments. One of the most relevant questions studied in the context of the rate control method proposed in [14] is the efficacy of several video object bit rate allocation

criteria, concluding that a weighted combination of the size, object activity, and texture complexity criteria leads to smoother quality variations both spatially within the composed scene and along time.

MPEG-4 OBJECT-BASED COMPLIANT VIDEO ENCODING

In order that a particular set of visual bitstreams building a scene may be considered compliant with a given MPEG-4 Visual profile@level, it must not contain any disallowed syntax element for that profile and additionally it must not violate the MPEG-4 Video Buffering Verifier mechanism constraints.

In the field of object-based compliant video encoding this Thesis provides a detailed analysis of the three normative MPEG-4 video buffering verifier models¹¹ – the video reference memory verifier (VMV), the video complexity verifier (VCV), and the video rate buffer verifier (VBV) [16] – demonstrating that these models exhibit some drawbacks, notably, in comparison with relevant alternative models proposed in the context of this Thesis. In particular, this Thesis proposes various improvements to the VMV and VCV models: for the VMV to better reflect the fact that composition is not normative in MPEG-4, and therefore the normative memory management process should reflect this, while for the VCV to better reflect the real complexity of visual scenes, notably those containing a high percentage of transparent macroblocks (MBs).

Still in this field, this Thesis proposes a coherent set of video coding rate control strategies, defined in terms of the reactions taken by the rate control module, to guarantee a compliant encoding, considering the status of the various video buffering verifier buffers [16]. Results obtained from applying some of the rate control strategies proposed in [16] to the encoding of video material, compliant with several MPEG-4 Visual profiles, notably the Simple and Core profiles, stressed the weaknesses of the MPEG-4 Video Buffering Verifier mechanism, notably the possible overestimation of the scene complexity due to the way the several types of MBs are counted [17, 18, 19]. Therefore, the decoding complexity of the several MB types used in MPEG-4 video coding is evaluated in [20, 21], by using statistics of the MB decoding times obtained with an optimized MPEG-4 video decoder. Based on these statistics, a set of relative complexity weights for the relevant MB complexity classes, which can be used to improve the standard MPEG-4 VCV model, is proposed. Following this work, in [22, 23] an alternative VCV model, based on a set of relative MB complexity weights assigned to the various MB coding types used in MPEG-4 video coding is proposed. This new VCV model allows a more efficient use of the available decoding resources by preventing the over-evaluation of the decoding complexity of certain MB types and thus making possible to encode scenes (for the same profile@level decoding resources) that otherwise would be considered too resource demanding and consequently not allowed to be coded at that profile@level.

RATE AND DISTORTION MODELING FOR LOW DELAY VIDEO ENCODING

Rate and distortion models can play a very important role in the context of real-time video encoding since they can be used to obtain near optimal operational performance in terms of the rate-distortion tradeoff without the drawback of having to encode multiple times the same video object plane (VOP) to find the best combination of coding parameters. In the context of object-based video encoding, rate and distortion models characterize the relation between the

¹¹ A detailed description of this mechanism is provided in [15].

average number of bits/pixel to code a given VOP, the average VOP distortion, and the relevant coding parameters. In this Thesis these models are defined in terms of rate-quantization (RQ), distortion-quantization (DQ), and rate-distortion (RD) functions. This Thesis proposes efficient RQ, RD, and DQ models for Intra and Inter coding [24, 25]. For Inter coding, the RQ, DQ, and RD functions depend also of the reference VOPs and become, respectively, $R(Q, Q_{ref})$, $D(Q, Q_{ref})$, and $R(D, D_{ref})$. Since these bidimensional rate and distortion functions are difficult to obtain, at least for a wide range of Q and Q_{ref} values, some assumptions need to be made. Therefore, this Thesis proposes a new type of models, for small Q variations between successive VOPs, where the RQ and DQ functions are approximated by a stationary component, depending only on Q , plus a delta component that represents the difference between the actual and the approximated function using only the stationary component. These models are used in the proposed rate-control algorithm to compute the target quantization parameters for the various VOs in the scene.

RATE CONTROL ALGORITHM FOR LOW DELAY VIDEO ENCODING

In the field of object-based rate control, this Thesis proposes an algorithm for single video object (SVO) and multiple video object (MVO) rate control built of six main modules. This algorithm optimizes the spatio-temporal quality trade-off, notably when compared with existing algorithms as those in MPEG-4 Visual Annex L, while meeting other relevant rate control objectives, such as, not violating the video buffering verifier constraints, and keeping a stable and close to the target VBV occupancy. This rate algorithm, partially described in [14, 17, 26], incorporates also some of the previously mentioned contributions.

The main contributions in this field are distributed over the six modules composing the proposed rate algorithm architecture: the scene-analysis for resource allocation, the spatio-temporal resolution control, the rate-distortion modeling, the bit allocation, the video buffering verifier control, and the coding mode control.

In object-based video coding it is important to extract the changing characteristics of the different VOs in the scene in order to be able to properly allocate the available encoding resources according to the amount and complexity of the data to be encoded in each encoding time instant. This task is performed by the scene-analysis for resource allocation module proposed in this Thesis, which extracts relevant characteristics of all VOPs to be encoded for each possible encoding time instant before encoding any VOP (i.e., the size, object activity, and texture complexity). To properly tackle this problem, a new analysis architecture is proposed and has been implemented, allowing to efficiently perform the necessary scene analysis functions [26, 27].

The adequate spatial and temporal resolutions for encoding each VO should be defined by the rate control mechanism in the context of optimizing the spatio-temporal quality trade-off under the constraints imposed by the selected profile@level. The spatio-temporal control module developed in this Thesis solely controls the temporal resolution of the different VOs in the scene, as for the MPEG-4 profiles implemented (Simple, Core, and Main) the spatial resolution cannot be altered during encoding. This module receives input from the scene analysis, the bit allocation, and the video buffering verifier modules, and based on this information either skips or encodes the current set of VOPs.

In order to be able to predict the behavior of the scene encoder, this Thesis proposes two levels of rate-distortion modeling: VOP-level and MB-level. While at the VOP-level the goal is to set a target average quantization parameter that leads closely to the target number of bits

allocated to the corresponding VOP. On the other hand, at the MB-level, the main goal of these models is to modulate the VOP-level quantization parameter in order to provide a fine adjustment of the VOP encoded bits and provide the rate controller with a faster reaction to deviations relatively to the nominal operation.

In order to maximize the subjective quality of the decoded video it is important to maintain the quality of the decoded VOPs approximately constant along the sequence. In this field, this Thesis proposes that the bit allocation to be performed along several hierarchical levels, notably, group of scene planes (GOS), scene plane (SP), video object plane (VOP), and macroblock (MB). For each hierarchical level a bit allocation strategy is proposed that defines a nominal target bit allocation and an adequate compensation mechanism for dealing with deviations relatively to this nominal target. In order to keep the VOP spatial quality approximately constant along the sequence and in each SP, even when different VOP coding types are used, this Thesis proposes also two feedback mechanisms for adjusting, respectively, the VOP coding type weights and the VO complexity weights.

In order that the set of bitstreams produced by the scene encoder can be considered compliant with the selected profile@level it must not violate the video buffering verifier mechanism. Therefore, this Thesis proposes a video buffering verifier control module that adequately controls the three MPEG-4 models, i.e., VMV, VCV, and VBV. The major contribution in this field is the proposed VBV control conducted at the SP and MB levels. This mechanism defines, for each level, two types of VBV control: soft VBV control and hard VBV control. Soft SP-level VBV control smoothly adjusts the SP bit allocation through a feedback function that aims at compensating the VBV occupancy relatively to a target VBV occupancy, while hard SP-level VBV control simply restricts the SP bit allocation in order that the VBV occupancy is kept within the defined VBV nominal area. The purpose of the soft MB-level VBV control is essentially to regulate the allowable MB QP variation range, while hard MB-level VBV control is used in extreme cases whenever skipping or stuffing data is required.

The coding mode control module is responsible for deciding the coding parameters of each coding unit, i.e., MB texture and shape (if applicable) coding modes, MB motion vectors (if applicable), and MB quantization parameter. The major contribution in this field is the MB-level quantization parameter selection proposed in this Thesis. The purpose of this MB-level quantization parameter selection is to provide a fine way to compensate deviations relatively to the nominal bit allocations and to maintain the spatial quality inside each VOP approximately constant. The algorithm proposed in this Thesis deals with these conflicting goals defining adequate dynamic conditions for changing the MB quantization parameter based on the VBV buffer occupancy, VOP bit allocation, the MB quantization parameter of the previous MB, and the target quantization parameter for the current MB.

Although each rate control module has its own merits, it is the interaction between all of them that brings the performance gains of the overall algorithm, notably by adequately balancing the frequent conflicting goals of some of these modules.

1.4 Outline of this Thesis

This Thesis tackles the problem of rate control for object-based video coding systems. The context and motivation for this work are presented here in Chapter 1, along with the definition of the objectives of the Thesis, a summary of the main original contributions and an outline of the current document.

Chapter 2 presents an overview of the MPEG-4 standard, since this standard provides the

adequate framework for testing and validating the concepts and tools developed under the scope of this Thesis. After analyzing the context and objectives that motivated the development of this standard, the MPEG-4 Visual coding architecture is introduced and its main video coding tools described. In this context, special attention is devoted to the non-normative parts of the MPEG-4 Visual standard, namely, the rate control tools presented in its informative Annex L. This chapter ends with a description of the MPEG-4 profiling mechanism and a brief analysis of the performance of several MPEG-4 video profiles in comparison with other existing video coding standards.

Chapter 3 analyzes the problem of video coding rate control fostered by the object-based video coding architecture adopted by MPEG-4, notably by highlighting the new dimensions of rate control associated to the semantic dimension of coded data. It also proposes a new framework for object-based video coding rate control where this task is performed by using two levels: the scene-level rate control and the object-level rate control.

The problem of how to control a video encoder in order to produce bitstreams that are compliant with a given MPEG-4 visual profile@level is the main motivation of Chapter 4. Consequently, a detailed analysis of the MPEG-4 video buffering verifier mechanism is provided, discussing its major features and drawbacks, notably in comparison with alternative solutions proposed in this Thesis. Furthermore, this chapter proposes a model for the integration of the MPEG-4 video buffering verifier mechanism into a generic video encoder rate control mechanism in order to produce bitstreams that comply with a chosen MPEG-4 video profile@level.

Chapter 5 considers the design of adequate models for describing the rate-distortion behavior associated to the encoding system. These models must capture the statistical characteristics of the source and describe the encoding process as a function of appropriate encoder control parameters, reflecting the lossy encoding rate-distortion trade-off. This chapter proposes a set of rate and distortion models for Intra and Inter coding in the context of object-based MPEG-4 video encoding, notably for low-delay video encoding scenarios. In the case of Inter coding, since the rate and distortion functions become bidimensional and consequently become more difficult to estimate during encoding, a new approach is proposed where the rate and distortion functions for Inter coding are modeled as one-dimensional functions plus an adaptation term.

Chapter 6 integrates the concepts and tools developed in the previous chapters and proposes a new rate control algorithm for low-delay and constant bit rate application scenarios. This algorithm is capable of efficiently dealing with deviations between the ideal and the actual behavior of the scene encoder, represented, respectively, by the rate-distortion models describing the encoder and the actual encoded results. To deal with these deviations between the theoretical models and the actual coding results, adequate adaptation and compensation mechanisms are proposed to track these deviations and compensate them in order to allow a stable and efficient operation of the encoder. These two problems (adaptation and compensation) are the main focus of Chapter 6 and were tackled along the different modules composing the architecture of the proposed rate control algorithm.

Finally, Chapter 7 presents the main achievements of this Thesis and identifies some directions for future work.

Chapter 2

MPEG-4 Standard: An Overview

2.1 Introduction

The set of standards commonly referred to as the MPEG-4 standard [28-49] emerged as the first audiovisual representation standard relying on the object-based representation model. This new audiovisual content representation framework introduced the concept of audiovisual scenes built using individual objects that have relationships in space and time, allowing new and improved functionalities, such as content-based interactivity, universal access through a wide range of terminals and networks, and improved coding efficiency [50].

Before tackling some of the video coding problems opened by this object-based representation model, notably in the non-normative field of video coding rate control, this chapter will present an overview of this standard with special emphasis on the video part of the MPEG-4 Visual standard [29].

This chapter is organized as follows: after this introduction, Section 2.2 briefly introduces the context and objectives of the MPEG-4 standard, highlighting its main functionalities, applications, and organization; Section 2.3 is devoted to the MPEG-4 visual coding architecture and syntactic organization; Section 2.4 presents the main video coding tools of the MPEG-4 Visual standard; Section 2.5 describes the non-normative video rate control algorithms of the informative Annex L on rate control of the MPEG-4 Visual standard; Section 2.6 describes the profiling mechanism of the MPEG-4 standard, presents the various MPEG-4 Visual profiles, and discusses the performance of MPEG-4 video coding in comparison with other existing international standards; and finally Section 2.7 presents some final considerations on the chapter.

2.2 MPEG-4 Context and Objectives

MPEG-4 was built upon the success of previous MPEG standards, MPEG-1 (ISO/IEC 11172) and MPEG-2 (ISO/IEC 13818), which aimed at making storage and transmission of audiovisual information more efficient by compressing the audiovisual data. Before MPEG-4, in MPEG-1 [9] and MPEG-2 Video [10], digital video was represented in the form of rectangular frames – the television model. In fact, the television paradigm has dominated audiovisual communications for many years. However, the production, delivery and consumption of audiovisual content has suffered a dramatic change over the last years [51].

Digital cameras and video recorders became very popular due the advances in microelectronic technology, making content production easier than ever. Additionally, computer generated content, such as music or 2D and 3D animations, is also very common nowadays. In this context, anyone is a potential content producer, capable of creating and re-using content that can be easily published and distributed on the Internet.

In terms of visual content delivery, the trend is nowadays towards the generalization of visual information in every single network, notably, mobile and wireless networks, while previously only a few networks used to transport this type of content (e.g., TV networks).

Furthermore, the widespread of handheld personal computing devices and the explosion of the World Wide Web with its interactive mode of operation, has also changed the way audiovisual content is consumed, with users requiring access to audio and video as they used to have access to text and graphics.

Recognizing the need for an open and timely international standard supporting the interworking requirements of many emergent audiovisual applications, in 1993, the Moving Picture Experts Group (MPEG) launched the MPEG-4 project, later formally called “Coding of Audio-Visual Objects” [52, 53, 54]. Therefore, the MPEG-4 standard (ISO/IEC 14496) emerged with the objective of defining an audiovisual coding standard for the communication, interactive, and broadcast service models, as well as for combinations of these service models resulting from their convergence, following the convergence of the telecommunications, computing, and TV/film/entertainment worlds.

2.2.1 The Object-based Representation

The MPEG-4 standard brought to the audiovisual representation arena a new video data representation model: the object-based model. In this model, the video data is no longer seen as a sequence of rectangular frames or fields, but consists of several independent (semantically) relevant video objects that together build the video scene. The object-based video coding architecture adopted by MPEG-4 can also be seen as a content-based video coding architecture due to the natural semantic nature of the objects, which by having semantic value can be subjected to semantically meaningful actions. This new representation approach allows, in addition to the advantages already provided by the digital frame-based representation, new and improved functionalities in terms of interactivity, coding efficiency and universal access, since, for the first time, the content is not only selectively processed but also independently represented, accessible and consumed.

To allow content-based access and manipulation, the coded data are organized in terms of elementary streams (ESs). In the MPEG-4 standard, all information is conveyed to the receiver through ESs, each containing the coded representation of an audiovisual object, the scene composition information, or control information, e.g., to animate a 3D face model.

These ESs are identified and characterized through object descriptors (ODs) [28], which convey information about the type of content and the necessary decoding resources. ODs are also used to gather information about an audiovisual object when alternative representations for the same object are available, e.g., lower or higher bit rate (and, therefore, quality) representations or scalable coding representations using multiple ESs for the same object. The most important information conveyed in an OD is the necessary information to locate the coded representation of the audiovisual object, i.e., the ES(s), in terms of the logic transport channel or uniform resource locator (URL). ODs can also refer to ESs containing content description information – object content information (OCI) – or information related to the intellectual property management and protection (IPMP) associated to the object or the scene. These mechanisms are specified in the MPEG-4 Systems standard [28].

The scene description format specified by the MPEG-4 standard is called binary format for scenes (BIFS). This format allows describing a given visual scene in terms of the spatial and temporal relations between the different objects in the scene. In addition, it allows creating and embedding graphical and textual objects in a given scene. BIFS commands can also be used to put new objects in a scene and dynamically modify the visual or acoustic properties of existing objects without modifying explicitly the object, e.g., changing the color of a 3D object. These commands can also be used to define object behaviors, either unconditionally (e.g., define a particular object movement for a pre-defined time instant) or in response to user interaction. The BIFS format has been defined as an extension to the virtual reality modeling language (VRML) format [55]. Unlike VRML, however, which is a textual format, the BIFS format is binary, to save bandwidth over a textual description. This feature is especially useful for low bandwidth environments. In addition, the BIFS format also supports streaming; unlike the VRML format, where a scene needs to be downloaded from the server to the client before being presented. With the BIFS format, the scene can be presented in real-time as the data is being received.

Figure 2.1 represents a simplified version of the MPEG-4 object-based representation architecture. At the transmitter, the various objects in the scene are separately encoded and the composition information is created. The generated ESs (in addition to locally stored ESs) are multiplexed to form a single (multiplexed) bitstream that is sent through the channel and contains all the information about the scene. At the receiver, the received bitstream is demultiplexed in order to obtain the various ESs corresponding to the objects and the scene composition information. The ESs are then decoded and passed on to the compositor, which will compose the scene based on the composition information. Additionally, at the receiver, local objects can also be added to the scene by the compositor.

A major novelty embedded in this representation architecture, where the several objects in the scene are independently coded, is the possibility of the end-user to interact with the objects, performing operations such as changing the spatial position or the size of objects, adding or deleting objects, triggering pre-defined behaviors, following hyperlinks, or even changing color properties. Depending on the type of interaction, the necessary action will be taken either locally (i.e., at the decoder) or remotely (i.e., at the encoder). For instance, if the user chooses to change the spatial position of a given object in the scene, this can be simply taken care of by the compositor. On the other hand, if the user chooses to delete one object, this can be taken care of more efficiently by the encoder, which does not have to send it any longer, thus saving bit rate resources for the other objects.

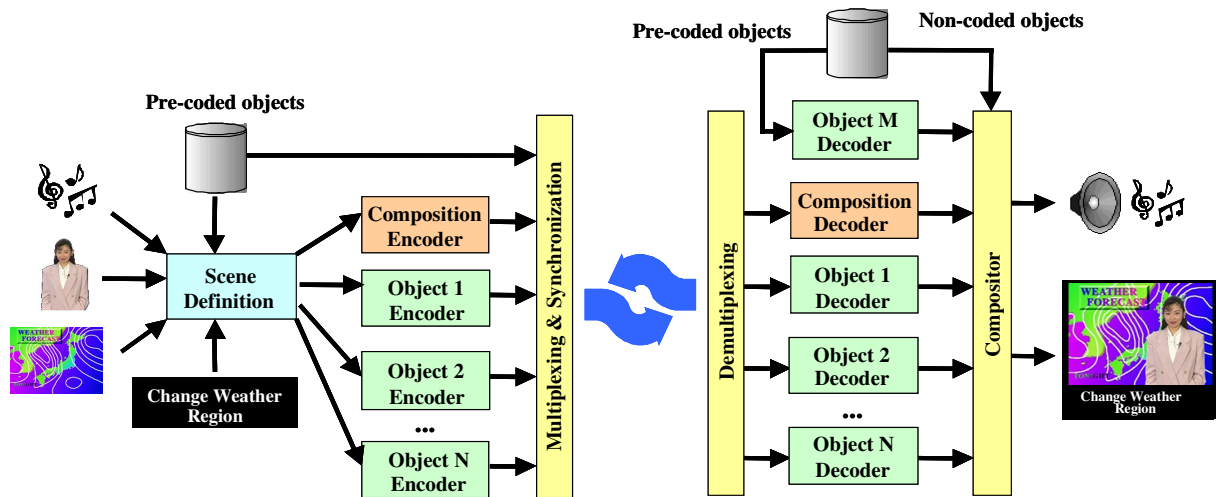


Figure 2.1 – MPEG-4 object-based representation architecture [51]

Having audiovisual scenes composed by independent and semantically relevant objects (i.e., relevant objects in the context of a given application) introduces several new technical and functional possibilities, that can be organized along the following five classes:

- Selective processing and coding of objects** – The object-based coding architecture allows processing and encoding each object with the adequate tools according to its nature. For example, encoding subtitles or other graphic information with the hybrid video coding scheme tools, as it was typically done in previous video coding standards, can be very inefficient since these coding tools are not adapted to these types of data. However, with the new object-based representation, subtitles, for example, can be independently encoded as text characters in addition to some formatting information. This more efficient representation of subtitles can save bits for other more demanding objects leading to more efficient representation of the video content with significant quality gains or bit rate reductions.
- Reusability of objects** – In object-based coding, objects are individually accessible, since an independent bitstream is generated for each object in the scene. Therefore, a given object can be easily reused in another scene, offering many new possibilities in terms of content creation. For example, it becomes now possible to create new content based only on already existing objects. This way, by using objects stored in large databases and possibly adding new ones, if the means to create them are available (e.g., digital cameras), anyone may become a content creator, capable of creating rather complex and high quality content that can be easily disseminated using the Internet.
- Integration of synthetic and natural content** – The object-based model allows to efficiently integrate in the same scene natural (i.e., acquired with a camera) and synthetic objects (i.e., computer generated). For example, it is possible to have scenes where cartoon characters coexist with live actors, each being encoded with the adequate tools. The synthetic characters can be encoded as three-dimensional (3D) wire frame models, thus allowing new possibilities in terms of interaction, typically not available with natural objects. For instance, a synthetic 3D object can be easily rotated, thus uncovering regions of the object that were not previously visible. This is possible with fully specified 3D synthetic objects that can be rendered in the chosen position.

- **Interaction with and between objects** – Independent object-based coding opens new degrees of interaction with the audiovisual content. For example, the user can drag a given object changing its spatial position, click on the object to obtain more information, or even modify the visualization perspective of a 3D scene. In addition, objects can also interact with each other, e.g., an object can change its trajectory due to a collision with another object. Therefore, with these kinds of interaction it becomes possible to create many interesting applications using the same coding framework.
- **Universal access to the content** – With the object-based model, true universal access may finally become a reality, meaning that the content is available through any type of network and accessed with any type of terminal. Since some of the targeted networks, such as some mobile networks, have low bandwidth and critical channel error characteristics, adequate object bit allocation and error robustness has to be provided. For instance, error resilience can be selectively introduced, making more robust to errors the more important objects, without sacrificing their quality, by allocating more bit rate resources to the more important objects and sacrificing the less important ones. Still another possibility is to consider content scalability, where only the more important objects will be sent to the receiver, in order to guarantee that the received content has acceptable quality. This last approach may also be the solution for low complexity terminals without enough resources to decode the whole scene.

As stated above the object-based representation provides many new functionalities that can be using in new or enhanced applications, through a wide range of applications. The next sections introduce, respectively, the major functionalities and applications that are enabled.

2.2.2 Functionalities

After a first definition phase, where the MPEG-4 project aimed at essentially providing improved compression efficiency relatively to previous audiovisual representation standards, the project changed its focus to provide new ways of communicating, accessing, and manipulating digital audiovisual data [51]. Instead of being targeted for a particular class of applications, as MPEG-1 (audiovisual storage on CD-ROM) or MPEG-2 (digital television), MPEG-4 specifies efficient coding tools for a broad class of applications and bit rates supported by the object-based representation model.

In this context, the MPEG-4 project identified a set of eight new functionalities that were not supported (or at least not well supported) by the available standards, reflecting the new audiovisual trends on the TV/film/entertainment, computing, and telecommunications worlds. These eight new or improved functionalities can be grouped in three different classes reflecting the convergence trends of these three worlds [56]: content-based interactivity, compression efficiency, and universal access.

CONTENT-BASED INTERACTIVITY

This class, coming essentially from the computing world, includes four main functionalities that address applications that support content-based user interaction.

Content-based Multimedia Data Access

The object-based data model allows efficient content-based organization and access to the audiovisual content, such as, indexing, hyperlinking, and browsing. For example, with this content-based representation the user can retrieve individual objects from online libraries or

databases, instead of retrieving the whole scene where the object is embedded. Content-based access is a key functionality for the reusability of the audiovisual objects.

Content-based Manipulation and Bitstream Editing

The object-based data model allows the user to interact directly with the audiovisual content by selecting, for example, one specific object from a scene and changing some of its characteristics without the need for transcoding (i.e., decoding and recoding). MPEG-4 supports changing the size of an object, changing its temporal and spatial position, obtaining extra information about objects, or selecting other types of information while seeing a given program.

Hybrid Natural and Synthetic Data Integration

MPEG-4 supports the efficient integration of synthetic and natural objects in the same scene, such as arbitrarily shaped natural video objects, text and graphics overlays, natural and synthetic audio, or animated 3D faces and bodies.

Improved Temporal Random Access

MPEG-4 provides efficient methods to randomly access, within a limited time and with fine granularity, parts (e.g., frames or objects) of an audiovisual sequence. For example, access to audiovisual information from a remote terminal over a low bandwidth channel, or fast-forward on a single object in the sequence.

COMPRESSION EFFICIENCY

This class includes the functionalities that target applications requiring high coding efficiency, notably for audiovisual data storage and transmission.

Improved Coding Efficiency

MPEG-4 specifies more efficient coding tools, relatively to previous coding standards such as MPEG-1 and MPEG-2, and additionally allows selecting the adequate coding tools for each data type. Consequently, the coding tools can be adapted to the characteristics of each object, achieving higher compression ratios, or equivalently, higher qualities for the same compression ratio. Additionally, MPEG-4 provides high coding efficiency at low bit rates, targeting, notably, low bandwidth mobile networks and limited capacity media, such as chip cards and several types of memories.

Coding of Multiple Concurrent Data Streams

MPEG-4 supports the ability to efficiently code multiple views (e.g., stereoscopic video) or various soundtracks of a scene and to efficiently synchronize the resulting ESs. This functionality has been addressed only recently [57] for stereoscopic and multiview video applications. For these applications, MPEG-4 shall exploit the redundancy in multiple views of the same scene, also permitting solutions that allow compatibility with normal video. This functionality shall provide efficient representation of natural 3D objects for applications such as virtual reality games, 3D movies, training and flight simulation, and multimedia presentations.

UNIVERSAL ACCESS

This class includes functionalities that support the ability to handle MPEG-4 encoded

audiovisual data, over a wide range of terminals and channels, with different levels of quality in terms of spatial and temporal resolutions of each object.

Robustness in Error-prone Environments

Universal multimedia access requires access to audiovisual content over different types of networks (wired and wireless) and storage media. To achieve this goal, the MPEG-4 standard provides error robustness tools that can improve the subjective quality of the received data, notably for low bandwidth mobile networks subject to severe error conditions. This includes resynchronization tools and provide the ability to conceal potential errors.

Content-based Scalability

Scalability means the ability to decode only a part of a whole bitstream, according to the channel and terminal resources, obtaining this way a useful representation of the scene or the object. Given a certain (multiplexed) bitstream, decoders with different computational capabilities and bandwidth available can still decode and present a useful part of the audiovisual content (not necessarily with the same quality).

In this context, an MPEG-4 terminal with less resources (computational and/or bandwidth) may decode only the more important objects – content- or object-based scalability; eventually, may decode each object with less quality – quality scalability; or less temporal and spatial resolutions – temporal and spatial scalability.

The MPEG-4 standard provides coding tools that support different types of scalability, notably, content/object, quality, spatial, and temporal scalability. This functionality is especially useful for applications such as video transmission for mobile terminals and over the Internet, variable quality video transmission and database querying and browsing at different scales of quality and temporal or spatial resolutions.

Notice that, although MPEG-4 supports many different functionalities, some important tools are not normatively specified since its specification is not essential for guaranteeing interoperability between different terminals. In fact, the non-normative fields of the standard encourage competition, since they open space for improvements, and increase the lifetime of a standard. For example, video analysis aiming at segmenting natural video sequences in order to define the semantically relevant objects in the scene, motion estimation, rate control, and error concealment, are a few examples of non-normative key tools influencing the performance of an MPEG-4 system and distinguishing terminals from different manufactures.

2.2.3 Applications

Unlike previous MPEG standards, where a “killer application” was clearly identified (e.g., storage on CD for MPEG-1 and digital television for MPEG-2), MPEG-4 addresses a wide range of applications in areas such as broadcasting, personal communications, remote surveillance, and multimedia applications.

Besides the set of applications identified by the MPEG group in the MPEG-4 Applications document [58], many others appeared meanwhile, revealing the potentialities of the MPEG-4 technologies [59]. Below, some of the more relevant applications are briefly described:

REAL-TIME COMMUNICATIONS

Real-time communications systems are targeted towards applications that encompass two-way human interaction (e.g., a videophone system) or one-way applications (e.g., a surveillance system) that impose strict one-way delay constraints. MPEG-4 provides coding efficiency and error resilience tools well suited for these environments.

INTERNET VIDEO

Video streaming over the Internet is assuming an increasing importance as shown by the popularity of video repositories and online transmission of news, TV shows and live-concerts. As in mobile networks, the available bandwidth is typically low and packet losses are frequent. Therefore, MPEG-4 coding efficiency, error resilience, and scalability tools are especially adequate for this environment.

MOBILE MULTIMEDIA APPLICATIONS

The explosive popularity of mobile phones and other type of handheld devices clearly unveils the increasing user interest in multimedia applications for such devices. Due to the limited computational capacity of such devices and the low bandwidth and error conditions of wireless networks, this application area can benefit from adopting MPEG-4 technologies, notably, coding efficiency, error resilience, and scalability tools. As an example, the third generation partnership project (3GPP) [60] has been adopting MPEG-4 technologies, thus allowing to foresee an important role of MPEG-4 content in future mobile networks.

TELEVISION PRODUCTION

Virtual environments are having an increasing role in TV content production, with actors being separately filmed in front of blue screens and the background (computer generated or recorded elsewhere) added later or in real-time using chroma-keying techniques. If this content is encoded as separate video objects instead of as rectangular video, composition and content reusability becomes highly flexible. Moreover, TV programs composed of video objects and additional textual and graphical objects can be broadcasted as such, allowing higher degrees of customization and personalization. Content reusability can also improve content production, since programs can be produced easily, faster, and creatively.

INTERACTIVE DIGITAL TELEVISION

Internet growth opened also an increasing interest for more content interactive applications than what can be provided by plain digital television. The user should be able to manipulate text, graphics, images, and audio information in order to customize the content according to his/her preferences. For example, the different objects in the screen can be rearranged at the user's wish or additional information (e.g., stored locally) can be obtained about the program being visualized (or other programs related to the user's interests). Besides the object-based interaction functionalities, hyperlinking mechanisms are also supported by MPEG-4, allowing the user to access remote information or additional objects with the desired information.

GAMES

The popularity of video games clearly reflects the growing interest in user-content interaction capabilities. Many popular video games use 3D graphical environments and characters. Natural video objects can make these games more realistic. Additionally, using standard coded representations can facilitate the personalization of the games through the use of online

databases of environments and characters, possibly connected to the game in real-time.

2.2.4 Organization of the MPEG-4 Standard

The ambitious goals that drove the conception of MPEG-4 led to the standardization of a wide range of tools related to different technological areas. Therefore, in order to facilitate its adaptation and implementation, the standard has been organized initially (Version 1) in six different parts and has been extended as new working items were being addressed. At the time of writing, the set of standards known as the MPEG-4 standard included 22 parts, briefly described below:

PART 1: SYSTEMS

ISO/IEC 14496-1 [28] specifies system-level functionalities for the communication of interactive audio-visual scenes. It specifies tools for scene description, multiplexing, synchronization, buffer management, and management and protection of the intellectual property.

PART 2: VISUAL

ISO/IEC 14496-2 [29] specifies the coded representation of visual objects of natural or synthetic origin. This part contains definitions of the bitstream syntax, bitstream semantics and the related decoding process. It does not specify the encoders, which can be optimized in different implementations.

PART 3: AUDIO

ISO/IEC 14496-3 [30] specifies the coded representation of audio objects of natural or synthetic origin, notably, definitions of the bitstream syntax and semantics, and the decoding process.

PART 4: CONFORMANCE TESTING

ISO/IEC 14496-4 [31] specifies how tests can be designed to verify whether bitstreams and decoders meet ISO/IEC 14496 (parts 1, 2, 3, and 6) specifications.

PART 5: REFERENCE SOFTWARE

ISO/IEC 14496-5 [32] provides software implementations of the ISO/IEC 14496 (parts 1, 2, 3, and 6) including normative and non-normative tools.

PART 6: DELIVERY MULTIMEDIA INTEGRATION FRAMEWORK (DMIF)

ISO/IEC 14496-6 [33] specifies a session protocol for the management of multimedia streaming over generic delivery technologies. DMIF is simultaneously a framework and a protocol. The functionalities provided by DMIF are expressed by a DMIF application interface (DAI) and translated into protocol messages, which may differ depending on the network they operate on.

PART 7: OPTIMIZED REFERENCE SOFTWARE FOR CODING OF AUDIO-VISUAL OBJECTS

ISO/IEC TR 14496-7 [34] specifies encoding tools that enhance both the execution and quality for the coding of visual objects as defined in ISO/IEC 14496-2.

PART 8: CARRIAGE OF ISO/IEC 14496 CONTENTS OVER IP NETWORKS (MP4oNIP)

ISO/IEC 14496-8 [35] provides a framework for the carriage of ISO/IEC 14496 contents over IP networks and guidelines for designing payload format specifications for the detailed mapping of ISO/IEC 14496 content into several IP-based protocols.

PART 9: REFERENCE HARDWARE DESCRIPTION

ISO/IEC TR 14496-9 [36] specifies descriptions of the main video coding tools in hardware description language (HDL) form. These specifications constitute an alternative description to the ones that are provided in ISO/IEC 14496 (parts 1, 2, 5, and 7).

PART 10: ADVANCED VIDEO CODING (AVC)

ISO/IEC 14496-10 [37] specifies video syntax and coding tools (improved coding efficiency tools for rectangular video) developed in the context of the joint project with ITU-T SG16 – the Joint Video Team (JVT). This specification is also published as ITU-T H.264 [61].

PART 11: SCENE DESCRIPTION (BIFS) AND APPLICATION ENGINE (MPEG-J)

ISO/IEC 14496-11 [38] specifies the coded representation of interactive audio-visual scenes and applications, notably: the coded representation of the spatio-temporal positioning of audio-visual objects as well as their behavior in response to interaction (scene description); the coded representation of synthetic two-dimensional (2D) or three-dimensional (3D) objects that can be manifested audibly and/or visually; a textual representation of the multimedia content using the Extensible Markup Language (XML) – the Extensible MPEG-4 Textual (XMT); and system level description of an application engine (format, delivery, lifecycle, and behavior of downloadable Java byte code applications).

PART 12: ISO BASE MEDIA FILE FORMAT

ISO/IEC 14496-12 [39] specifies the structure and uses of the ISO base media file format. Identical text is published as ISO/IEC 15444-12 [62]. This file format is used to contain time-based media such as video and audio. The storage of particular coding schemes is defined in specifications that derive from and reference ISO/IEC 14496-12 [39] and ISO/IEC 15444-12 [62], such as the MPEG-4 file format specified in ISO/IEC 14496-14 [41], or the Motion JPEG file format specified in ISO/IEC 15444-3 [63].

PART 13: INTELLECTUAL PROPERTY MANAGEMENT AND PROTECTION EXTENSIONS

ISO/IEC 14496-13 [40] specifies extensions to the IPMP framework specified in 14496-1 [28].

PART 14: MP4 FILE FORMAT

ISO/IEC 14496-14 [41] specifies the MP4 file format as derived from ISO/IEC 14496-12 [39] and ISO/IEC 15444-12 [62], the ISO base media file format. It revises and completely replaces Clause 13 of ISO/IEC 14496-1, in which the file format was previously specified. The MP4 file format defines the storage of MPEG-4 content in files. It is a flexible format, permitting a wide variety of usages, such as editing, display, interchange and streaming.

PART 15: ADVANCED VIDEO CODING (AVC) FILE FORMAT

ISO/IEC 14496-15 [42] specifies a storage format for video streams compressed using MPEG-4 AVC, notably how MPEG-4 AVC streams are stored in file formats derived from

the ISO base media file format.

PART 16: ANIMATION FRAMEWORK EXTENSION (AFX)

ISO/IEC 14496-16 [43] specifies a general organization of synthetic models in a six-component hierarchy: geometry, modeling, physical, biomechanical, behavioral and cognitive component.

PART 17: STREAMING TEXT FORMAT

ISO/IEC 14496-17 [44] specifies a text stream as a concatenation of text access units that contain text information of a specific format.

PART 18: FONT COMPRESSION AND STREAMING

ISO/IEC 14496-18 [45] specifies font data representation, compression and streaming, providing an efficient mechanism to embed font data in MPEG-4 encoded presentations.

PART 19: SYNTHESIZED TEXTURE STREAM

ISO/IEC 14496-19 [46] specifies the transmission of synthesized texture data as part of the MPEG-4 encoded audio-visual presentation.

PART 20: LIGHTWEIGHT APPLICATION SCENE REPRESENTATION (LASER) AND SIMPLE AGGREGATION FORMAT (SAF)

ISO/IEC 14496-20 [47] defines a scene description format (LASER) and an aggregation format (SAF) suitable for representing and delivering rich-media services to resource-constrained devices such as mobile phones.

PART 21: MPEG-J GRAPHICS FRAMEWORK EXTENSIONS (GFX)

ISO/IEC 14496-21 [48] specifies a fully programmatic solution for creation of custom interactive multimedia applications. In such applications, synthetic and natural media assets are composed in real-time according to a programmed logic expressed in Java.

PART 22: OPEN FONT FORMAT

ISO/IEC 14496-22 [49] defines the Open Font Format Specification (OFFS) – an open font format based on the OpenType specification [64].

2.3 MPEG-4 Visual Coding Architecture

As previously mentioned, the major novelty of the MPEG-4 standard regarding the data model is the concept of audiovisual scenes composed by audiovisual objects. In terms of visual information, a visual scene is, therefore, a scene composed by one or more visual objects. Although MPEG-4 supports several types of visual objects (e.g., video, still texture, 2D and 3D meshes, and face and body animation objects), this Thesis considers only the case of video objects, notably, the problem of controlling the encoding of this type of objects.

As opposed to previous video coding standards that addressed mainly compression efficiency, the MPEG-4 Visual standard [29] specifies coding tools that support additional functionalities, such as coding of arbitrarily shaped video objects, efficient compression of video sequences and still images over a wide range of bit rates, various types of scalability (i.e., spatial, temporal, and quality), and robust transmission in error error-prone

environments. Therefore MPEG-4 provides a set of advanced features that support the creation of rich multimedia content, notably video scenes.

In MPEG-4 video scenes, each object can be characterized by its temporal and spatial position, as well as its shape and texture data. Applications that do not require arbitrarily shaped video objects due to complexity constraints, or simply because they cannot be easily obtained, can still use rectangular video objects, which are a particular case of arbitrarily shaped video objects.

2.3.1 Hierarchical Syntactic Structure

Figure 2.2 presents the hierarchical bitstream structure used to encode a given scene. This structure is described by the following levels:

- **Visual Object Sequence (VS)** – Represents the highest hierarchical level of the coded scene, wrapping the visual objects that compose the visual scene. This hierarchical level defines the profile and level of the scene, and consequently the type and number of visual objects that can be included in the scene (see Section 2.6).
- **Visual Object (VO)** – A visual object (of natural or synthetic origin) is the elementary entity of the scene, typically with semantic relevance (e.g., a person or a background), that the user can access and manipulate. This hierarchical level defines the type of visual object (see Section 2.6) and consequently the coding tools that can be used for each visual object. In the case of video sequences, the visual object is of type video and is commonly referred as a video object¹.
- **Video Object Layer (VOL)** – Each video object can be coded with several layers (scalable encoding) or using a single layer (non-scalable encoding) called video object layers. Each layer corresponds to a certain spatial resolution, temporal resolution, and image quality. Scalability allows reconstructing a given video object from a base layer (which can be decoded on its own) with several enhancement layers (which require decoding of the lower layers – reference layers). Therefore, each video object can be encoded with spatial, temporal, or quality scalability, offering several decoding resolution layers. The receiver can decode the adequate resolutions (spatial, temporal, and quality) according to the channel and terminal characteristics (e.g., available bandwidth and computational power). This hierarchical level covers all the information about the corresponding video object layer, including the coding and decoder configuration parameters and the ES.
- **Group of Video Object Planes (GOV)** – This hierarchical level wraps a set of successive VOPs providing random access points to the ES. Each GOV carries header information (resynchronization marks and timing information) that, besides being useful for random access, is also useful for resynchronization purposes, in case of errors, and for non-linear bitstream editing. In order for the receiver to be able to decode part of an elementary stream without needing to decode all previous VOPs, each GOV starts with an Intra coded VOP, which can be decoded without having to decode any previous VOPs.
- **Video Object Plane (VOP)** – Represents an instance of a video object at a particular

¹ Since this Thesis considers only the encoding of visual objects of type video, from this point onwards the acronym VO will be used to indicate this type of visual objects.

time instant, t . A VOP is composed by its texture data (Y, U, and V components) and additionally its shape information² in the case of arbitrarily shaped video objects. The coding process generates a coded representation of each VOP, together with composition information that allows to arrange the various VOPs into a composed scene.

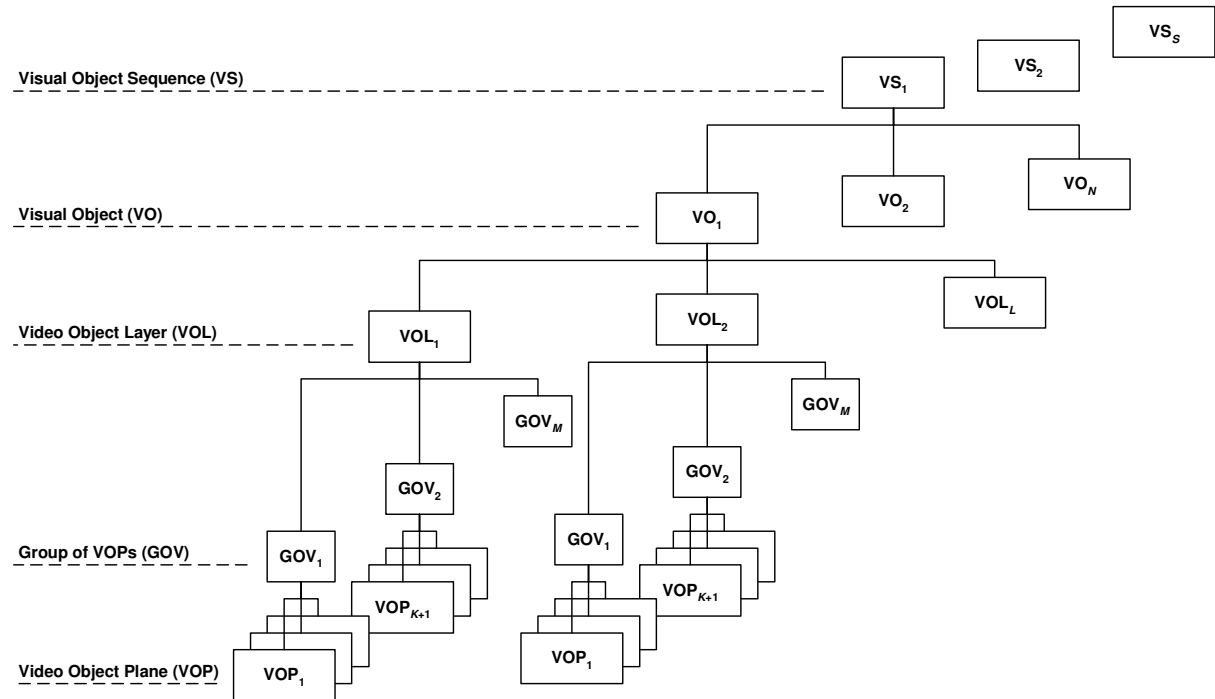


Figure 2.2 – Hierarchical structure of MPEG-4 video bitstreams

As mentioned previously, MPEG-4 does not standardize how a video object is generated (notably, the arbitrary shape information), since it is not required for ensuring interoperability. Therefore, video objects can be obtained using fully automatic or user assisted segmentation techniques [65] (see Figure 2.3) or through chroma-keying techniques where the characters are filmed in front of a single color screen (e.g., blue or green screen) and the video object is defined by the image pixels that have a different color from the background (see Figure 2.4).

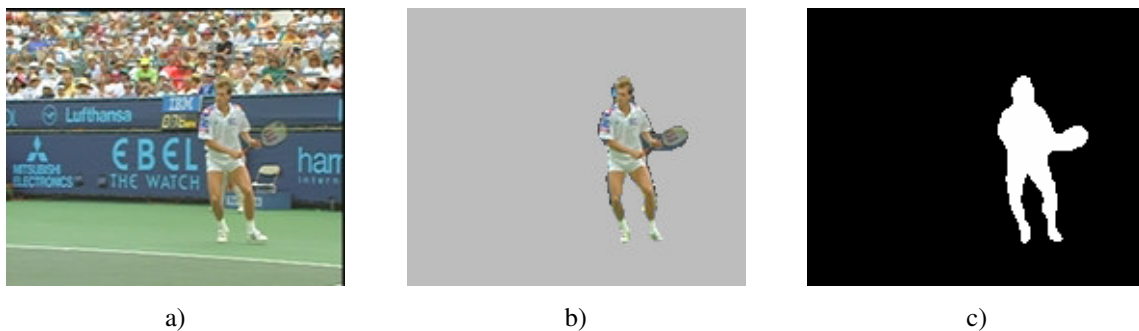


Figure 2.3 – Example video object obtained through segmentation: a) sample image of the Stefan video sequence; b) video object (Player); c) shape information

² The VO shape data matrix is also referred as the “alpha plane” – binary or gray level (see Section 2.4).

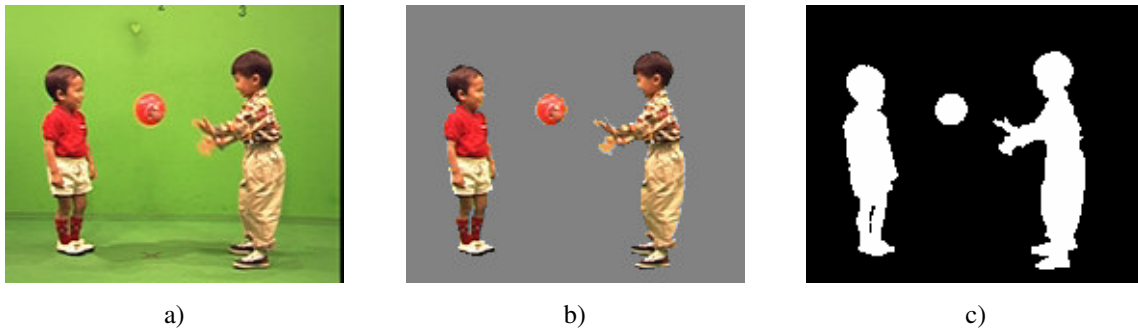


Figure 2.4 – Example video object obtained through chroma-keying: a) sample image of kids shot in front of a green screen; b) video object (Kids); c) shape information

2.3.2 Video Object Coding Scheme

In terms of coding architecture, each VO, encoded according to the MPEG-4 video syntax, is still processed blockwise VOP by VOP, using macroblocks (MBs) of 16×16 pixels³, as in previous MPEG standards. Consequently, for coding purposes, each VOP is enclosed in a tight rectangular bounding box (BB) arranged in blocks of 16×16 pixels (MBs) confining the texture and shape data (see Figure 2.5). This bounding box, computed for each encoding time instant, should be built minimizing the number of MBs with shape pixels. This way of creating the BB is especially useful in the case of performing together shape and texture coding since it decreases the number of MBs that have to be processed and transmitted⁴. In this context, three different types of MBs may appear in the VOP BB (see Figure 2.5):

- **Transparent** – MBs that fall completely outside the VOP. For these MBs, there is no texture (YUV data) to be coded. These MBs are not visible in the decoded scene (this information is conveyed to the receiver through the shape data).
- **Opaque** – MBs that fall completely inside the VOP. These MBs are processed through the block-based hybrid-coding scheme described later in this section, e.g., either Intra coded by coding the YUV samples, or Inter coded by applying motion-compensated prediction and coding the prediction error.
- **Boundary** – MBs that fall in the boundary of the VOP, i.e., containing both transparent and opaque pixels. These MBs are processed using specific tools for coding the MB arbitrary shape data and the MB texture data (see Section 2.4).

³ For the 4:2:0 format each MB contains four blocks of 8×8 luminance samples and two blocks of 8×8 chrominance samples (one block for each chrominance).

⁴ The MPEG-4 video syntax allows to encode binary shape VOs, i.e., VOs without texture information.

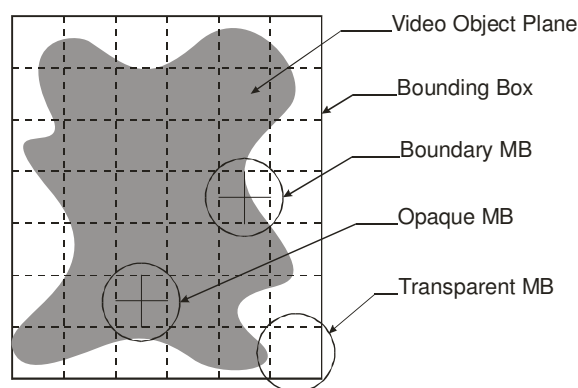


Figure 2.5 – MB types within the VOP bounding box

As mentioned above, the MPEG-4 video coding architecture is based on the typical block-based motion-compensated hybrid-coding scheme. With respect to previous standards, based on the same basic coding scheme, MPEG-4 introduces improvements to most of its modules, adding new or enhanced tools⁵ and algorithms⁶. However, the major novelty, relatively to previous coding schemes is the new shape coding module that supports coding of arbitrarily shaped video objects. Similarly to previous MPEG coding schemes, but considering now arbitrarily shaped video objects, the MPEG-4 video coding syntax supports three VOP coding modes, illustrated in Figure 2.6:

- **I-VOP mode** – The complete VOP (Intra VOP or I-VOP) is coded without reference to other VOPs.
- **P-VOP mode** – The VOP (Predicted VOP or P-VOP) can be coded using predictions, e.g., motion compensated predictions, from previous decoded VOPs.
- **B-VOP mode** – The VOP (Bidirectional VOP or B-VOP) can be coded using predictions from previous and/or future (in terms of presentation) VOPs.

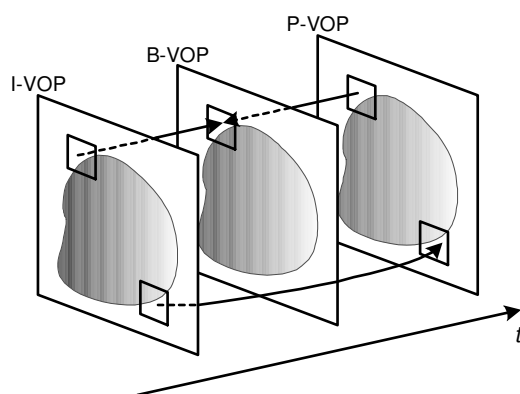


Figure 2.6 – MPEG-4 VOP coding modes: Intra (I), Predicted (P), and Bidirectional (B)

⁵ A tool, in MPEG-4 terminology, is a technique that enables one or more MPEG-4 functionalities (e.g., motion compensation). Tools may, themselves, consist of tools [53].

⁶ An algorithm, in MPEG-4 terminology, is an organized collection of tools that fulfils one or more requirements (e.g., DCT image coding). Algorithms may, themselves, be composed of tools and/or algorithms [53].

Figure 2.7 shows the generic block diagram of an MPEG-4 VOP encoder. This diagram includes the following major modules:

- **Discrete Cosine Transform (DCT)** – The DCT is used to transform either the luminance and chrominance samples (Intra mode) or the prediction error (Inter mode) from blocks of 8×8 samples into blocks of 8×8 coefficients:
 - **Intra mode** – The DCT is applied to the YUV samples of the given VOP, i.e., the texture data are coded independently of previous or future VOPs.
 - **Inter mode** – The DCT is applied to the prediction error, i.e., to the difference between the YUV samples and the corresponding predictions obtained from previous and/or future VOPs. In this case the texture data are coded differentially.
- **Quantization** – To reduce the number of bits used to encode the DCT coefficients, two quantization modes are available in the MPEG-4 video coding syntax:
 - **Method 1** – This method is derived from the coefficient quantization method used in the MPEG-2 Video standard [10]. The quantization step for each DCT coefficient in the block (except the Intra DC coefficients) is obtained from a quantization weighting matrix and the quantization parameter of the corresponding MB.
 - **Method 2** – This method is derived from the coefficient quantization method used in the ITU-T H.263 standard [8]. The quantization step is the same for all DCT coefficients (except the Intra DC coefficients).
- **Entropy Coding** – The quantized DCT coefficients, coding mode data, motion vectors data, and some shape data, are coded using variable length coding (VLC) using different VLC tables according the different types of data and coding modes.
- **Motion Estimation/Motion Compensation** – To generate the motion-compensated prediction for each input VOP, motion estimation is performed between the input VOP and the previous or future reconstructed VOPs located in the reference memory. The MPEG-4 video syntax supports several motion compensation tools (see Section 2.4). Essentially two motion modes are available for each MB:
 - **1MV mode** – One motion vector (MV) is transmitted for each MB.
 - **4MV mode** – The MB is divided into four 8×8 luminance blocks, and one MV is transmitted for each of these blocks.

Notice that, the motion estimation procedure is not defined normatively since it does not impact on interoperability, providing that the syntax and semantics of the motion information are respected.

- **Shape Coding** – For arbitrarily shaped video objects, it is necessary to encode the shape data. Similarly to the YUV samples, the shape samples can also be encoded in Intra mode or Inter mode. The resulting symbols are then entropy coded.
- **Rate Control** – This module is essentially responsible for controlling the VOP encoder bit production (but not only). Although not a normative module, it is essential to the efficient and standard compliant behavior of the VOP encoder. This module can receive input from various modules and provides decisions that aim at efficiently coding the input data. Since this Thesis is mainly concerned with this problem, a more detailed analysis of this mechanism is provided later, as well as innovative proposals.

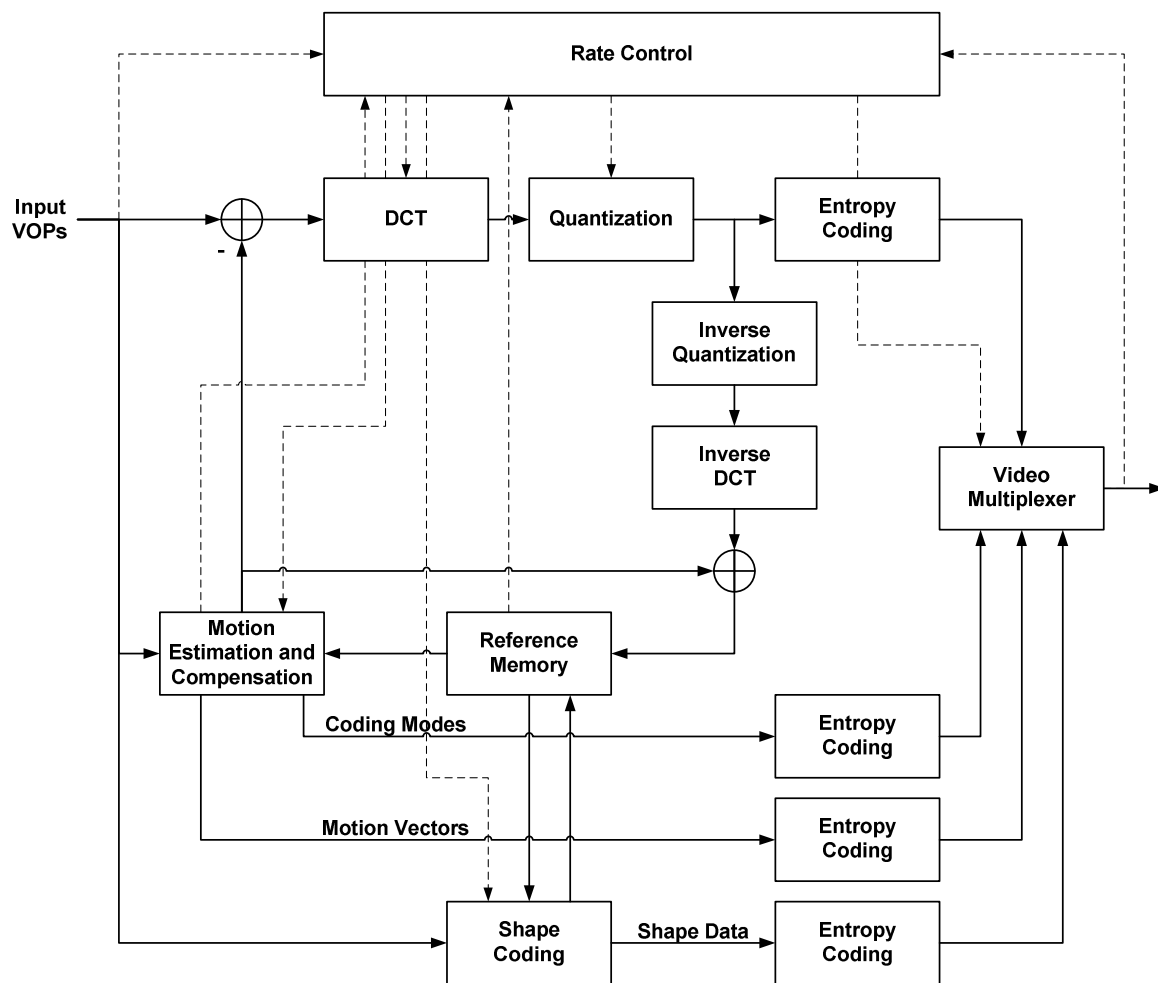


Figure 2.7 – MPEG-4 VOP encoder block diagram

2.4 MPEG-4 Video Coding Tools

As mentioned previously, the main MPEG-4 functionalities are the content-based interactivity, compression efficiency, and universal access. The video coding tools that support this type of functionalities can be grouped into: shape coding, motion estimation/compensation, texture coding, sprites, still texture coding, error resilience, scalability, reduced resolution coding, interlaced coding, and short video header. This section briefly describes these coding tools. A more detailed description can be found in [51] and [29].

2.4.1 Shape Coding

One of the main features of the MPEG-4 standard is, therefore, the capability of coding arbitrarily shaped video objects. Consequently, shape coding tools are required to support this functionality. With respect to its shape data (alpha plane), VOs can be classified in three types: 1) opaque, 2) with constant transparency, and 3) with variable transparency [66]. Therefore, to independently encode a VO, it is necessary to encode its alpha plane, notably the support of the alpha plane (i.e., the binary data) and the transparency information (i.e., the multi-level data). While for binary shapes only the alpha plane support has to be coded, gray-level shapes require also the coding of transparency information. In MPEG-4, transparency

information is encoded either with a single transparency value or with techniques very similar to those used for the luminance information, i.e., using DCT + quantization + entropy coding [29]. The efficient encoding of the shape information has required the development of very sophisticated techniques, during a process where many technical proposals were carefully evaluated [67]. During the MPEG-4 standardization process, two main families of binary shape coding approaches have been proposed: bitmap-based and contour-based techniques. Bitmap-based techniques directly encode the shape pixels as belonging to the object or to the background. Two main methods have been presented in this area: the modified modified Reed [68] and context-based arithmetic encoding [69]. Contour-based techniques represent the shape by its boundary. Three different approaches have been proposed relying on the contour representation: vertex-based [70], baseline-based [71], and chain-code-based [72] shape encoders. Due to its robustness and superior performance for block-based coding, the CAE technique has been adopted by MPEG.

To code the binary shape of a given VOP with CAE, the alpha plane confined in the VOP bounding box is divided in blocks of 16×16 samples called binary alpha blocks (BABs) and is encoded in a raster scan order. There are three different categories of BABs, following the MB classification inside the VOP:

- **Transparent** – All BAB alpha values are 0.
- **Opaque** – All BAB alpha values are 255.
- **Boundary** – BAB alpha values are either 0 or 255.

While transparent and opaque BABs are encoded by a single word describing the BAB type, boundary BABs may need further data, besides the BAB type, to completely encode all the shape pixels, notably motion information and context data.

BINARY ALPHA BLOCK TYPE ENCODING

The method used to encode the BAB is indicated by the BAB type. Each BAB can be encoded with one of seven different types listed in Table 2.1, where MVDS is the motion vector difference for shape, i.e., the difference between the actual motion vector and the motion vector predictor for the current BAB, and NO UPDATE means that the current BAB can be completely reconstructed by motion compensation without any further information needed.

Table 2.1 – List of BAB types for CAE

BAB Type	Semantic	VOP Type
0	MVDS == 0 && NO UPDATE	P-, B- VOPs
1	MVDS != 0 && NO UPDATE	P-, B- VOPs
2	TRANSPARENT	All VOP Types
3	OPAQUE	All VOP Types
4	INTRA CAE	All VOP Types
5	MVDS == 0 && INTER CAE	P-, B- VOPs
6	MVDS != 0 && INTER CAE	P-, B- VOPs

Intra VOPs

For intra coded VOPs, the number of BAB types allowed is restricted to three out of the seven different values available – 2, 3, and 4 – encoded using a VLC table indexed by a BAB type context C . For each BAB, the context is computed based on the values of the already encoded neighboring BAB types, and the BAB type of the BAB being encoded. Figure 2.8 shows the template used to compute the context C using equation (2.1), where c_k is the type of the k -th BAB in the template.

$$C = \sum_{k=0}^3 (c_k - 2) \cdot 3^k \quad (2.1)$$

If the computation of the context C involves BABs outside of the current VOP BB, the corresponding BAB types are assumed to be transparent.

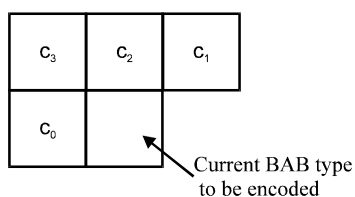


Figure 2.8 – Template for computing the BAB type context for I-VOPs

Inter VOPs

In the case of inter coded VOPs (P- and B-VOPs) all BAB types listed in Table 1 are allowed and the BAB type information is encoded using a different VLC table indexed by the co-located BAB type in the selected reference VOP and the BAB type of the BAB being encoded.

If the sizes of the current and reference VOPs are different, some BABs in the VOP being encoded may not have a co-located equivalent in the reference VOP. In this case the BAB type matrix of the reference VOP is manipulated in order to match the size of the current VOP [29].

BINARY ALPHA BLOCK MOTION COMPENSATION

Besides the type information, motion information can also be used to encode the BAB data. In this case the BAB is encoded in inter mode and one motion vector (MVs – motion vector for shape) for each BAB is used. Similarly to what happens in texture coding, in the MPEG-4 shape coding technique, the shape motion vectors are encoded differentially, which means that for each motion vector a predictor is built from neighboring shape motion vectors already encoded and the difference between this predictor and the actual MVs – the MVDs – is VLC encoded. When no valid shape motion vectors are available for prediction, texture motion vectors can be used.

BINARY ALPHA BLOCK SIZE CONVERSION

The compression ratio of the CAE technique can be increased by allowing the shape to be encoded in a lossy mode. This can be achieved by encoding a down-sampled version of the BAB. In this case, the BAB can be down-sampled by factor of 2 or 4 before context encoding and up-sampled by the same factor after context decoding. The relation between the original

and the down-sampled BAB sizes is called conversion ratio (CR). Table 2.2 presents the different BAB sizes allowed, depending on the value of the CR parameter. The error introduced by this down-sampling/up-sampling operation is responsible for the lossy encoding and is called the conversion error.

Encoding the shape using a high down-sampling factor provides a better compression ratio; however, the reconstructed shape after up-sampling may exhibit some annoying artifacts. In order to minimize these artifacts, an adaptive non-linear up-sampling filter has been adopted by the MPEG-4 Visual standard [29].

Table 2.2 – List of BAB sizes for CAE

CR	BAB Size in Pixels
1	16×16
2	8×8
4	4×4

CONTEXT-BASED ARITHMETIC ENCODING

For boundary blocks which are not encoded by motion information only, context-based arithmetic encoding is applied. For these blocks, the BAB data is encoded by a single binary arithmetic codeword (BAC). In this case, each binary shape pixel in the BAB (or in its down-sampled version) is encoded in a raster scanning order (until all pixels are encoded). The process of encoding a given shape pixel using CAE is the following:

1. Compute a shape context number based on a template.
2. Use this context number to index a probability table.
3. Use the indexed probability to drive a binary arithmetic encoder.

Since the encoding performance depends on the scanning order of the BAB, the encoder is allowed to transpose (matrix transposition) the BAB before encoding. This operation is signaled to the decoder through a 1 bit flag called scanning type.

Context Computation

The core of the CAE technique is the exploitation of the spatio-temporal redundancy in the motion-compensated BAB data using causal contexts to predict the shape pixels according to pre-defined templates. Figure 2.9 shows the different templates used in CAE encoding.

For intra coded BABs, no motion compensation is used, thus only the spatial redundancy is exploited by computing a 10 bit context for each pixel of the BAB, using equation (2.2) with $N=9$, according to the template of Figure 2.9a, where $c_k=0$ for transparent pixels and $c_k=1$ for opaque pixels.

$$C = \sum_{k=0}^N c_k \cdot 2^k \quad (2.2)$$

Temporal redundancy can additionally be exploited for inter coded BABs by computing a 9 bit context using also pixels from the motion compensated BAB (besides the pixels from the BAB being encoded). In this case the context is computed using equation (2.2) with $N=8$, according to the template of Figure 2.9b. As in the intra case, $c_k=0$ for transparent pixels and $c_k=1$ for opaque pixels.

When building contexts, some special cases may appear, notably some pixels may be out of the current VOP BB or the template may cover pixels from BABs that are not known at decoding time. These cases have special pre-defined rules that are known both by the encoder and the decoder [29].

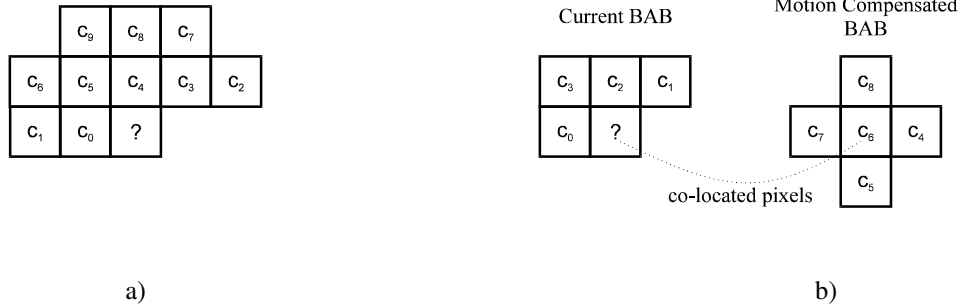


Figure 2.9 – CAE templates for context computation ('?' represents the pixel being encoded):
a) Intra-coded BABs; b) Inter-coded BABs

2.4.2 Motion Estimation and Compensation

As in previous MPEG standards, motion compensation aims at exploring the temporal redundancy between VOPs of a given VO. Therefore, motion compensation in MPEG-4 is very similar to previous MPEG standards (with the necessary adaptations to deal with arbitrarily shaped VOs) except for the new motion compensation tools introduced by MPEG-4 (described below), notably: quarter-pel motion compensation, global motion compensation (GMC), and direct-mode in bi-directional predictions.

As mentioned above, the MPEG-4 syntax allows sending one or four MVs per MB. The method for deciding between these two possibilities (if any) and to compute the MVs is not normatively specified, since it is not required for ensuring interoperability. However, the MPEG-4 video verification model (VM) [73] describes the method used in the MPEG-4 video reference software [32], which is a modified block-matching⁷ algorithm.

FULL-PIXEL MOTION ESTIMATION

For each non-transparent luminance MB in the VOP to be encoded, the motion estimation algorithm computes the sum of absolute differences (SAD) between the original MB and the displaced MB in the reference VOP (i.e., the previous I- or P-VOP in the case of P-VOPs, and the previous of future I- or P-VOP in the case of B-VOPs) over a given displacement area. Then, the smallest $SAD_{16}(x, y)$ for all tested displacements is recorded as SAD_{16} , where $V_0 = (x, y)$ represents the motion vector that corresponds to a horizontal displacement of x pixels and a vertical displacement of y pixels with respect to the MB in the original VOP.

In order to favor the (0,0) motion vector, since in this case no motion vector data needs to be sent to the decoder, the $SAD_{16}(0,0)$ is reduced by attributing it a new value as follows

⁷ For arbitrarily shaped VOs, notably for the boundary MBs, the block matching technique is called polygon matching since only the original pixels inside the MB (i.e., with a nonzero alpha value) are used to compute the motion estimation prediction error.

$$SAD_{16}(0,0) = SAD_{16}(0,0) - (N_B/2 + 1) \quad (2.3)$$

where $N_B = \#\{\text{non-transparent MB pixels}\} \times 2^{(\text{bits_per_pixel}-8)}$.

In order to reduce the motion estimation computational complexity and to avoid large discrepancies between neighboring MB MVs⁸, the search for the best 8×8 MVs is only performed with a window of ± 2 pixels around V_0 . The (x, y) displacements leading to the lowest $SAD_8^k(x, y)$, $k=1, \dots, 4$, define the 4 MVs, V_1 , V_2 , V_3 , and V_4 . The MB 8×8 SAD, $SAD_{K \times 8}$, is the sum of $SAD_8^k(x, y)$ for all 8×8 blocks that have at least a non-transparent pixel.

Therefore, the SAD for the Inter mode, SAD_{inter} , is computed as follows

$$SAD_{\text{inter}} = \min[SAD_{16}, SAD_{K \times 8}] \quad (2.4)$$

INTRA/INTER MODE DECISION

After full-pixel motion estimation the encoder takes a decision, for each MB, on whether to code the referred MB using the Intra mode or Inter mode. For this purpose the SAD for the Intra mode, SAD_{intra} , is computed as the sum of absolute differences between the MB luminance samples and the MB average luminance samples, including only the non-transparent MB pixels.

The Intra mode is chosen if

$$SAD_{\text{intra}} < SAD_{\text{inter}} - 2N_B \quad (2.5)$$

otherwise the Inter mode is chosen and a further motion estimation step is performed to refine the MVs with half-pixel accuracy.

HALF-PIXEL MOTION ESTIMATION

Half-pixel motion estimation is performed for the 16×16 and the 8×8 MVs, only around the full-pixel MVs, i.e., with a search area of ± 1 half sample around the full-pixel MVs V_0 , V_1 , V_2 , V_3 , and V_4 . The half-pixel samples, located between the full-pixel positions (on the reference reconstructed VOP) are obtained through bilinear interpolation (see Figure 2.10).

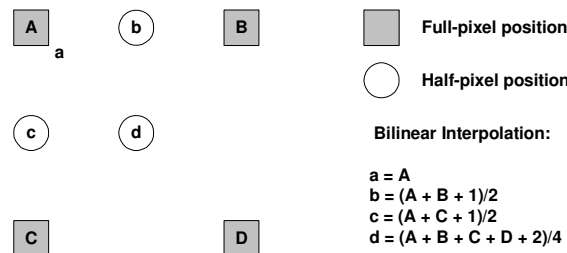


Figure 2.10 – Sample interpolation for half-pixel MVs

The MVs resulting in the best matching during half-pixel motion estimation are recorded, as well as the new values of SAD_{16} and SAD_8^k , $k=1, \dots, 4$, for taking a decision on whether to

⁸ Since in MPEG-4 the MVs are coded differentially, a more homogeneous MV field requires, typically, less bits to be coded.

use 1MV or 4MV Inter mode.

MOTION-PREDICTION MODE DECISION

Using the new SAD_{16} and $SAD_{K \times 8}$ values obtained during the half-pixel motion estimation, the 4MV mode is chosen if

$$SAD_{K \times 8} < SAD_{16} - (N_B/2 + 1) \quad (2.6)$$

otherwise the 1MV mode is chosen. Notice that since choosing the 4MV mode requires sending more motion information (four MVs instead of one), the 4MV mode is penalized, relatively to the 1MV mode, by $(N_B/2 + 1)$ as expressed in equation (2.6).

DIFFERENTIAL CODING OF MOTION VECTORS

When the Inter mode is selected for each $MV = (MV_x, MV_y)$ the components MV_x and MV_y are coded differentially, as mentioned above, using a spatial neighborhood of three motion vectors already transmitted (see Figure 2.11 and Figure 2.12). The motion vector prediction $PMV = (PMV_x, PMV_y)$ is computed as

$$\begin{aligned} PMV_x &= \text{median}(MV1_x, MV2_x, MV3_x) \\ PMV_y &= \text{median}(MV1_y, MV2_y, MV3_y) \end{aligned} \quad (2.7)$$

The motion vector difference (MVD) is computed according to (2.8) and coded using VLC codes.

$$\begin{aligned} MVD_x &= MV_x - PMV_x \\ MVD_y &= MV_y - PMV_y \end{aligned} \quad (2.8)$$

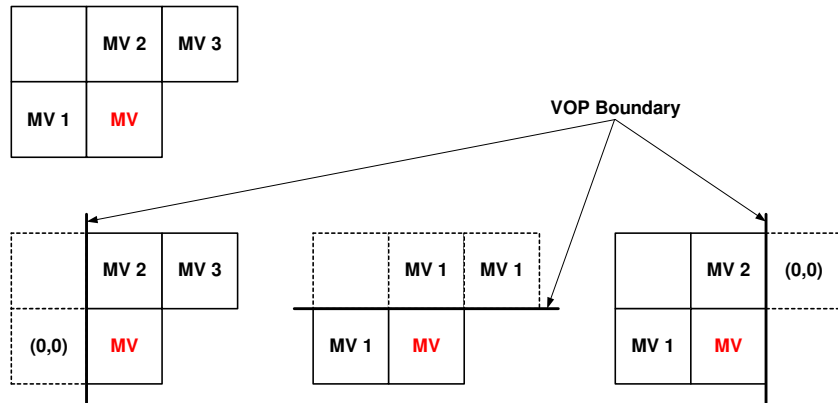


Figure 2.11 – Motion vector prediction for the 1MV mode

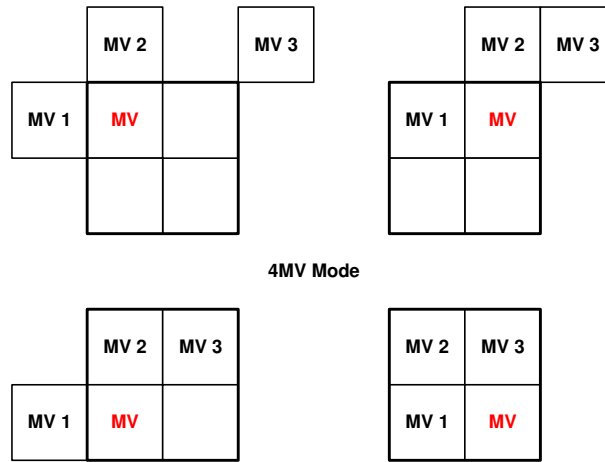


Figure 2.12 – Motion vector prediction for the 4MV mode

PADDING OF BOUNDARY AND TRANSPARENT MBs

In the case of arbitrarily shaped VOs, the reference VOPs are also of arbitrary shape. Therefore, it is possible that MVs refer to pixels outside the reference VOP (i.e., transparent pixels). In order to avoid this, the reference VOP is generated from the previously decoded VOP by a padding process (see Figure 2.13). During this process, all transparent pixels in the BB of the reference VOP are attributed YUV values that are derived from the non-transparent pixels in the same VOP. Boundary MBs and transparent MBs are differently padded: for boundary MBs the process is called repetitive padding, while for non-transparent MBs the process is called extended padding [29]. Notice, that this padding process is normative since the encoder and the decoder must generate identical reference VOPs.

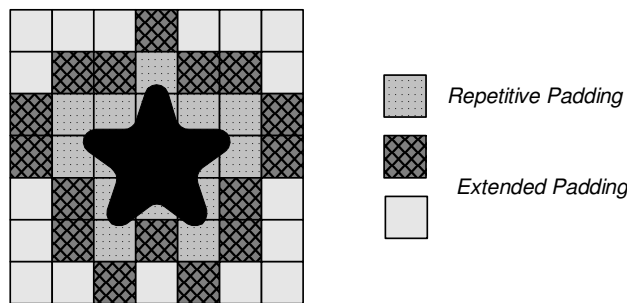


Figure 2.13 – Padding process for motion estimation and compensation

NEW MOTION COMPENSATION TOOLS

Besides the traditional motion compensation tools, with the necessary adaptations for dealing with arbitrarily shaped VOs, the MPEG-4 Visual standard [29] introduces the following new and improved tools and algorithms:

- **Quarter-pel motion compensation** – A new algorithm for motion compensation using MVs with an increased resolution of one-quarter pixel instead of only half- or full-pixel resolution, as described above. This tool allows improving the prediction by decreasing the prediction error without increasing the bit rate for the motion information. This resulted mainly from the new sample interpolation process introduced in MPEG-4.

- **Global motion compensation** – A single set of motion parameters is coded for the complete VOP, representing the VOP global motion. These parameters can be used alternatively to the (local) MVs of a MB for motion compensation. In this case, a different MV (interpolated MV) is computed for each pixel of the MB using the VOP global MVs (see Figure 2.14). This tool is particularly important for sequences with a large portion of global motion (e.g., caused by a moving camera) and for non-translational motion like zoom or rotation.
- **Direct mode in bidirectional prediction** – This mode uses the MVs of neighboring P-VOPs in order to decrease the bit rate required for coding the B-VOP MVs. This is an improvement of the bidirectional motion-compensated prediction. It is a generalization of the “PB frames” introduced by H.263 [8].

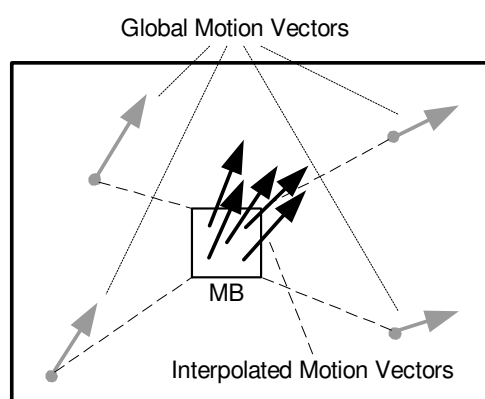


Figure 2.14 – Global motion compensation

2.4.3 Texture Coding

As already mentioned, in MPEG-4 the DCT, applied to blocks of 8×8 samples, is used for coding the VOP texture data, as well as the transparency data of gray-level alpha planes. For the texture data, the DCT is applied, separately to the luminance and chrominance components of each MB. As for shape coding, there are three different types of MBs (see Figure 2.5):

- **Transparent** – All MB pixels are outside the VOP (though inside the VOP BB). Therefore, no texture data is available for them and consequently no DCT data is coded.
- **Opaque** – All MB pixels are inside the VOP. In this case the usual DCT is applied.
- **Boundary** – These MBs have both transparent and non-transparent pixels. Therefore, it is necessary to take into account each 8×8 block case: for the 8×8 blocks with only non-transparent pixels, the usual DCT is applied; for the 8×8 blocks with only non-transparent pixels, no DCT is applied; and finally, for the remaining blocks the transparent pixels are padded before applying the usual DCT.

The padding process for texture coding is not normatively specified, since the decoded pixel data outside the VOP should be discarded and is not used for prediction of other VOPs. In the MPEG-4 video VM [73], Inter coded 8×8 boundary blocks are padded with zero value samples, while Intra coded 8×8 boundary blocks are padded using the low pass extrapolation (LPE) padding technique.

The DCT coefficients are quantized using one of the allowable quantization methods and zigzag scanned in order to produce a vector of coded symbols that favor the transmission of the more relevant coefficients. These three-dimensional symbols (LAST, RUN, LEVEL) are then bit encoded using VLC codes for the more frequent symbols.

NEW TEXTURE CODING TOOLS

Besides the traditional DCT-based texture coding tool, with the necessary adaptations for dealing with arbitrarily shaped VOs, the MPEG-4 Visual standard [29] introduces the following new and improved tools and algorithms:

- **Nonlinear Intra DC coefficient quantization method** – The DC coefficients of Intra coded MBs are quantized using an optimized, nonlinear quantization method, where the value of the quantization step for these coefficients, dc_scaler , depends on the value of the corresponding MB quantization parameter, QP (see Table 2.3).

Table 2.3 – Quantization step for the Intra DC coefficients

QP	1 – 4	5 – 8	9 – 24	25 – 31
dc_scaler (luminance)	8	$2QP$	$QP + 8$	$2QP - 16$
dc_scaler (chrominance)	8	$(QP + 13)/2$	$(QP + 13)/2$	$QP - 6$

- **AC/DC prediction for Intra MBs** – The AC and DC coefficients of neighboring blocks usually exhibit some statistical dependencies and, therefore, the MPEG-4 Visual standard [29] allows the value of some of these coefficients in a given block to be predicted from the corresponding value of one of the neighboring blocks.
- **Alternative scan modes** – In MPEG-4 Visual [29], the scanning process used to convert the two-dimensional DCT coefficient matrix into a one-dimensional vector can use two additional scanning modes besides the well know zigzag scanning mode. Although this technique was already available in MPEG-2 Video [10], the zigzag scan was almost always used.
- **N-Bit coding tool** – In extension to the 8-bit pixel amplitude resolution used in MPEG-2 Video [10], MPEG-4 Visual [29] supports the use of 4 to 12 bits per sample for the luminance and chrominance components. This is signaled to the decoder in the VOL header. The coding of gray-level alpha values is always done with 8 bits.
- **Shape-Adaptive DCT (SA-DCT)** – This tool allows an arbitrary region of an 8×8 block within a boundary BAB to be efficiently transformed and coded using the BAB shape information and a pre-defined set of one-dimensional DCT basis functions.

TEXTURE CODING FOR HIGH QUALITY APPLICATIONS

For high-quality applications, such as video in the studio, special coding tools were added to the MPEG-4 Visual specification [29]; these tools are used in the Simple and Core studio object types (see Section 2.6.3):

- Higher DCT precision (3 additional bits), allowing up to lossless coding.
- Uncompressed (PCM) coding mode as a fallback mode to avoid data expansion due to

the high precision DCT coefficient representation.

- 4:2:2 and 4:4:4 chrominance sampling modes (including the corresponding adaptation of the chrominance padding for arbitrarily shaped VOs).
- Independent chrominance quantization weighting matrices⁹ for the 4:2:2 and 4:4:4 chrominance sampling modes.
- Low complexity (PCM) mode for binary shape coding (i.e., without CAE).
- Extension of the gray-level alpha resolution to 4-12 bits.
- New sprite coding (see Section 2.4.4) parameters.
- Additional display control parameters, e.g., for pan-scan applications.
- Inclusion of the MPEG-2 Video 4:2:2 Profile [10] header extensions.
- MPEG-2 compatible MV coding/motion compensation to ease MPEG-2 Video to MPEG-4 Visual transcoding.

2.4.4 Sprite Coding

Sprite coding is a high coding efficiency tool specially suited for the object-based representation and coding of video sequences. It is based on the use of a long-term memory associated to one specific object in the scene that contains all pixel information of such object that can be visible along the sequence. The typical example of a sprite is a background sprite, which contains the pixels belonging to the background of a scene during, for example, a camera panning. This way, the background sprite can be used for direct reconstruction or predictive coding of the background. In this case, the reconstructed scene for each presentation time instant is obtained by composing the visible portion of the background (for that time instant) with the foreground objects, as illustrated in Figure 2.15.

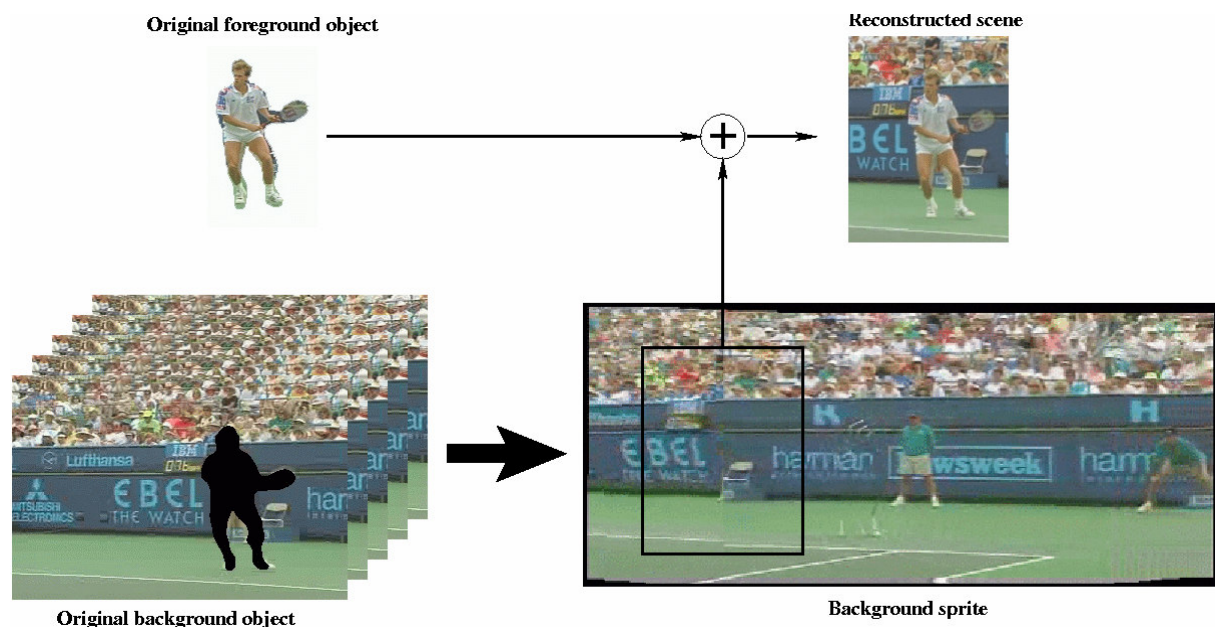


Figure 2.15 – Reconstruction of a scene using its background sprite [51]

⁹ For the quantization method 1.

In MPEG-4 Visual [29], sprites are static, i.e., they are built off-line and sent all at once to the receiver at the beginning (basic sprites) or sent piece by piece along the video transmission (low-latency sprites). Fully dynamic sprites, i.e., sprites that are built on-line, are not supported by MPEG-4. However, MPEG-4 supports a simplified form of dynamic sprites, called global motion compensation (see Section 2.4.2). To represent a VOP that is coded using the sprite coding tool, a new VOP coding type, named Sprite VOP (S-VOP) is defined.

Notice, that, similarly to other coding tools, the procedure to generate the sprites is not standardized by MPEG-4, because it does not impact interoperability.

2.4.5 Still Texture Coding

As mentioned above, the MPEG-4 Visual standard [29] supports the mapping of static textures in 2D and 3D animated meshes specifying an efficient tool for coding such textures, known also as visual texture coding (VTC). This tool uses an efficient compression algorithm based on the discrete wavelet transform (DWT) [29, 74] (also adopted by JPEG2000 [75]) where the texture components (luminance and chrominance) are separately decomposed in several subbands using recursive filtering.

Due to this subband decomposition, the VTC tool allows fine levels of spatial scalability, notably in comparison with those that can be achieved with DCT-based techniques. This is the main reason for including an additional transform in the MPEG-4 Visual standard [29] besides the DCT. The MPEG-4 VTC tool adopts the following core tools:

- Wavelet and zero-tree-based compression algorithm to achieve efficient compression.
- Shape-adaptive wavelet coding for compressing arbitrarily shaped still texture objects.
- Three quantization schemes and two wavelet coefficient scanning modes to provide different granularity of spatial and quality scalability for both rectangular and arbitrarily shaped still texture objects.
- Bitstream packetization approach to support error robustness in error-prone environments, where the bitstream data is organized in packets separated by resynchronization markers and sensitive data is specially grouped.
- Tiling scheme to reduce encoding/decoding memory requirements and provide random access, where a large image can be divided into several subimages that are independently coded using the VTC tool.

A more detailed description of these core tools can be found in [51, 29].

2.4.6 Error Resilience

Error resilient coding is a fundamental tool to provide robust video transmission over error-prone environments and, therefore, to support universal access, notably, over mobile and wireless networks. The MPEG-4 Visual standard [29] specifies five main error resilience tools: 1) packet-based periodic resynchronization, 2) data partitioning (DP), 3) reversible variable length codes (RVLCs), 4) header extension code (HEC), and 5) new prediction (NEWPRED). A more general analysis of error resilient coding for object-based video can be found in [76].

PACKET-BASED RESYNCHRONIZATION

The use of VLCs makes compressed bitstreams especially sensitive to channel errors.

Therefore, when errors occur during transmission, the decoder may lose the syntactic synchronization with the encoder and the bitstream is no longer decodable unless adequate measures are taken¹⁰. This problem can be solved introducing resynchronization markers in the bitstream. These markers must be unique codes (i.e., a sequence of bits that cannot be emulated by any code, or combination of codes, used by the encoder). When an error is detected¹¹, the decoder may recover from this error by searching the bitstream for the next resynchronization marker and continue normal decoding from this point onwards (see Figure 2.16).

The MPEG-4 standard supports this approach allowing a given encoder to divide the video bitstream into video packets of an integer (but not fixed) number of MBs and to insert a resynchronization marker at the beginning of each video packet. The MPEG-4 approach is different from the H.261 [7] and H.263 [8] approaches that support resynchronization after an integer number of MBs.

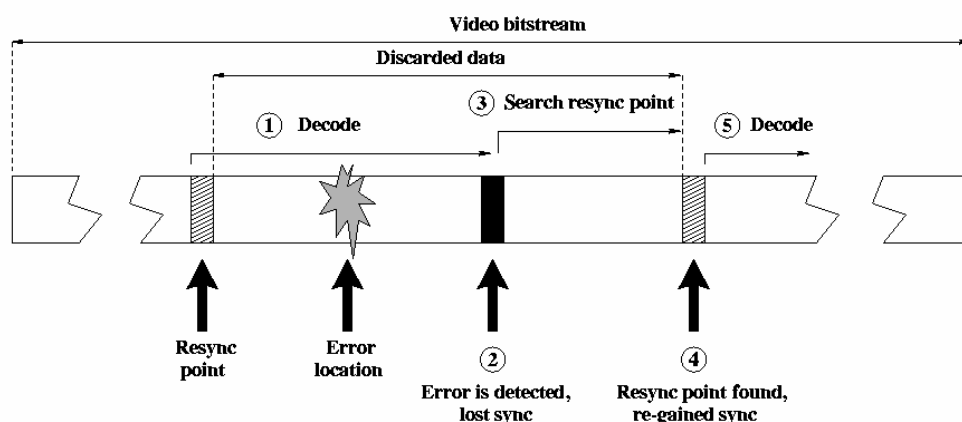


Figure 2.16 – Decoder resynchronization following error detection [51]

DATA PARTITIONING

After detecting an error in the bitstream and resynchronizing to the next resynchronization marker, the decoder has to decide what to do with the MB data located between the two resynchronization markers. Typically this data is discarded since: 1) the decoder can hardly know the exact location of the error; 2) even if the extension of corrupted bits can be identified, the decoding of the uncorrupted data will almost for sure require information that is conveyed by the corrupted bits.

Since the error concealment process¹² is not normatively specified, each decoder implementation can use its own concealment technique. A simple error concealment technique consists in replacing the non-decoded MBs (due to the errors) with MBs from the previous VOP. MPEG-4 data partitioning technique separates the MB data (e.g., between texture data – DCT coefficients – and motion data – MVs) using special markers. These

¹⁰ Notice, that all video coding standards specify a decoder behavior assuming error-free bitstreams, consequently the decoder behavior under error conditions is not normatively specified.

¹¹ The procedure for detecting errors is not normatively specified. Therefore, more or less sophisticated techniques can be used for this purpose.

¹² The process that aims at reducing the negative subjective impact of the decoding errors.

markers allow a more precise localization of the errors and, consequently, reduce the amount of data that has to be discarded and allow improving the error concealment. For example, if an error is detected in the motion data, the decoder should usually replace the affected MBs by the corresponding MBs in the previous VOP; however, if an error is detected in the texture data, the decoder can still use the motion vectors to obtain better predictions for the non-decoded MBs from the previous decoded VOP.

REVERSIBLE VARIABLE-LENGTH CODING

Reversible variable length coding (RVLC) enables the decoder to better isolate the errors, thus improving data recovery, since these codes can be uniquely decoded both in the forward and reverse directions.

When the decoder detects an error, while decoding the bitstream in the forward direction, it can search for the next resynchronization marker and from there decode the bitstream backwards until it encounters a new error (see Figure 2.17). Based on the location of the two errors, the decoder can recover some of the data that otherwise, i.e., using traditional VLCs, would have to be discarded¹³.

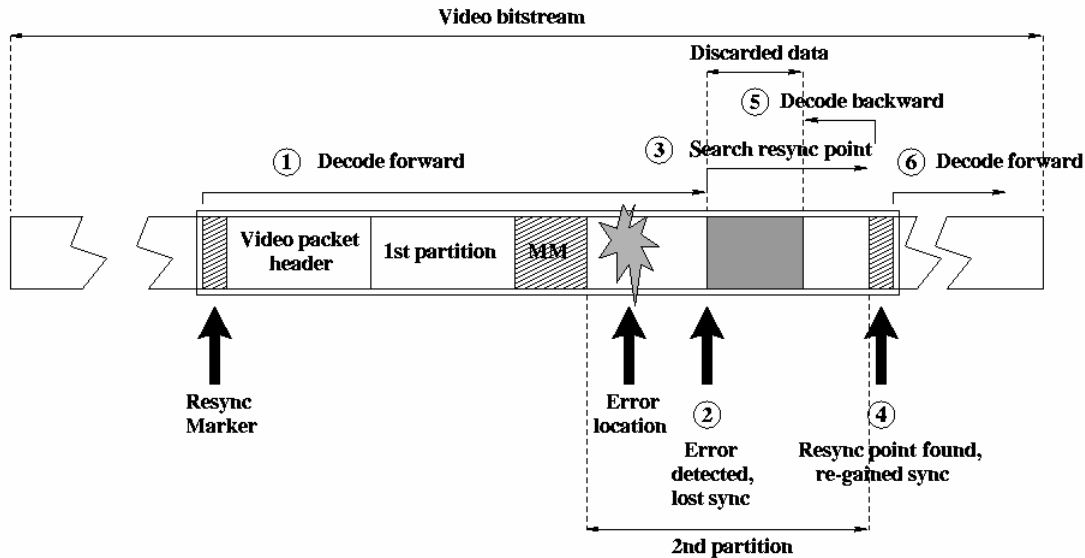


Figure 2.17 – Data recovery with RVLC [51]

HEADER EXTENSION CODE

Headers, e.g., VOP headers, contain very sensitive information for the decoding process, such as information about the VOP spatial dimensions, the temporal information associated with its decoding and presentation, and its coding mode (Intra or Inter). If some of this information is corrupted by errors, the decoder is no longer able to decode the VOP and the only option is to discard the entire VOP.

A simple approach to improve data recovery in these situations is to replicate part of the header data at other locations in the bitstream. This approach has been proved successful in

¹³ Notice, that the procedure for decoding RVLCs in the presence of errors is not normatively specified.

preserving sensitive information, provided that the amount of extra redundant information is not too high.

MPEG-4 supports this approach through the header extension code (HEC). The HEC is a 1-bit code used in video packets after the video packet header that when set to 1 indicates it is followed by the duplicated VOP header information (see Figure 2.18). When the decoder detects an error in the VOP header¹⁴ it can search the bitstream for the next resynchronization marker and the HEC bit set to 1, in order to recover the uncorrupted VOP header that allows decoding part of the VOP data.

Notice that there is a trade-off between the overhead introduced when using HEC and the error recovery capability. Consequently, the MPEG-4 standard does not specify how frequently this tool can be used, being an encoder issue the decision on how and when to use this tool. Moreover, as for the other MPEG-4 error resilience tools it is not specified how the decoder should use HEC-duplicated information in the presence of errors.

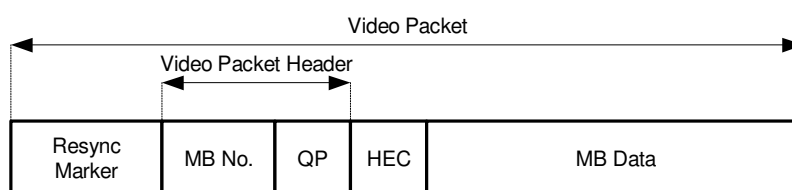


Figure 2.18 – MPEG-4 video packet with header extension code

NEW PREDICTION

One of the main problems of using the Inter coding mode in error-prone environments is the propagation of errors due to the temporal predictions. The error propagation can typically be stopped by periodically or adaptively performing Intra refreshments. The main problem with these approaches is that they do not adapt well to error condition changes during transmission, notably, either provide slow error recovery or they significantly degrade the coding efficiency.

To improve error recovery, MPEG-4 allows the decoder to send back to the encoder, via a back channel, information about the successful/unsuccessful decoding of Inter-coded VOPs (or video packets) and in case of errors the encoder is able to change the prediction of future VOPs to previously successful decoded VOPs.

This tool is called new prediction (NEWPRED). When NEWPRED is used additional memory is required to store several previous decoded VOPs and a trade-off between error-recovery and coding efficiency has to be set, since using older references leads typically to higher prediction errors and consequently a higher number of bits is required to keep the same quality.

2.4.7 Scalability

In scalable video coding the video signal is represented in the coded domain by a base layer that is independently coded and one or more enhancement layers coded relatively to previous lower layers. Typically, a lower resolution/quality signal is obtained if only the base layer is

¹⁴ The decoder can detect an error in the VOP header if the decoded header information is syntactically or semantically inconsistent.

decoded. To fully decode the video signal at its maximum resolution(s) and quality all enhancement layers besides the base layer are required. Therefore, scalability tools provide simultaneously several temporal/spatial resolutions and quality levels for the same content.

Besides object-based scalability, which is inherent to the object-based representation model adopted by MPEG-4, the MPEG-4 Visual standard [29] offers three additional natural video coding scalable tools: spatial scalability, temporal scalability, and SNR fine granularity scalability (FGS)¹⁵.

TEMPORAL SCALABILITY

Temporal scalability allows the decoder to increase the temporal resolution of a given scene or single VO by combining the base layer VOPs with those from the enhancement layer, i.e., the enhancement layer adds new information to be presented between the base layer VOPs (see Figure 2.19). MPEG-4 temporal scalability supports both rectangular and arbitrarily shaped video objects.

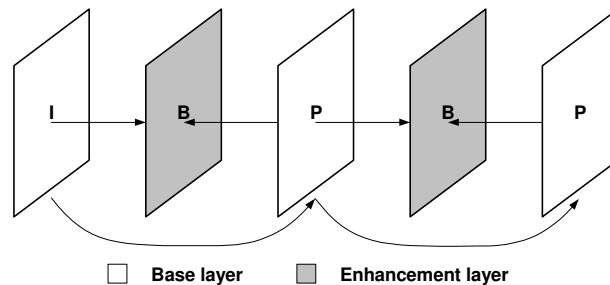


Figure 2.19 – MPEG-4 temporal scalability

SPATIAL SCALABILITY

Spatial scalability allows the decoder to increase the spatial resolution of the decoded video combining the enhancement layer with the reference base layer. Figure 2.20 shows the different types of prediction that can be used in spatial enhancement layers: I-VOPs that are coded independently from base and enhancement layer VOPs¹⁶; P-VOPs predicted from either I-VOP or P-VOPs of the base layer; and B-VOPs predicted simultaneously from VOPs of the base and enhancement layer¹⁷.

MPEG-4 spatial scalability supports both rectangular and arbitrarily shaped video objects. In the second case also a binary shape spatial scalability coding tool is provided, allowing to encode the binary shape of the enhancement layer with reference to the base layer shape.

¹⁵ These different scalability tools can be used independently or in combination, although not all combinations are supported by the MPEG-4 video syntax.

¹⁶ This is a major difference from MPEG-2 video [10], where I-VOPs in the enhancement layer are predicted from the co-located VOP in the base layer.

¹⁷ Notice, that in the base layer B-VOPs cannot be used as predictions to other VOPs; this is only allowed in the enhancement layers.

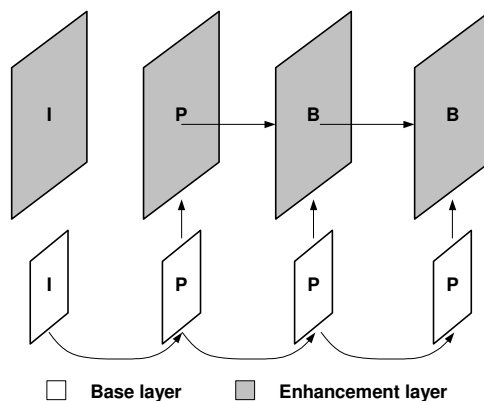


Figure 2.20 – MPEG-4 spatial scalability

FINE GRANULARITY SCALABILITY

The fine granularity scalability tool follows a different approach from traditional SNR or quality scalability tools. The main idea under this tool is to allow separation between the encoding and the distribution process. Therefore, the encoder produces, typically, one low bit rate (low quality) base layer bitstream and only one (rather large) enhancement layer bitstream, and it is the FGS server application that controls the amount of enhancement layer data that is sent to the channel, i.e., the FGS server truncates this bitstream according to the characteristics of the transmission channel (e.g., bit rate) or the decoder (e.g., computational power).

While in MPEG-2 Video [10], the requantized reconstruction error of the base layer is coded in the enhancement layer bitstream using the same quantization mechanisms and VLC tables as the base layer, in MPEG-4 FGS, the reconstruction error of the base layer is encoded in the enhancement layer using a bit plane representation of the DCT coefficients (see Figure 2.21). Additionally, in contrast to MPEG-2, only the reconstructed base layer data is used for temporal prediction in the enhancement layer, so that no drift of the prediction signals between the encoder and a base layer decoder may occur. MPEG-4 FGS only supports rectangular non-arbitrarily shaped video objects.

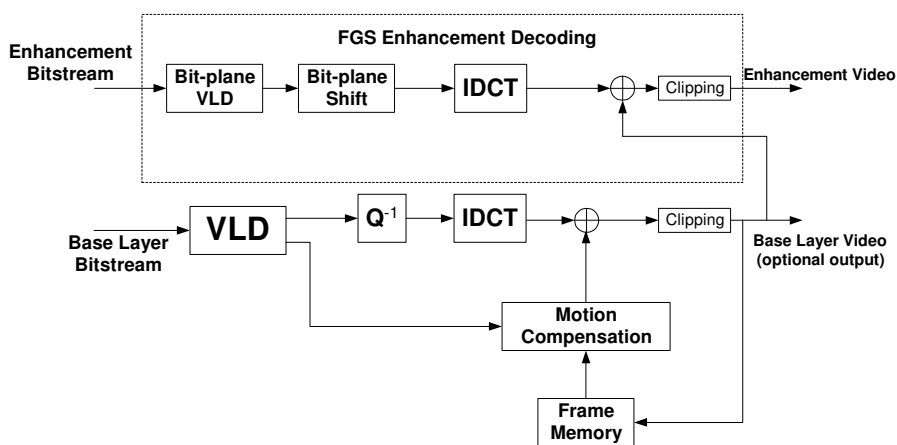


Figure 2.21 – Basic FGS decoder structure [51]

2.4.8 Reduced Resolution Video Coding

Reduced resolution (RR) coding, also known as dynamic resolution conversion (DRC), enables the encoder to encode a given VOP with reduced spatial resolution. This tool can be specially useful for low-delay low bit rate coding during scene changes, where the encoder to maintain approximately constant spatial quality would need to spend more bits than what is available due, for example, to buffer constraints. Therefore, instead of exchanging exclusively temporal smoothness for spatial quality, i.e., skipping several consecutive encoding time instants, the encoder may use the DRC tool to send to the decoder a lower spatial resolution¹⁸ encoded VOP (with less bits), which is useful to maintain motion smoothness without penalizing dramatically the spatial quality.

When a RR VOP is sent to the decoder, each MB in this VOP is decoded and up-sampled so that each 8×8 luminance block in the RR VOP covers a display area of 16×16 pixels. The up-sampled MB is then motion compensated using, now, a reference area of 32×32 samples for the luminance and 16×16 samples for each chrominance component. The MPEG-4 video syntax only allows the use of the RR tool for I- and P-VOPs.

Although RR VOPs have less detail than normal VOPs, they require few bits to be encoded and, therefore, allow the encoder to overcome localized bit rate control problems without dramatic spatial and temporal quality reductions.

2.4.9 Interlaced Video Coding

Because interlaced video is still widely used in TV broadcasting and, therefore, there is a considerable amount of this type of content, both in analog and digital formats, MPEG-4 has adopted a set of tools to support the compression of interlaced content.

In this context, MPEG-4 extended the coding of interlaced video to arbitrarily shaped interlaced VOPs. While some of these tools are extensions of the MPEG-2 interlaced video coding tools [10] (e.g., frame/field DCT, frame/field motion compensation, and interlaced direct mode), some MPEG-4 video coding tools have also been extended to support interlaced VOPs (e.g., field padding and AC/DC prediction).

2.4.10 Short Video Header

The short video header tool aims at providing compatibility between MPEG-4 Visual [29] and ITU-T H.263 video coding standard [8]. An I- or P-VOP encoded with the short video header mode¹⁹, has identical syntax to an I- or P-picture encoded with the baseline mode of ITU-T H.263 [8] (i.e., without the optional features include in the H.263 annexes [8]).

This means that the MPEG-4 Visual standard, when operating in the short video header mode, is backward compatible with H.263 baseline. And because all compliant MPEG-4 video decoders have to support the short video header mode, MPEG-4 is also forward compatible with H.263 baseline. However, due to the limited capability of signaling information in the short video header format, certain values of parameters present in the MPEG-4 Visual syntax have pre-defined and fixed values.

¹⁸ Half the normal horizontal and vertical resolutions.

¹⁹ This is signaled at the VOL header.

2.5 MPEG-4 Video Rate Control

Recognizing the importance of the rate control mechanism (although a non-normative tool) to achieve efficient video encoding, the MPEG-4 Visual standard [29] includes on its Annex L, an informative description of a rate control mechanism composed of three main algorithms:

1. Frame rate control.
2. Multiple video object rate control.
3. Macroblock rate control.

This section is devoted to these algorithms, pointing some of its limitations and open issues.

2.5.1 Frame Rate Control

The frame rate control algorithm can be applied to single video object (SVO) and independent multiple video object (MVO) encoding. It assumes that the encoder rate-quantization function can be modeled as

$$R(Q) = \left(X_1 \frac{1}{Q} + X_2 \frac{1}{Q^2} \right) \cdot E_c \quad (2.9)$$

where X_1 , and X_2 are the model parameters, and E_c is the encoding complexity expressed by the mean absolute difference (MAD) between the VOP being encoded and its reference.

The frame rate control algorithm is divided into four main steps described below.

STEP 1 – INITIALIZATION

This step consists in computing the number of bits left to encode the remaining VOPs of the given VO, R_r , as

$$R_r = T_s \cdot R_s - R_f \quad (2.10)$$

where T_s is the remaining time to encode, R_s is the target average bit rate, and R_f is the number of bits used to encode the first I-VOP of the VO. The first encoded VOP of each VO (I-VOP) is encoded with a constant quantization parameter ($Q=15$), while the remaining encoded VOPs are adaptively quantized with a quantization step computed by the algorithm.

Additionally, also the average number of bits to remove from the encoder buffer per encoding time instant is computed here as

$$R_p = \frac{R_r}{N_r} \quad (2.11)$$

where N_r is the remaining number of P-VOPs to encode (although the algorithm can be extended to encode also B-VOPs, it only contemplates P-VOPs)

STEP 2 – TARGET BIT ALLOCATION

I – Initial target estimation

Before encoding the current VOP, the target number of bits to encode the given VOP, T , is computed based on remaining available bits, R_r , and the number of bits used to encode the

previous VOP, S , as

$$T = \max \left[\frac{R_s}{30}, (1-\alpha) \frac{R_r}{N_r} + \alpha S \right] \quad (2.12)$$

where $\alpha = 0.05$.

The rationale for equation (2.12) is that if the previous VOP was complex and used a high number of bits, then the number of bits to encode the upcoming VOP should be increased. However, since the remaining number of bits is lower, fewer bits can be allocated to this VOP. The weighted average controlled by α reflects this trade-off. In order to guarantee a minimum quality, a lower bound of $R_s/30$ bits is allocated to each VOP.

II – Buffer Control

In order to try to prevent video rate buffer verifier (VBV) (see Section 4.2 for a detailed description of this mechanism) violations, T is further adjusted according to the buffer occupancy; if the buffer occupancy is less than half the buffer size, then T is increased, otherwise it is decreased, i.e.,

$$T = T \frac{B + 2(B_s - B)}{2B + (B_s - B)} \quad (2.13)$$

where B is the current video rate buffer occupancy and B_s is the video rate buffer size.

For the cases where even after the adjustment provided by (2.13) the allocated number of bits can still lead to a potential violation of the VBV constraints, a more drastic adjustment is performed, i.e.,

$$T = \begin{cases} \max \left[\frac{R_s}{30}, \beta B_s - B \right] & \Leftarrow B + T > \beta B_s \\ R_p + (1-\beta) B_s - B & \Leftarrow B + T - R_p < (1-\beta) B_s \end{cases} \quad (2.14)$$

where $\beta = 0.9$. The first term of (2.14) attempts to avoid imminent encoder buffer overflows, while the second term attempts to avoid imminent encoder buffer underflows.

STEP 3 – QUANTIZATION PARAMETER COMPUTATION

In this step, the algorithm computes the quantization parameter, Q_c , for quantizing the residual texture data (luminance and chrominance DCT coefficients) of the whole current VOP, by solving equation (2.9).

Before using (2.9), T is further adjusted to ensure that the number of bits used to encode the residual texture data is “higher” than the header, motion vector, and shape (if applicable) bits, H_p , estimated from the previous encoded VOP, i.e.,

$$T = \max \left[\frac{R_p}{3} + H_p, T \right] \quad (2.15)$$

Using T given by (2.15) in (2.9) leads to

$$Q_c = \begin{cases} X_{11} \frac{1}{(T-H_p)} E_c & \Leftarrow (X_2 = 0) \vee ((X_1 E_c)^2 + 4X_2 E_c (T-H_p)) < 0 \\ \frac{2X_2 E_c}{\sqrt{(X_1 E_c)^2 + 4X_2 E_c (T-H_p)} - X_1 E_c} & \Leftarrow \text{otherwise} \end{cases} \quad (2.16)$$

where X_{11} is the model parameter for the fall back first order model.

The quantization parameter obtained through (2.16) must be clipped in order to remain between 1 and 31. Additionally, in order to avoid large quality fluctuations between consecutive encoding time instants, the current VOP quantization parameter is further limited to vary within a certain fraction of the previous VOP quantization parameter, i.e.,

$$\begin{aligned} Q_c &= \min \left[\lceil (1+\gamma) Q_l \rceil, Q_c, 31 \right] \\ Q_c &= \max \left[\lceil (1-\gamma) Q_l \rceil, Q_c, 1 \right] \end{aligned} \quad (2.17)$$

where $\gamma=0.25$, Q_l is the quantization parameter used to encode the previous VOP, and $\lceil x \rceil$ represents the smallest integer larger than x .

Using the quantization parameter, Q_c , obtained through (2.17), the current residual VOP texture data is quantized and the resulting symbols are entropy encoded.

STEP 4 – POST ENCODING

This step performs three main tasks:

I – Bit count and buffer occupancy updating

$$\begin{aligned} R_r &= R_r - R_c \\ B &= B + (R_c - R_p) \end{aligned} \quad (2.18)$$

where R_c is the number of bits used to encode the current VOP.

II – Rate-quantization model parameters updating

Rewriting (2.9) as

$$\frac{R(Q) \cdot Q}{E_c} = X_1 + X_2 \frac{1}{Q} \quad (2.19)$$

and let $y_i = \frac{R_{c_i} - H_{c_i}}{E_{c_i}} Q_i$ and $x_i = \frac{1}{Q_i}$, where R_{c_i} , H_{c_i} , E_{c_i} , and Q_i are respectively: the total number of bits used to encode VOP i ; the header, motion vector, and shape (if applicable) bits; the encoding complexity; and the quantization parameter. Notice that, since the header, motion vector, and shape bits do not directly depend on the quantization parameter used, they are not taken into account to estimate the rate-quantization model parameters.

A first estimate of the model parameters is obtained through the minimization of

$$\chi^2 = \sum_{k=i-w+1}^i (y_k - (X_1 + X_2 x_k))^2 \quad (2.20)$$

where w is a sliding window size that controls the number of data points used to estimate the model parameters, i.e., $w = \min[i, w_{\max}]$, where $w_{\max} = 20$. If the encoding VOP complexity changes significantly, a smaller window with the more recent data points is used, i.e.,

$$w = \begin{cases} \left\lceil \frac{E_{C_i}}{E_{C_{i-1}}} w \right\rceil & \Leftarrow E_{C_i} < E_{C_{i-1}} \\ \left\lceil \frac{E_{C_{i-1}}}{E_{C_i}} w \right\rceil & \Leftarrow E_{C_i} \geq E_{C_{i-1}} \end{cases} \quad (2.21)$$

In order to avoid model biasing due to outlier data points, a second estimate of the model parameters is obtained, considering, now, only the data points for which the prediction error is less than a rejection threshold, i.e., the data points that verify the following condition

$$\left| (R_{c_i} - H_{c_i}) - \left(X_1 \frac{1}{Q_i} + X_2 \frac{1}{Q_i^2} \right) E_{C_i} \right| < \sqrt{\frac{\sigma^2}{w}} \quad (2.22)$$

where

$$\sigma^2 = \sum_{k=i-w+1}^i \left[(R_{c_k} - H_{c_k}) - \left(X_1 \frac{1}{Q_k} + X_2 \frac{1}{Q_k^2} \right) E_{C_k} \right]^2 \quad (2.23)$$

Notice that a new sliding window size is also computed for the second estimation step that directly derives from the number of data points in the previous window that verify (2.22).

III – Post-encoding VOP skipping control

Finally, if the buffer occupancy is above a certain threshold, the next VOP(s) will be skipped until the buffer occupancy reaches again a nominal point of operation, in order to prevent encoder buffer overflow for the upcoming VOPs, i.e.,

while

$$B > \delta B_s, \quad (2.24)$$

where $\delta = 0.8$, the next VOP to be encoded is skipped and the buffer occupancy is updated as

$$B = B - R_p \quad (2.25)$$

2.5.2 Multiple Video Object Rate Control

This rate control algorithm is an extension of the frame rate control algorithm for the encoding of multiple VOs. The same type of rate-quantization model is used relatively to the frame rate control algorithm. However, in this case, a different model is used for each VO in the scene. The MVO rate control algorithm is divided into four main steps described below.

STEP 1 – INITIALIZATION

This step is similar to the frame control initialization step. However, a combined VBV buffer approach is adopted where a single VBV buffer with a single VBV drain rate is used for all VOs in the scene. This algorithm, as most of the currently available MPEG-4 rate control solutions [14, 77, 78] (rate control is non-normative in MPEG standards) assume synchronous VOs, this means that all VOs are coded at the same VOP rate. However, this approach may

reveal itself inefficient since the various VOs in the scene may exhibit very different needs in terms of temporal resolution, notably during object fast movements and stationary periods. This problem is addressed in the rate control algorithm proposed in [26] where the various VOs composing a scene can be encoded at different VOP rates.

Notice that, in this case, most of the algorithm parameters are represented as vector parameters, being the vector size equal to the number of VOs in the scene.

For the MVO rate control algorithm, the number of remaining bits to encode the scene, R_r , and the average number of bits to remove from the common VBV buffer at the end of each encoding time instant, R_p , are computed in the same way as in (2.10) and (2.11).

In the case of non-synchronous VOs, as the case considered in [26], these values are not so straightforward to compute and a more careful analysis of the whole scene is required for each encoding time instant.

Regarding the initial quantization parameters for the first I- and P-VOPs of each VO in the scene, this algorithm considers two possibilities: a first one, where these values are set a priori by the user, and a second one where a different quantization parameter is assigned to each MB of these VOPs based on MB classifications and table lookups.

STEP 2 – TARGET BIT ALLOCATION

This step performs six main tasks:

I – Rate control mode Decision

Before estimating a target number of bits to encode the set of VOPs for the current time instant, the algorithm selects a mode of operation between two possible modes, i.e.,

$$\text{mode} = \begin{cases} \text{LowBitRate} \Leftarrow N_{\text{skip}} > \text{Skip}_{TH} \\ \text{HighBitRate} \Leftarrow N_{\text{skip}} \leq \text{Skip}_{TH} \end{cases} \quad (2.26)$$

where N_{skip} is the number of skipped time instants computed on the last actual encoding time instant, and Skip_{TH} is the mode decision threshold (typically $\text{Skip}_{TH} = 2$).

The mode of operation is essentially used to impose constraints on some rate control algorithm parameters, such as the target bit allocation distribution weights and the shape rate control parameters.

II – Initial target estimation

The initial target number of bits for the current encoding time instant is computed as in (2.12). However, in the MVO case, $\alpha = 0.2$, increasing the weight of the previous encoding time instant results in the target number of bits estimation for the current time instant.

III – Joint buffer control

As in the frame rate control algorithm, the initial target number of bits is further adjusted in order to prevent the encoder buffer violations according to (2.13) and (2.14). However, in this case, $\beta = 0.75$, reflecting this way the need for a higher safety margin.

IV – Pre-encoding VOP skipping control

When the bit rate resources are scarce, notably in low bit rate encoding, the target number of bits for the current time instant that emerges from the joint buffer control may not be sufficient to even encode the auxiliary data (i.e., header, motion vector, and shape data), thus, no bits would be left for encoding the texture data. In these cases, the MVO rate control algorithm issues an alert for the following algorithmic steps to signal this scarceness in the form of pre-encoding skip information. The rate control algorithm estimates the number of encoding time instants to skip, N_{pre_skip} , such that

$$T - H_p + N_{pre_skip} \cdot R_p \geq \varepsilon \quad (2.27)$$

where T is the target number of bits for the current encoding time instant resulting from the joint buffer control, H_p is the total number of bits used in the previous encoding time instant for encoding the auxiliary data (i.e., header, motion vector, and shape data), and ε is a control threshold greater than or equal to zero.

V – Target distribution

After the target adjustment performed by the joint buffer control, if enough bits are left to encode the set of current VOPs, it is necessary to distribute this target number of bits among the various VOPs to encode. Since the different VOPs to encode may have different sizes, and different complexities, it is important to take this into account. In this case, this is done by assigning a weight to each VOP

$$\omega_i = \omega_M \cdot M_i + \omega_S \cdot S_i + \omega_V \cdot V_i \quad (2.28)$$

where M_i , S_i , and V_i are respectively the normalized motion, size and variance (MAD^2) of VOP i , and ω_M , ω_S , and ω_V are weights that control the importance of the different types of data in the target number of bits distribution. Other types of data can also be considered for distributing the bit rate, such as the VO priority and the VOP coding mode [14]. Typically, for the LowBitRate mode, $\omega_M = 0.6$, $\omega_S = 0.4$, and $\omega_V = 0.0$; for the HighBitRate mode, $\omega_M = 0.25$, $\omega_S = 0.25$, and $\omega_V = 0.50$.

The target number of bits is finally distributed among each VOP to encode, according to the weight computed through (2.28) as

$$T_i = \omega_i \cdot T \quad (2.29)$$

VI – Shape rate control

The number of bits used to encode the shape data can be controlled through the α_{th} parameter. A value of $\alpha_{th} = 0$ implies that the shape must be encoded in a lossless manner, while higher α_{th} values allow the shape to be encoded in lossy manner and consequently with fewer bits. Allowing the shape to be encoded in a lossy mode has the purpose of saving bits to encode the other types of data, such as the texture data. This has particular relevance if the LowBitRate mode is selected, thus α_{th} is updated according to the current mode of operation as follows

$$\alpha_{th} = \begin{cases} \alpha_{th} + \Delta_+ & \leftarrow \text{mode=LowBitRate} \\ \alpha_{th} - \Delta_- & \leftarrow \text{mode=HighBitRate} \end{cases} \quad (2.30)$$

where, typically, for rate control purposes $\Delta_+ = \Delta_- = 4$ and $0 \leq \alpha_{th} \leq 64$. In terms of the standard, $0 \leq \alpha_{th} \leq 255$. However, experience has shown that allowing high levels of degradation in the shape data can be subjectively very annoying and, therefore, the maximum value of α_{th} should be selected conservatively.

STEP 3 – QUANTIZATION PARAMETER COMPUTATION

The quantization parameter computation for each VOP to encode is performed in the same way as for the frame rate control algorithm, i.e., by following equations (2.16) and (2.17). Notice that a different rate-quantization model is used for each VO in the scene.

STEP 4 – POST ENCODING

This step performs three main tasks:

I – Bit count and the buffer occupancy updating

As for the previous algorithm, after encoding all VOPs for the current encoding time instant, the remaining number of bits and the buffer occupancy are updated similarly to (2.18). However, in this case, R_c is replaced by the number of bits used to encode all the VOPs for the current encoding time instant, i.e., the sum of all R_c^j , where j is the VO index.

II – Rate-quantization model parameters updating

After encoding, the rate-quantization model parameters for each VO are independently updated, similarly to the frame rate control algorithm, i.e., given the number of texture bits, $R_c^j - H_c^j$, the quantization parameter, Q^j , and the encoding complexity, E_C^j , for the current VOP of VO j , and all previous w^j VOPs, the new X_1^j , and X_2^j model parameters are estimated through linear least-squares estimation in two passes, as in the former case.

III – Post-encoding VOP skipping control

The post-encoding VOP skipping control determines the number of encoding time instants to skip due to imminent encoder buffer overflow, i.e., if after encoding and buffer updating, the encoder buffer occupancy is above a certain threshold, δB_s (in this case $\delta = 0.8$), the rate control algorithm estimates the number of encoding time instants to skip, N_{post_skip} , such that a virtual encoder buffer occupancy B_v verifies the condition

$$B_v \leq \delta B_s \quad (2.31)$$

where

$$B_v = B + S - (N_{post_skip} + 1) R_p \quad (2.32)$$

where S is the total number of bits used in previous encoding time instant. After finding N_{post_skip} , the total number of skipped time instants is updated as

$$N_{skip} = N_{pre_skip} + N_{post_skip} \quad (2.33)$$

This algorithm assumes that all VOs in the scene are synchronous and skipping one encoding time instant means skipping the encoding of all VOs for that time instant.

2.5.3 Macroblock Rate Control

The rate control methods described in sections 2.5.1 and 2.5.2 use a fixed quantization parameter for all the MBs in a VOP, computed based on a given target number of bits and the corresponding VO rate-quantization model. However, in certain situations, notably, in low delay video encoding, using a fixed quantization parameter for all MBs in a VOP may lead frequently to imminent buffer violations due to large deviations between the actual and the estimated number of encoded bits.

This algorithm tries to circumvent the above problem by providing a way to achieve a more accurate rate control, i.e., by allowing the quantization step to change from MB to MB inside a VOP.

The macroblock rate control assumes that the encoder rate-quantization function can be modeled as

$$B_i = \begin{cases} A_1 \frac{1}{Q_i^2} MAD_i^2 & \Leftarrow R_{bpp} > \varepsilon_T \\ A_2 \frac{1}{Q_i^2} MAD_i + A_3 \frac{1}{Q_i} MAD_i & \Leftarrow R_{bpp} \leq \varepsilon_T \end{cases} \quad (2.34)$$

where B_i is the number of bits to encode the texture data of MB i , A_1 , A_2 , and A_3 are the model parameters, Q_i is the MB quantization parameter, MAD_i is the mean absolute difference for the MB, R_{bpp} is the target number of bits per pixel for encoding the MB texture data, and ε_T is a control parameter (by default $\varepsilon_T = 0.085$).

In this algorithm, all the MBs in a VOP are encoded sequentially in a raster scan order according to the following steps described below.

STEP 1 – INITIALIZATION

At the beginning of each VOP the following parameters need to be initialized:

I – Rate-quantization model parameters

A_1 , A_2 , and A_3 are initialized with the latest encoded VOP model parameters A_1^{prev} , A_2^{prev} , and A_3^{prev} (for the first VOP of a given VO, $A_1^{prev} = 100$, $A_2^{prev} = 400$, and $A_3^{prev} = 0$).

II – VOP target number of bits

Set the target number of bits to encode the current VOP texture data, T , as in the case of the frame rate control algorithm.

III – MB rate control operation mode

$$K = \begin{cases} 1 & \Leftarrow R_{bpp} > \varepsilon_T \\ 2 & \Leftarrow R_{bpp} \leq \varepsilon_T \end{cases} \quad (2.35)$$

where R_{bpp} is the target number of bits per pixel for the current VOP, i.e., T divided by the number of pixels in the current VOP; and $\varepsilon_T = 0.085$.

STEP 2 – MB TARGET BIT ALLOCATION

In this step, the algorithm computes the number of bits to encode the current MB, based on the remaining number of bits to encode the VOP texture data and the MB energy expressed by its MAD , as follows

$$T_i = \frac{\omega_i \cdot MAD_i}{S_i} \left(T - \sum_{k=1}^{i-1} B_k \right) \quad (2.36)$$

where ω_i is the MB perceptual importance (by default $\omega_1 = \omega_2 = \dots = \omega_N = 1$) and

$$S_i = \sum_{k=i}^N \omega_k \cdot MAD_k$$

where N is the total number of MBs in the current VOP.

STEP 3 – MB QUANTIZATION PARAMETER COMPUTATION

In this step, the algorithm computes the quantization parameter to encode the current MB, based on the target number of bits assigned to a given MB, T_i , using (2.34), i.e.,

$$Q_i = \begin{cases} MAD_i \cdot \sqrt{\frac{A_1}{T_i}} & \Leftarrow K = 1 \\ \frac{2A_2 \cdot MAD_i}{\sqrt{(A_3 \cdot MAD_i)^2 + 4A_2 \cdot T_i \cdot MAD_i - A_3 \cdot MAD_i}} & \Leftarrow K = 2 \end{cases} \quad (2.37)$$

If $((A_3 \cdot MAD_i)^2 + 4A_2 \cdot T_i \cdot MAD_i) < 0$, for $K = 2$, equation (2.37) has no real solution. In this case, a fall back first order model is used instead, i.e.,

$$Q_i = A_{31} \frac{1}{T_i} MAD_i \quad (2.38)$$

where A_{31} is similar to the X_{11} parameter in (2.16).

Notice that in the MPEG-4 visual standard [29], the quantization parameter is bounded between 1 and 31. Furthermore, inside a VOP, the quantization parameter of consecutive MBs cannot differ by more of ± 2 , meaning that Q_i must be clipped to comply with these constraints.

Notice that combining (2.36) and (2.37) leads to the following expression:

$$Q_i^2 \approx \begin{cases} MAD_i \cdot C_1 & \Leftarrow K = 1 \\ C_2 & \Leftarrow K = 2 \end{cases} \quad (2.39)$$

where C_1 and C_2 depend, on $\{A_1, MAD_1, \dots, MAD_N, T\}$ and $\{A_2, A_3, MAD_1, \dots, MAD_N, T\}$, respectively.

The model parameters A_1 , A_2 , and A_3 are estimated while encoding. However, if the model fits well the experimental data, these values are approximately constant within a VOP. Therefore, C_1 and C_2 are also approximately constant. Consequently, for low bit rates (i.e., $K = 2$), the quantization parameter tends to be approximately constant for all MBs in a VOP,

trying to minimize the overhead bits necessary to signal quantization parameter changes, while for higher bit rates (i.e., $K=1$) the quantization parameter tends to increase with the MB energy expressed by its MAD .

STEP 4 – MB RATE-QUANTIZATION MODEL PARAMETERS UPDATING

After encoding MB i , the rate-quantization model parameters are updated based on the encoding results for the current and previous MBs.

If $B_i = 0$ (i.e., no texture data bits were produced), only the number of MBs without texture data bits, n_s , is updated, i.e., $n_s = n_s + 1$.

Otherwise, for $K = 1$

$$\hat{A}_1 = \frac{B_i}{MAD_i^2} Q_i^2, A'_1 = \hat{A}_1 \frac{1}{i - n_s} + A'_1 \frac{i - n_s - 1}{i}, A_1 = A'_1 \frac{i}{N} + A_1^{prev} \frac{N - i}{N} \quad (2.40)$$

and for $K = 2$

A_1 , A_2 , A_3 , and A_{31} are estimated through linear least squares estimation using a sliding window containing the last w MBs for which $B_i > 0$, i.e.,

$$w = \max[n_c, w_{\max}] \quad (2.41)$$

where n_c is the number of encoded MBs in the current VOP for which $B_i > 0$, and $w_{\max} = 20$.

After this step, the algorithm returns to STEP 2 until all MBs are done.

2.6 MPEG-4 Profiling

The MPEG-4 standard has been designed to be generic in the sense that it is not targeted for a particular application but includes many coding tools and algorithms that can be used in a variety of applications under different operating conditions, notably, different bit rates, types of channels and storage media [58]. This toolbox approach provides the mechanisms to cover a wide range of audio-visual applications from mobile multimedia communications to studio and interactive TV [58, 54].

Since it is not reasonable that all terminals support the complete toolbox, subsets of the MPEG-4 standards have been defined, through the concept of profiling, to address classes of applications with similar functional and operational requirements. This approach allows manufacturers to implement only the subset of the standard they need to achieve particular functionalities, under two important conditions: 1) maintain interoperability with other MPEG-4 devices built within the same conditions, and 2) restrict the computational resources required by the given terminals. Additionally, the profiling mechanism allows adding new functionalities to the standard or significantly improving existing ones when there is enough evidence and requirements for such achievements.

2.6.1 Profiling Concepts

In MPEG-4, the profiling mechanism relies on the following concepts: object types, profiles, levels, and conformance points. These concepts are the key elements in providing interoperability between different MPEG-4 enabled devices/applications since the defined profiles at defined levels specify conformance points for decoders and bitstreams to adhere to.

OBJECT TYPES

Although a large set of audio and visual coding tools has been standardized by MPEG, not all combinations of these tools are allowed within an audio or visual object. Some of these combinations are simply not supported by the syntax while others, although supported by the syntax, are not allowed due to a lack of requirements and in order to minimize complexity. In this last case, if requirements arise for these combinations, they may be later supported.

As mentioned previously, in the MPEG-4 object-based coding architecture, the data model relies on the concept of objects that are independently coded. Therefore, each coded object needs to be characterized syntactically and semantically.

An object type defines the combination of tools that can be used to encode single audio or visual objects that can represent meaningful entities in the scene, thus it defines the syntax and semantics of the ESs²⁰ for a single audio or visual object. In MPEG-4, two classes of object types have been defined: audio object types and visual object types. Audio object types define possible combinations of audio tools to produce valid audio ESs, while visual object types represent possible combinations of visual tools to produce valid visual ESs.

PROFILES

Profiles define subsets of the complete MPEG-4 toolbox targeting large classes of applications that can be used in a certain MPEG-4 terminal or application. However, profiles only define the syntax and semantics of ESs, they do not define complexity bounds, i.e., a given profile only defines which object types can be used in a given audio-visual scene.

Although profiles have been defined for the several parts of MPEG-4, i.e., Systems, Audio, and Visual, it is not specified which combinations of these profiles are acceptable giving the market the possibility to decide. It is even possible to combine one or more of the MPEG-4 parts, e.g., Visual or Audio, with non-standard solutions, e.g., for Systems, although this will not exploit the synergies between the various MPEG-4 standards [51]. MPEG-4 defines the following types of profiles:

- **Audio and Visual Profiles** – Profiles that define the type of audio and visual objects the MPEG-4 terminal needs to be able to decode and, hence, give a list of admissible object types (ES types). Profiling of audio and visual tools intends to restrict the decoder complexity in terms of processing power and memory usage required to process the audio and visual ESs building the scene.
- **Graphics Profiles** – Profiles that define the graphical elements that can be combined in the scene. They are expressed in terms of the BIFS nodes, which provide means to represent graphical visual objects in a scene. Profiling of graphics elements of the BIFS tool serves to restrict the computational complexities of the composition and rendering processes and the memory requirements for their storage.
- **Scene Graph Profiles**²¹ – Profiles that define the scene graph nodes that are allowed to build an audio-visual scene. They specify the scene graph elements of the BIFS tool that are allowed. These elements provide means to describe the spatio-temporal locations, the hierarchical dependencies as well as the behavior of audio-visual objects in a scene. Profiling of scene graph elements of the BIFS tool aims at restricting the

²⁰ In the case of scalable objects more than one elementary stream can be associated with a single object.

²¹ Scene graph profiles are usually also referred as scene description profiles.

memory requirements and computational complexities of the scene graph traversal and processing of specified behaviors during the composition and rendering processes.

- **Object Descriptor Profiles** – Profiles that define the configurations of the object descriptor and the sync layer tools that are allowed to build an audio-visual scene. The object descriptor tool provides a structure for all descriptive information, while the sync layer tool provides the syntax to convey, timing, synchronization, fragmentation, and random access information for ESs. These profiles are used, mainly, to reduce the amount of asynchronous operations as well as the amount of permanent storage.
- **MPEG-J Profiles** – Profiles that define subsets of the MPEG-J Application Programming Interfaces. These profiles are used to restrict the power of the virtual machine a device needs to support.

LEVELS

Profile definitions alone do not provide a full characterization of the receiving terminal capabilities and the resources needed for a presentation. Profiles give no bounds on parameters such as bit rate or decoding memory. For this reason, levels are defined within each profile to constrain the values of some parameters in order to specify upper complexity bounds.

A level is a specification of the constraints and performance criteria on a given profile, and therefore on the corresponding tools. Profiles can only exist at a certain level; there are no profiles without a level, although a profile can be defined at a single level²².

CONFORMANCE POINTS

A conformance point is a specification of a particular profile at a certain level at which decoder and bitstream conformance may be tested. Conformance points are normatively specified within the MPEG-4 standard. MPEG-4 also provides guidelines for constructing tests to verify bitstream and decoder conformance. Typically, such tests define decoder input (i.e. bitstreams) and decoder output (e.g. waveforms for decoded audio and pixel values for video objects) [31].

2.6.2 Version Management

Profiles and levels are the final outcome of the standardization process, providing high performance solutions for various classes of applications [79]. The challenge of profiling is to maximize interoperability while minimizing implementation costs. However, a large number of profiles may generate some confusion in the industry and makes interoperability more difficult. Therefore, the MPEG group has been trying to restrict the number of profiles by allowing the inclusion of new tools only if they bring new functionalities or significantly improve the performance of existing ones. In this context, a new profile is only created when there is enough evidence that an existing one is not usable (e.g., not sufficiently powerful). As for levels, new levels for a given profile are only created when there is enough evidence (notably, through expressions of interest by companies) that existing levels are not sufficient.

In this context, during the MPEG-4 development process, it was decided to issue successive versions (later called extensions) of the several MPEG-4 Parts whenever new tools needed to

²² At the time of writing this Thesis no levels have been defined for object descriptor and MPEG-J profiles.

be added to that Part of the standard. Therefore, versions are used to specify new tools, either offering new or significantly improving existing functionalities [51].

MPEG-4 version 1 was approved by MPEG in December 1998; version 2 was frozen in December 1999, and version 3 was issued in 2004. It is important to notice that new versions of a Part of the standard do not substitute or redefine tools specified in previous versions but just add more tools. At each stage of specification, a certain Part of the MPEG-4 standard is the set of all the tools specified in all versions for that part. In that sense, it is common to say that versions are backward compatible meaning that Version N may only add new tools and profiles to Version N-1 and not remove or redefine any tool or profile. This implies that existing terminals will always remain compliant, since profiles will not be changed retrospectively. Figure 2.22 illustrates the relationship between the different versions.

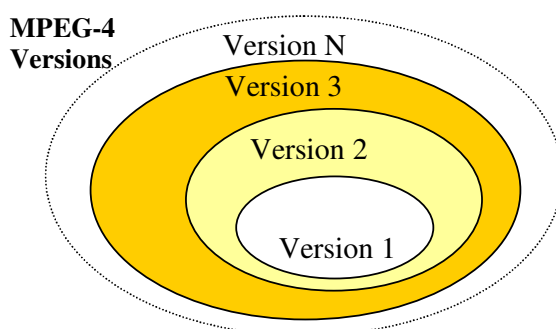


Figure 2.22 – Relation between MPEG-4 versions [54]

2.6.3 Visual Object Types

In MPEG-4 Visual [29] a visual scene may contain several objects of different kinds, e.g., rectangular or arbitrarily shaped VOs, still textures, or synthetic objects. However profiles aim at characterizing the scene as whole and not individual objects. Therefore, it is necessary to specify an intermediate level between the coding tools and the scene profiles – the object type – that defines the subset of MPEG-4 Visual tools that can be used to code a given object in the scene. Table 2.4 presents the correspondence between the MPEG-4 Visual tools and the object types specified in version 3 of the MPEG-4 Visual standard [29].

Below, the visual object types presented in Table 2.4 are briefly described following the organization adopted in [51], where the different visual object types are grouped in four classes: rectangular video, arbitrarily shaped video, still visual, and synthetic visual object types.

RECTANGULAR VIDEO OBJECT TYPES

The MPEG-4 standard provides five different object types for representing natural video information of rectangular shape.

- **Simple Object Type** – Rectangular video object of arbitrary height-to-width ratio supporting error resilience tools. It uses relatively simple and inexpensive coding tools, based on I- and P-VOPs, and targets low bit rate coding.
- **Advanced Simple Object Type** – Rectangular video object supporting enhanced compression efficiency tools, such as quarter-pel motion estimation, global motion estimation, and B-VOPs.

- **Advanced Real-Time Simple (ARTS) Object Type** – Superset of the Simple object type providing a back channel to monitor throughput, to adapt VOP spatial coding resolution, and to dynamically resend lost information. It also adds dynamic resolution conversion to the Simple object type tool set. It targets real-time coding applications.
- **Simple Scalable Object Type** – Scalable extension of the Simple object type providing temporal and spatial scalability. The base layer is of type Simple and the enhancement layer is still rectangular.
- **Fine Granularity Scalable Object Type** – Superset of the Advanced Simple object type providing temporal and fine-granular SNR scalability.

ARBITRARILY SHAPED VIDEO OBJECT TYPES

The MPEG-4 standard provides seven different object types for representing natural arbitrarily shaped video information.

- **Core Object Type** – Superset of Simple object type with arbitrary binary shape²³ and B-VOPs. It supports temporal scalability (extra P-VOPs only).
- **Core Scalable Object Type** – Scalable extension of the Core object type providing rectangular temporal and spatial scalability as well as object-based spatial scalability.
- **Main Object Type** – Superset of the Core object type providing additionally gray-level shape coding, sprites, and interlaced coding.
- **Advanced Coding Efficiency Object Type** – Similar to the Main object type, excluding sprites but including some extra coding efficiency tools, namely, quarter-pel motion compensation, GMC, and SA-DCT.
- **N-Bit Object Type** – Superset of the Core object type providing flexible pixel depth from 4 to 12 bits for the luminance as well as the chrominance components. It targets less-usual image types, such as remote sensing and thermal images.
- **Simple Studio Object Type** – Arbitrarily shaped video object with multiple alpha planes for very high quality and bit rates. Only supports I-VOP coding with a syntax closer to MPEG-2 Video [10] to allow easy transcoding.
- **Core Studio Object Type** – Superset of the Simple Studio object that provides also P-VOP coding, increasing its complexity but making it more coding efficient.

STILL VISUAL OBJECT TYPES

The MPEG-4 standard provides two different object types for representing natural still visual information.

- **Scalable Texture Object Type** – Arbitrarily shaped still image object that uses wavelet coding for scalability providing incremental download and build-up.
- **Advanced Scalable Texture Object** – Superset of the Still Scalable Texture object type providing error resilience, better scalable shape coding, and partial decoding of the bitstream.

²³ Includes constant transparency but excludes the variable transparency offered by gray-level shape coding.

SYNTHETIC VISUAL OBJECT TYPES

The MPEG-4 standard provides four different object types for representing synthetic visual information. These object types use synthetic coding tools, although some of them can be combined with natural video or still image coding tools.

- **Simple Face Animation Object Type** – Animated facial object. This object type provides the necessary tools for animating 3D face models. It does not define the face model, just its animation. The animation can be applied to any local model selected by the receiver.
- **Simple Face and Body Animation (FBA) Object Type** – Superset of the Simple Face Animation object type providing also tools for body animation.
- **Basic Animated 2D Texture Object Type** – Animated 2D mesh object mapped with arbitrarily shaped still images (using the Scalable Texture object type). MPEG-4 supports two types of meshes: uniform and Delaunay; in this object type, only uniform meshes can be used.
- **Animated 2D Mesh Object Type** – Animated 2D mesh object mapped with natural arbitrarily shaped video. Both uniform and Delaunay meshes are supported. The natural video coding uses the same tools as the core object type. The video can be mapped onto the mesh and deformed by moving the mesh vertices providing interesting animation possibilities. Only binary shaped video is supported.

Table 2.4 – Visual tools versus visual object types [51]

Object Types → ↓ Visual Tools	Simple	Advanced Simple	Advanced Real-Time Simple	Simple Scalable	Fine Granularity Scalable	Core	Core Scalable	Main	Advanced Coding Efficiency	N-Bit	Simple Studio	Core Studio	Scalable Texture	Advanced Scalable Texture	Simple Face Animation	Simple FBA	Basic Animated Texture	Animated 2D Mesh
I-VOP	•	•	•	•	•	•	•	•	•	•	•	•						•
P-VOP	•	•	•	•	•	•	•	•	•	•	•	•						•
B-VOP		•		•	•	•	•	•	•	•								•
P-VOP with OBMC (Texture)																		
Basic Tools	•	•	•	•	•	•	•	•	•	•								•
Basic Tools (Studio Object Types)											•	•						
Error Resilience	•	•	•	•	•	•	•	•	•	•								•
Short Header	•	•	•		•	•	•	•	•	•								•
Method 1/Method 2 Quantization		•			•	•	•	•	•	•								•
P-VOP Based Temporal Scalability						•	•	•	•	•								•
Rectangular																		
Arbitrary Shape																		

Object Types → ↓ Visual Tools	Simple	Advanced Simple	Advanced Real-Time Simple	Simple Scalable	Fine Granularity Scalable	Core	Core Scalable	Main	Advanced Coding Efficiency	N-Bit	Simple Studio	Core Studio	Scalable Texture	Advanced Scalable Texture	Simple Face Animation	Simple FBA	Basic Animated Texture	Animated 2D Mesh
Binary Shape						•		•	•	•				•			•	•
Grey Shape								•	•									
Interlace		•			•			•	•		•	•						
Sprite								•										
Temporal Scalability (Rectangular)				•			•											
Spatial Scalability (Rectangular)				•			•											
Object Based Spatial Scalability							•											
Fine Granularity Scalability (FGS)					•													
FGS Temporal Scalability					•													
Global Motion Compensation		•							•									
Quarter-pel Motion Compensation		•							•									
Dynamic Resolution Conversion			•															
NewPred			•															
SA-DCT									•									
N-Bit										•								
Scalable Still Texture													•	•			•	•
Error Resilience for Visual Texture Coding														•				
Wavelet Tiling														•				
Scalable Shape Coding for Still Texture														•				
Facial Animation Parameters															•	•		
Body Animation Parameters																•		
2D Dynamic Mesh with Uniform Topology																	•	•
2D Dynamic Mesh with Delaunay Topology																		•

2.6.4 Visual Profiles

Visual profiles in MPEG-4 specify which visual object types can be present in the scene. Consequently, they determine which coding tools can be used to code the objects. Therefore, visual profiles are defined as lists of admissible object types. Table 2.5 presents the object types supported and the number of levels of each profile specified in version 3 of the MPEG-4 Visual standard [29].

Below, the visual profiles presented in Table 2.5 are briefly described, indicating some possible application areas they address. Level definitions are presented in Section 2.6.5 for the relevant profiles in the context of this Thesis, notably, some video profiles. Notice, that profiles are not defined for specific applications but to be rather generic in the sense that they may address a class of applications or services. Similarly to visual object types, visual profiles can be grouped in four classes: rectangular video, arbitrarily shaped video, still visual, and synthetic and hybrid natural/synthetic visual profiles.

RECTANGULAR VIDEO PROFILES

The MPEG-4 standard provides five different profiles for applications requiring only natural video information of rectangular shape.

- **Simple Profile** – Supports only the Simple object type and was created with low-complexity applications in mind, e.g., mobile audiovisual services, video transmission over the Internet, and small camera devices, for bit rates from 64 kbit/s to 384 kbit/s

(depending of the level used) and typical visual sessions of QCIF and CIF. Level 0 was defined at the request of the 3GPP consortium and allows only one single object in the scene, while levels 1 – 3 support up to four objects.

- **Advanced Simple Profile** – Supports the Simple and Advanced Simple object types. It is useful in real-time streaming applications and other low bandwidth applications, although it can support up to TV-size pictures and TV quality.
- **Advanced Real-Time Simple (ARTS) Profile** – Supports the Simple and Advanced Real-Time Simple object types. It is suitable for real-time coding applications, such as videophones, teleconferencing, remote surveillance, taking advantage of the back channel and the adaptive encoding to create higher resilience to errors and better performance under changing bandwidth conditions.
- **Simple Scalable Profile** – Supports the Simple and Simple Scalable object types in the same operational environments foreseen for the Simple Profile, but with scalability. It is useful for applications that provide services at more than one level of quality due to bit rate or decoder resource limitations, such as Internet use and software decoding.
- **Fine Granularity Scalability (FGS) Profile** – Supports FGS scalability using either the Simple or the Advanced Simple object types as base layers. It is useful for streaming applications over networks without quality of service (QoS).

ARBITRARILY SHAPED VIDEO PROFILES

The MPEG-4 standard provides five different profiles for applications requiring natural video information of rectangular and arbitrary shape.

- **Core Profile** – Supports the Core and Simple object types. It is useful for high-quality interactive and mobile broadcast services. Combines good quality with limited complexity and supports arbitrarily shaped video objects.
- **Advanced Core Profile** – Supports the natural video coding provided by the Core object type with the possibilities of the Advanced Scalable Texture object type. It is suitable for various content-rich multimedia applications such as interactive multimedia streaming over Internet.
- **Core Scalable Profile** – Superset of the Simple, Simple Scalable, and Core profiles. It adds scalability to Core, according to the Core Scalable object type.
- **Main Profile** – Supports the Simple, Core, Main, and Scalable Texture object types. It was created for audiovisual broadcast services, addressing progressive as well as interlaced content. It combines the highest quality with the versatility of arbitrarily shaped objects using gray-level shape coding. It is useful for interactive and entertainment-quality broadcast and DVD applications.
- **Advanced Coding Efficiency (ACE) Profile** – Similar to the Main Profile, supports the Advanced Coding Efficiency object type (besides Simple and Core) that includes the GMC and quarter-pel motion compensation coding efficiency tools, but it does not support sprites that are found in the Main object type (not supported in this profile). It is suitable for applications such as mobile broadcast reception, the acquisition of image sequences (camcorders) and other applications where high coding efficiency is requested and computational complexity is not the prime constraint.

- **N-Bit Profile** – Supports the Simple, Core, and N-Bit object types. It is useful for applications that use thermal images, such as surveillance applications, and for medical applications that may benefit from the enhanced pixel depth, giving a larger dynamic range in color and luminance.
- **Simple Studio Profile** – Supports only the Simple Studio object type. It is meant for editing video in the studio and other professional applications requiring very-high quality (i.e., bit rates from 180 Mbit/s to 1800 Mbit/s).
- **Core Studio Profile** – Supports the Simple Studio and Core Studio object types. It is intended for editing uses in the studio for applications allowing higher complexity. Therefore, predictive coding (P-VOPs) can be used to reduce the bit rate (the maximum bit rate allowed is lowered to 900 Mbit/s due to the extra coding efficiency).

STILL VISUAL PROFILES

The MPEG-4 standard provides two different profiles for applications requiring natural still visual information.

- **Scalable Texture Profile** – Supports only the Scalable Texture object type. It is meant for applications including still visual material, possibly, in combination with audio. It was requested by companies interested in building rather simple mobile terminals that combine sound with synchronously displayed images and graphics.
- **Advanced Scalable Texture Profile** – Supports only the Advanced Scalable Texture object type. It is useful for fast content-based still image browsing on the Internet, multimedia-enabled PDA's, and Internet-ready high-resolution digital still cameras.

SYNTHETIC AND HYBRID NATURAL/SYNTHETIC VISUAL PROFILES

The MPEG-4 standard provides four different profiles for applications requiring synthetic as well as the combination of synthetic and natural visual information.

- **Simple Face Animation Profile** – Supports only the Simple Face Animation object type. It provides simple means to animate a face model, suitable for applications such as audio/video presentation for the hearing impaired. Depending on the level, either one or a maximum of four faces can appear in the scene, e.g., for a virtual meeting. It is meant for operating at very low bit rates, e.g., 16 kbit/s to 32 kbit/s, since only animation parameters are transmitted.
- **Simple Face and Body Animation (FBA) Profile** – Similar to the Simple Face Animation Profile, but now for face and body animation.
- **Basic Animated Texture Profile** – Supports the Scalable Texture, Basic Animated Texture, and Simple Face Animation object types. It is intended for creating rich content at very low bit rates.
- **Hybrid Profile** – Supports the combination of natural and synthetic objects in the same scene while keeping complexity reasonable. It is meant for content-rich multimedia applications. On the natural side, it compares to the Core Profile, whereas on the synthetic side, it adds animated meshes, scalable textures, and animated faces. This profile allows to place natural objects into a synthetic (virtual) world and vice versa, i.e., adding synthetic objects to a natural environment.

A partial hierarchy exists in the visual profiles, derived from a similar hierarchy for the

corresponding object types. More specifically, Main is a superset of Core, which is a superset of Simple; N-Bit is a superset of Core; Simple Scalable is a superset of Simple; Advanced Simple is a superset of Simple, and FGS is a superset of both Simple and Advanced Simple.

Table 2.5 – Visual profiles versus visual object types [51]

Visual Object Types → ↓ Visual Profiles	Simple	Advanced Simple	Advanced Real-Time Simple	Simple Scalable	Fine Granularity Scalable	Core	Core Scalable	Main	Advanced Coding Efficiency	N-Bit	Simple Studio	Core Studio	Scalable Texture	Advanced Scalable Texture	Simple Face Animation	Simple FBA	Basic Animated Texture	Animated 2D Mesh	Number of levels
Simple	•																		4
Advanced Simple	•	•																	6
ARTS	•		•																4
Simple Scaleable	•			•															2
FGS	•	•			•														6
Core	•					•													2
Advanced Core	•					•								•					2
Core Scalable	•			•		•	•												3
Main	•					•		•					•						3
ACE	•					•			•										4
N-Bit	•					•				•									1
Simple Studio											•								4
Core Studio											•	•							4
Scaleable Texture													•						3
Advanced Scalable Texture														•					3
Simple FA															•				2
Simple FBA																•			2
Basic Animated Texture													•		•		•		2
Hybrid	•					•							•		•		•	•	2

2.6.5 Video Profile@Level Definitions

As mentioned in Section 2.6.1, profiles alone do not characterize the complexity of the encoded data; for this, it is necessary to specify also the level. MPEG-4 has defined a set of profile@level combinations covering a wide range of coding tools subsets and coding parameters. Table 2.6 presents the list of the supported video profile@level combinations and the corresponding constraining parameters for the Version 1 and Version 2 profiles²⁴. The constraints imposed by these parameters can be divided into generic constraints applicable to all profile@level combinations and specific constraints only applicable to some tools for the relevant profiles. These constraints are defined and described in the following.

GENERIC PROFILE@LEVEL CONSTRAINTS

- **Typical Visual Session Size** – The typical target pixel visualization area. It is a merely informative value since no restrictions are directly specified for it. It is indirectly constrained by the picture memory and decoding complexity constraints (see below).
- **Number and Type of Objects** – The number and type of objects supported in each

²⁴ Since in the remainder of this Thesis only some video profiles are considered, the profile@level definitions for the remaining visual profiles are not presented here. A detailed description can be found in [29,51].

profile@level. It is defined by the *Maximum Number of Objects* specifying the maximum total number of objects supported in the profile@level (enhancement layers are not counted as separate objects), and by the *Maximum Number of Objects per Type* specifying the maximum number of objects of each type.

Picture Memory – The maximum amount of picture memory required at the decoder by each profile@level. It is defined by the *Maximum VMV Buffer Size* in terms of MB memory (see Section 4.2 for the VMV description).

- **Decoding Complexity** – The maximum decoding speed capabilities required by each profile@level. It is defined, in terms of MBs/s, by the VCV mechanism parameters *VCV Decoder Rate*, *VCV Boundary MB Decoder Rate*²⁵, and the *Maximum VCV Buffer Size* (see Section 4.2 for the VCV description).
- **Bitstream Memory** – The maximum total bitstream memory required by each profile@level. It is defined by the *Maximum Total VBV Buffer Size*, which specifies the maximum aggregated occupancy of all VOL VBV buffers of all VOs at any time instant (see Section 4.2 for the VBV description). Additional bitstream memory is needed when data partitioning is enabled. The amount of this memory supported by each profile@level is constrained by the *Maximum Video Packet Length*, that defines the maximum number of bits from the start of one slice to the start of the next slice. When data partitioning is disabled, the video packet length is not limited.
- **Bit Rate** – The maximum bit rate that a given terminal or application needs to process required by each profile@level. It is defined by the *Maximum Bit Rate* parameter considering all VOLs for all VOs in the scene.

SPECIFIC PROFILE@LEVEL CONSTRAINTS

- **Maximum Unique Quantization Tables** – For the profiles supporting both MPEG-4 quantization methods, when quantization method 1 is used (MPEG-2 like), the default quantization matrices may be replaced by new ones sent to the receiver through VOL configuration information. This parameter constrains the maximum number of unique quantization tables required by each profile@level.
- **Maximum Sprite Size** – As mentioned in Section 2.4.4, sprites can be progressively transmitted and stored at the decoder for later use through a spatial transformation process driven by a set of motion parameters. This parameter constrains the amount of sprite memory, defined in terms of MB memory, required by each profile@level.
- **Wavelet Restrictions** – Some profiles support the Scalable Still Texture object type subject to some restrictions. These restrictions are defined in terms of the Scalable Texture profile@level that the receiver needs to support. For more details see [29].
- **Number of Enhancement Layers per Object** – This parameter specifies the maximum number of enhancement layers per object supported by each profile@level allowing scalable video coding.

²⁵ This is only applicable for the profiles supporting arbitrarily shaped objects.

Table 2.6 – Levels for video profiles [51]

Visual profile	Level	Typical visual session size	Max. number of objects	Max. number of objects per type	Max. unique quant. tables	Max. VMV buffer size (MB units)	Max. VCV buffer size (MB units)	VCV decoder rate (MB/s)	VCV boundary MB decoder rate (MB/s)	Max. total VBV buffer size (units of 16384 bits)	Max. VOL VBV buffer size (units of 16384 bits)	Max. video packet length (bits)	Max. sprite size (MB units)	Wavelet restrictions	Max. bit rate (kbit/s)	Max. enhancement layers per object
Simple	L0	QCIF	1	1 x Simple	1	198	99	1485	N.A.	10	10	2048	N.A.	N.A.	64	N.A.
Simple	L1	QCIF	4	4 x Simple	1	198	99	1485	N.A.	10	10	2048	N.A.	N.A.	64	N.A.
Simple	L2	CIF	4	4 x Simple	1	792	396	5940	N.A.	40	40	4096	N.A.	N.A.	128	N.A.
Simple	L3	CIF	4	4 x Simple	1	792	396	11880	N.A.	40	40	8192	N.A.	N.A.	384	N.A.
Advanced Simple	L0	176x144	1	1 x Adv. Simple or Simple	1	297	99	2970	100	10	10	2048	N.A.	N.A.	128	N.A.
Advanced Simple	L1	176x144	4	4 x Adv. Simple or Simple	1	297	99	2970	100	10	10	2048	N.A.	N.A.	128	N.A.
Advanced Simple	L2	352x288	4	4 x Adv. Simple or Simple	1	1188	396	5940	100	40	40	4096	N.A.	N.A.	384	N.A.
Advanced Simple	L3	352x288	4	4 x Adv. Simple or Simple	1	1188	396	11880	100	40	40	4096	N.A.	N.A.	768	N.A.
Advanced Simple	L4	352x576	4	4 x Adv. Simple or Simple	1	2376	792	23760	50	80	80	8192	N.A.	N.A.	3000	N.A.
Advanced Simple	L5	720x576	4	4 x Adv. Simple or Simple	1	4860	1620	48600	25	112	112	16384	N.A.	N.A.	8000	N.A.
Advanced Real-Time Simple	L1	QCIF	4	4 x Simple or Adv. Real-Time Simple	1	198	99	1485	N.A.	10	10	8192	N.A.	N.A.	64	N.A.
Advanced Real-Time Simple	L2	CIF	4	4 x Simple or Adv. Real-Time Simple	1	792	396	5940	N.A.	40	40	16384	N.A.	N.A.	128	N.A.
Advanced Real-Time Simple	L3	CIF	4	4 x Simple or Adv. Real-Time Simple	1	792	396	11880	N.A.	40	40	16384	N.A.	N.A.	384	N.A.
Advanced Real-Time Simple	L4	CIF	16	16 x Simple or Adv. Real-Time Simple	1	792	396	11880	N.A.	80	80	16384	N.A.	N.A.	2000	N.A.
Simple Scalable	L1	CIF	4	4 x Simple or Simple Scal.	1	1782	495	7425	N.A.	40	40	2048	N.A.	N.A.	128	1*
Simple Scalable	L2	CIF	4	4 x Simple or Simple Scal.	1	3168	792	23760	N.A.	40	40	4096	N.A.	N.A.	256	1*
FGS	L0	176x144	1	1 x Adv. Simple or FGS or Simple	1	297	99	2970	100	10	10	2048	N.A.	N.A.	128	4
FGS	L1	176x144	4	4 x Adv. Simple or FGS or Simple	1	297	99	2970	100	10	10	2048	N.A.	N.A.	128	4
FGS	L2	352x288	4	4 x Adv. Simple or FGS or Simple	1	1188	396	5940	100	40	40	4096	N.A.	N.A.	384	4
FGS	L3	352x288	4	4 x Adv. Simple or FGS or Simple	1	1188	396	11880	100	40	40	4096	N.A.	N.A.	768	4
FGS	L4	352x576	4	4 x Adv. Simple or FGS or Simple	1	2376	792	23760	50	80	80	8192	N.A.	N.A.	3000	4
FGS	L5	720x576	4	4 x Adv. Simple or FGS or Simple	1	4860	1620	48600	25	112	112	16384	N.A.	N.A.	8000	4
Core	L1	QCIF	4	4 x Core or Simple	4	594	198	5940	2970	16	16	4096	N.A.	N.A.	384	1
Core	L2	CIF	16	16 x Core or Simple	4	2376	792	23760	11880	80	80	8192	N.A.	N.A.	2000	1
Advanced Core	L1	QCIF	4	4 x Core or Simple or Adv. Scal. Texture	4	594	198	5940	2970	16	8	4096	N.A.	**	384	1
Advanced Core	L2	CIF	16	16 x Core or Simple or Adv. Scal. Texture	4	2376	792	23760	11880	80	40	8192	N.A.	**	2000	1
Core Scalable	L1	CIF	4	4 x Core or Simple or Core Scal. or Simple Scal.	4	2376	792	14850	7425	64	64	4096	N.A.	N.A.	768	1
Core Scalable	L2	CIF	8	8 x Core or Simple or Core Scal. or Simple Scal.	4	2970	990	29700	14850	80	80	4096	N.A.	N.A.	1500	1
Core Scalable	L3	CCIR 601	16	16 x Core or Simple or Core Scal. or Simple Scal.	4	12906	4032	120960	60480	80	80	16384	N.A.	N.A.	4000	2
Main	L2	CIF	16	16 x Main or Core or Simple	4	3960	1188	23760	11880	80	80	8192	1584	Scalable Texture @L1	2000	1
Main	L3	CCIR 601	32	32 x Main or Core or Simple	4	11304	3240	97200	48600	320	320	16384	6480	Scalable Texture @L1	15000	1
Main	L4	1920x1088	32	32 x Main or Core or Simple	4	65344	16320	489600	244800	760	760	16384	65280	Scalable Texture @L2	38400	1
Advanced Coding Efficiency	L1	CIF	4	4 x Adv. Coding Efficiency or Core or Simple	4	1188	792	11880	5940	40	40	8192	N.A.	N.A.	384	1
Advanced Coding Efficiency	L2	CIF	16	16 x Adv. Coding Efficiency or Core or Simple	4	2376	1188	23760	11880	80	80	8192	N.A.	N.A.	2000	1
Advanced Coding Efficiency	L3	CCIR 601	32	32 x Adv. Coding Efficiency or Core or Simple	4	9720	3240	97200	48600	320	320	16384	N.A.	N.A.	15000	1
Advanced Coding Efficiency	L4	1920x1088	32	32 x Adv. Coding Efficiency or Core or Simple	4	48960	16320	489600	244800	760	760	16384	N.A.	N.A.	38400	1
N-Bit	L2	CIF	16	16 x Core or Simple or N-Bit	4	2376	792	23760	11880	80	80	8192	N.A.	N.A.	2000	1

* Spatial or temporal enhancement layer. ** Defined in the Scalable Texture profile @level definitions (see [29, 51]).

2.6.6 Performance Evaluation of Video Profiles

In the context of this Thesis the problem of comparing the performance of MPEG-4 video coding, in terms of coding efficiency, with other existing video coding standards has not been specifically addressed since that was not the focus of the Thesis. Nevertheless, several experts have reported this type of comparisons between some MPEG-4 Visual profiles and between some MPEG-4 Visual profiles and other video coding standards [81, 82, 83,]. Since this information may be useful to give an idea of the relative performance of MPEG-4 video coding, below some of these results are briefly reviewed (only the video profiles defined in MPEG-4 Visual Part 2 [29] are considered).

These results should be viewed with careful, since the performance of any video encoder is highly dependent on several non-normative parts. Therefore, the performance evaluation of a given video coding technology is very much conditioned by the technology itself, the content, and the coder optimizations, notably, by non-normative tools such as, pre- and post-processing, motion estimation, and, of course, rate control.

MPEG-4 VISUAL SIMPLE VERSUS H.263 BASELINE

In [83], for videoconferencing scenarios, Wiegand *et al.* compared the MPEG-4 Visual Simple Profile (SP) with the H.263 Baseline Profile [8], i.e., without optional tools.

Result: *MPEG-4 SP* outperforms *H.263 Baseline* by 16% in terms of bit rate savings.

MPEG-4 VISUAL SIMPLE VERSUS H.263 CONVERSATIONAL HIGH COMPRESSION

In the same conditions of the previous test, Wiegand *et al.*, compared also MPEG-4 SP with H.263 Conversational High Compression (CHC)²⁶ Profile, which includes coding efficiency tools, such as advanced prediction and enhanced reference picture selection (not available in MPEG-4 SP).

Result: *H.263 CHC* outperforms *MPEG-4 SP* by 2% in terms of bit rate savings.

MPEG-4 VISUAL ADVANCED SIMPLE VERSUS H.263 HIGH LATENCY

In [83], for video streaming scenarios, Wiegand *et al.*, compared the MPEG-4 Advanced Simple Profile (ASP) with H.263 High Latency Profile (HLP)²⁷, which includes, in addition to H.263 CHC Profile, also B-Pictures and reference picture re-sampling.

Result: *MPEG-4 ASP* outperforms *H.263 HLP* by 17% in terms of bit rate savings.

MPEG-4 VISUAL ADVANCED SIMPLE VERSUS MPEG-2 VIDEO MAIN

In the same conditions of the previous test, Wiegand *et al.*, compared also MPEG-4 ASP with the MPEG-2 Video Main Profile.

Result: *MPEG-4 ASP* outperforms *MPEG-2 Main* by 43% in terms of bit rate savings.

MPEG-4 VISUAL MAIN VERSUS MPEG-1 VIDEO

In [81], Sikora and Ebrahimi, reported some tests comparing the MPEG-4 Visual Main Profile

²⁶ H.263 conversational high compression profile includes Annexes D, F, I, J, L.4, T, and U [8].

²⁷ H.263 high latency profile includes Annexes D, F, I, J, K, L.4, O.1, P.5, T, and U [8].

with MPEG-1 Video for target bit rates between 40 and 768 kbit/s.

Result: *MPEG-4 Main* outperforms *MPEG-1* by 30% in terms of bit rate savings.

MPEG-4 VISUAL MAIN VERSUS MPEG-2 MAIN

In [82], Wood, compared MPEG-4 Visual Main with MPEG-2 Main for standard-definition television (SDTV), common intermediated format (CIF) and quarter common intermediated format (QCIF) and not excessively complex content.

Result: *MPEG-4 Main* outperforms *MPEG-2 Main* by 15–20% for SDTV, 20–30% for CIF and 30–50% for QCIF.

MPEG-4 VISUAL ADVANCED CODING EFFICIENCY VERSUS MPEG-4 VISUAL MAIN

In [81], Sikora and Ebrahimi, compared also the MPEG-4 Visual Main Profile with MPEG-2 Video Main Profile for target bit rates between 218 kbit/s and 1 Mbit/s using critical video test sequences.

Result: *MPEG-4 Advanced Coding Efficiency* outperforms *MPEG-4 Main* by 30–50% in terms of bit rate savings.

2.7 Final Remarks

This chapter provided an overview of the first international media representation standard relying on the concept of media objects that can be combined to build audiovisual scenes. Special attention has been devoted to the visual part of this standard, notably, the video coding architecture and the main video coding tools, in particular the non-normative rate control techniques.

As may be inferred from the flexibility, in terms of content representation, introduced by the object-based approach, it is natural that some of the traditional non-normative tools such as, for example, error concealment and bit rate control become much more complex when this object-based representation is used. Therefore, after this overview, the next chapter will be devoted to analyze the new rate control dimensions and strategies opened by this new audiovisual data representation model.

Chapter 3

Object-based Video Coding Rate

Control: A Review

3.1 Introduction

Whatever the type of coding architecture, frame-based or object-based, the rate controller is the mechanism responsible for controlling the video encoder in order that it meets relevant constraints of the encoding framework, notably, channel, delay/buffer, complexity, and quality constraints. Therefore, it has been one of the key components of any video encoder targeting compliance with any of the traditional video coding standards, such as H.261, H.263, MPEG-1 Video, and MPEG-2 Video [7, 8, 9, 10]. Additionally, due to its non-normative nature, the rate control mechanism provides one of the most challenging research areas in improving the performance of video encoders, as recent developments in the area of video coding have also shown [29, 37, 12].

Object-based rate control is, in principle, more complex than frame-based rate control due to the new rate control dimensions involved. Therefore, it is convenient to clearly identify and organize the different aspects of the problem and dissect the commonalities and main differences of frame-based and object-based rate control.

This Chapter analyzes the problem of video coding rate control fostered by the object-based video coding architecture adopted by MPEG-4, notably by highlighting the new dimensions of rate control associated to the semantic dimension of coded data. It also proposes a new framework for object-based video coding rate control where this important function of any video encoder is performed by using two levels: the scene-level rate control and the object-

level rate control.

The chapter is organized as follows: after this introduction, Section 3.2 introduces the basic objectives of video coding rate control; Section 3.3 describes the main rate control constraints; Section 3.4 analyzes the traditional frame-based rate control approach, highlighting its basic dimensions, strategies, and generic architecture; Section 3.5 analyzes the new dimensions and rate control strategies for object-based video coding, and proposes a new framework for the object-based rate control mechanism; Section 3.6 reviews the object-based rate control literature; and, finally, Section 3.7 summarizes the main conclusions and contributions of this chapter.

3.2 Video Coding Rate Control Basic Objectives

When dealing with digital video, one of the main problems faced is the large amount of data to process. Since transmission and storage media are bandwidth limited, video compression plays a very important role in enabling applications involving transmission and storage of digital video. Video coding systems achieve compression by reducing the spatio-temporal and statistical redundancies, by eliminating the irrelevancy, and sometimes by introducing a controlled degradation in the decoded pictures. Since the amount of redundancy and irrelevancy in the input sequences is variable, video coding schemes produce “naturally” output streams of variable bit rate, mainly due to the changes of activity in the sequences and to the entropy coding that exploits the statistical characteristics of the symbols produced.

This characteristic of video encoders brings some problems since the streams have to be sent through constrained media channels. While in the case of constant bit rate (CBR) channels it is required that a constant number of bits per time unit be sent to the channel, in the case of variable bit rate channels (VBR) it is required that a set of specified traffic parameters be met since otherwise data losses may occur. In both cases, it is necessary to buffer the output of the video encoder in order to absorb the natural variability of the encoder output stream. The size of the buffer typically depends on the maximum acceptable end-to-end delay: the bigger the size of the buffer, the bigger the capacity to absorb stream variations but also the bigger the maximum end-to-end delay. The rate controller is then the mechanism responsible for controlling the video encoder in order that it meets both the traditional buffer and channel constraints.

In a lossy video coding scenario where the compression ratio can be increased through an increase in the level of distortion of the coded video, the problem of rate control can be seen as a typical rate-distortion problem that may be formulated as follows [84]: *“given a source distribution and a distortion measure, what is the minimum expected distortion achievable at a particular rate? Or, equivalently, what is the minimum rate description required to achieve a particular distortion?”*

Whenever this rate-distortion formulation is valid, whatever the type of coding architecture, frame-based or object-based, a set of objectives that every rate control mechanism must fulfill can be identified. These objectives can be divided into long-term and short-term objectives as referred, for example, in [85]; below these objectives are analyzed in detail.

RATE CONTROL SHORT-TERM OBJECTIVES

Short-term objectives should be usually undertaken at the time periods of one or few video frames or even smaller time periods, such as a few MBs. These objectives can be summarized as follows:

- **Minimization of picture quality fluctuations** – Pleasant visual consumption requires that the video data is coded with approximately constant quality or, at least, with smoothly changing quality. Fluctuations in picture quality between consecutive time instants, or even inside a given picture, can be visually annoying and thus should be reduced as much as possible.
- **Minimization of buffer occupancy fluctuations** – Smaller buffer occupancy fluctuations shall allow the usage of smaller buffers, and consequently to reduce the delay and the delay fluctuation experienced by the data in the buffer. Additionally, the cost of the decoders could also be reduced but this is nowadays a less relevant objective.
- **Minimization of encoder control** – Since changes in terms of encoding modes or parameters (e.g., MB quantization parameter) may have a visually noticeable impact, e.g., due to the introduction of varying coding artifacts, the frequent control of the encoder for rate control purposes should be reduced as much as possible. This reduction should also contribute to minimize the buffer occupancy fluctuations resulting from the usage of different coding modes and to avoid possible instabilities of the encoding process.

RATE CONTROL LONG-TERM OBJECTIVES

Long-term objectives should usually deal with longer time periods, such as a few seconds or even minutes¹ (e.g., several groups of pictures). These objectives can be summarized as follows:

- **Prevention of encoder buffer overflow** – Encoder buffer overflow occurs when the encoder is producing too much coded data, which, in CBR encoding scenarios, leads to decoder buffer underflows, i.e., not enough coded data can be removed from the decoder buffer at the decoding time of a given picture. Since this always causes loss of data, encoder buffer overflows (decoder buffer underflows) must be avoided.
- **Prevention of encoder buffer underflow** – Encoder buffer underflow usually occurs in CBR channels when the encoder does not produce enough coded data to “feed the channel”, which can lead to decoder buffer overflows if nothing is done to avoid it. Encoder buffer underflows result in sub-optimal use of the channel, therefore they are not as critical as encoder buffer overflows but should also be avoided for optimal use of the available resources
- **Prevention of traffic contract violation** – In VBR networks, like asynchronous transfer mode (ATM) networks, packets transmitted outside the bounds established by the negotiated traffic contract may be discarded if traffic congestion occurs, resulting in loss of data, which must be avoided.

In conclusion, the major objectives of a rate control mechanism, whatever the coding architecture, can be summarized as:

- Regulation of the video encoder output data rate according to the channel constraints.
- Maximization of the subjective impact of the decoded video.

¹ This does not mean that the adequate rate control actions for achieving these goals could only be taken at the end of each of these periods of time.

Notice that video coding rate control is not normatively specified in any of the currently available and emerging video coding standards, since this is not necessary for interoperability. This provides of the main degrees of freedom available for manufactures to distinguish their equipment and associated performance in a compatible way.

3.3 Rate Control Constraints

Designing efficient rate control algorithms, sufficiently generic to cover a wide range of applications, although extremely desirable, is not usually easy or even possible, since the rate control requirements of different applications are many times incompatible. For example, while for interactive applications the maximum end-to-end delay should be kept low and thus the amount of buffering and the pre-processing delay has to be limited, for non interactive applications requiring good picture quality, latency can be increased in order to allow extra coding flexibility and thus pre-processing delays are not so constraining. Besides delay, the channel characteristics, such as the error and bit rate characteristics, also have great impact in the design of the rate control mechanism. Additionally, nowadays software-based video codecs are very popular. In this case, sometimes, the performance of these codecs is not limited exclusively by the end-to-end delay and the channel characteristics but also by the computational power available at the terminal.

Generically, when designing a rate control mechanism for a given application, or set of applications, the following constraints have to be taken into account: delay, bit rate, and complexity.

3.3.1 Delay Constraints

Independently of the transmission media, the total end-to-end delay for real-time video applications is usually constant since the encoder and the decoder are attached to synchronous devices, i.e., usually both the acquisition and the display devices operate at constant frame rate. However, it is possible that some systems vary the encoding/decoding frame rates due to the encoder or the decoder dropping some frames in critical conditions, e.g., eminent encoder buffer overflow or decoder buffer underflow.

Even for stored video, after decoding has started, a frame must be displayed at equally spaced time instants, corresponding to frame periods. The initial delay, corresponding to the time elapsed between the decoder receiving the first bits and the time it starts decoding, results from a trade-off between the variability in the number of bits per frame, because a variable number of bits per frame can provide better subjective quality, and the necessary decoder buffering delay to support it. The larger the variability wanted, the larger the initial delay must be in order to ensure that the decoder receives all the bits necessary for decoding each frame before the scheduled decoding time.

The rate control mechanism exploits the variability in the amount of bits per frame in order to maintain the spatio-temporal quality as constant as possible while fulfilling the buffer constraints. Thus, the initial delay or latency mainly depends on the amount of buffering in the system and the channel throughput. The greater the admissible initial delay, the greater the flexibility of the rate control mechanism.

The maximum end-to-end delay is, therefore, a factor that greatly influences the design of a video coding system, in particular the rate control mechanism. A typical two-end video coding system is composed by the following parts:

- Video encoder and decoder.
- Encoder and decoder buffers.
- Network/storage device interfaces.
- Transmission channel.

Each of these parts is responsible for introducing some delay in the overall system. Thus, the total end-to-end delay measured as the time between the instant a frame is available to be encoded at the encoder side and the instant the corresponding decoded frame is presented at the decoder side, has the following components:

- Processing delay
 - Pre-processing delay (e.g., due to picture pre-analysis, color correction, or noise reduction – encoder).
 - Post-processing delay (e.g., due to picture interpolation, or blocking and ringing artifact reduction– decoder).
 - Coding delay (e.g., picture processing during encoding – encoder).
 - Decoding delay (e.g., picture processing during decoding – decoder).
 - Algorithmic delay (e.g., picture reordering due to use of non-causal encoding – encoder & decoder).
- Buffering delay (e.g., due to the encoder and decoder buffers for bit rate smoothing).
- Network interface delay (e.g., due to packetization and depacketization).
- Transmission delay (e.g., due to the physical limitations of the transmission medium).

The amount of delay introduced by the different components of the video coding system impacts on the type of functionalities the system can support. Below several characteristics of the video coding system are analyzed with respect to their delay constraints.

INTERACTIVITY

Interactivity is an important characteristic of a video system that has great impact on its design. An interactive application is an application where one or more of the following types of interaction may occur:

- User-to-user (e.g., cooperative work with document sharing).
- User-to-machine (e.g., video on CD-ROM).
- Machine-to-user (e.g., video games).
- Conversational (e.g., videotelephony or videoconference).
- User remote control (e.g., remote monitoring and control through a back channel).

Interactivity is related with the capability of the user, when faced with a particular stimulus conveyed by the system, to react to this stimulus or, to induce some action in the system for which he/she is expecting a particular behavior.

Interactivity may impose critical delay constraints on the system. For example, in video game applications, the system must be able to react rapidly to an action taken by the user, thus decoding latency must be low.

With the new object-based coding approach, content-based interactivity plays also a relevant role in enabling improved applications where besides delay, content-based quality has also to be taken into account. In applications like videotelephony or remote monitoring, the system must be able to provide content-based quality control since, typically, for these applications, the scene content may have different subjective importance for the user.

REAL-TIME ENCODING AND DECODING

An application uses real-time encoding if acquisition and encoding take place simultaneously, notably at the same rate, in opposition to non real-time or off-line encoding where acquisition and processing times are not related, i.e., encoding can be much slower than acquisition.

Analogously, an application uses real-time decoding when decoding and presentation occur simultaneously or more generically when the scheduling decoding times are driven by the timing information in the bitstream. Non real-time decoding occurs when there is no time-constrained presentation involved, e.g., edition of pre-encoded video.

One application uses end-to-end real-time if delay is the limiting factor, i.e., if acquisition, processing, transmission and possibly use of the decoded information in the receiver occur at the same temporal rate and with a small constant delay between the different tasks.

Although in real-time encoding/decoding the transmission may not be synchronous (e.g., video transmission over ATM networks), the delay between the time a given frame is acquired at the input of the encoder and the time the same frame is available at the output of the decoder, is approximately constant. This makes delay especially important in applications involving interaction between the two ends like video teleconferencing and less crucial for applications involving no interaction like live broadcasting. If the channel delay is constant, then the maximum admissible delay for a given application can roughly be allocated between processing and buffering, thus making the choice of the buffer size essentially delay-constrained.

NON REAL-TIME ENCODING AND REAL-TIME DECODING

If only decoding has to be done in real-time (i.e., there are no timing constraints for the encoding process) delay between encoding and decoding is no longer the critical issue. An example of such system is the case of encoding for applications such as DVD video. In this example, the only contribution to the delay comes from the decoding buffer since data is read at a constant rate from the disc. In this case, the absolute delay between data reading and display will be noticeable only when the user makes use of the interactive controls such as the “Play”, “Fast-Forward”, or “Fast-Reverse” modes. Hence, the maximum delay constraint will be more flexible (e.g., delay between invoking the Play and start seeing the video).

REAL-TIME ENCODING AND NON REAL-TIME DECODING

This is the dual case of the previous one. A typical example is the edition of pre-encoded material that has been recorded in real-time and stored in coded format, instead of being displayed in some output device. This scenario is mainly limited by the amount of memory to be used in interfacing the encoder output with the storage device. Here also delay is not the primary issue.

NON REAL-TIME ENCODING AND DECODING

Non real-time problems can be essentially seen as “static” rate-constrained bit allocation problems as those studied in [86, 87]. In this case, the problem is simply the allocation of a

given number of bits to code a particular amount of data, using some performance measure, e.g., minimization of average distortion given a pre-defined number of bits to encode the data.

3.3.2 Channel Constraints

The type of channel directly connecting the encoder and decoder typically characterizes the type of video encoding (i.e., CBR, if the channels requires that a constant number of bits per time unit be sent to the channel, or VBR, otherwise). Additionally, the quality of service (QoS) provided by the network is also an important aspect characterizing the video transmission that should be taken into account during video encoding.

CONSTANT BIT RATE ENCODING

Traditionally, video has been transmitted using channels that have constant bit rate [88]. In this scenario, the channel capacity available to the end user is constant throughout the duration of the transmission. Thus buffering is necessary both at the encoder and the decoder: at the encoder in order to translate the variable bit rate output of the video encoder into the constant channel rate, and at the decoder to recover from the constant channel rate the variable number of bits per frame at the scheduled decoding times².

The main advantage of CBR encoding is reliability since the channel capacity is guaranteed throughout the duration of the connection; however, this leads to inefficient use of the available capacity since a fixed capacity is allocated regardless of the amount of information that needs to be sent. In terms of the rate control, this scenario requires a tight control of the encoding algorithm in order to meet both the channel rate and the buffer constraints.

A key aspect in a CBR scenario is the variability in terms of the number of bits per coded frame. This variability is usually dealt with by delaying the transmission through buffering in the encoder and, consequently, decoding is also delayed. The greater the maximum end-to-end delay, the greater the number of bits per frame variability allowed. A trade-off between maximum end-to-end delay and subjective quality variation must be achieved in order that the “naturally” variable bit rate compressed video is accommodated into the constant bit rate channel. Examples of CBR video encoding include: video communications over integrated services digital networks (ISDN), terrestrial and satellite digital TV broadcast operating at constant bit rate, and disk-based video storage and playback such as CD or DVD video.

VARIABLE BIT RATE ENCODING

In VBR encoding, the channel capacity available to the end user is variable, allowing data to be transmitted as it is being produced (without buffering). At the first glance, this it would seem to be the natural way to transmit video. Although this may still be true, it is not efficient from the point of view of network utilization and management that VBR networks be able to accommodate the bit rate variability of fully “uncontrolled” or “open-loop” video sources. In such scenario, resource management is virtually impossible, at least in a graceful manner, due to the unpredictability of the video source behavior. Thus, in VBR video encoding, the rate control mechanism plays also an important role in adapting the output rate of the video source to the traffic constraints of the network. In this scenario, the video encoder needs to ensure that the traffic produced is between permitted bounds, e.g., in ATM it needs to adapt its output

² In a CBR encoding scenario, the encoder buffer is filled in bursts and emptied with a constant rate while the decoder buffer is filled with constant rate and emptied in bursts.

rate to the negotiated traffic parameters. Here, two different strategies may be used in conjunction:

- **Coding rate control** – Consists in controlling the amount of data produced by the video encoder and sent to the network through the selection of the appropriate coding strategies and, consequently, affecting the quality of encoded video.
- **Data rate shaping** – Consists in smoothing the traffic generated by the video encoder by changing the schedules at which the data packets are sent to the network. Shaping the traffic of the video encoder does not change the amount of information sent and, consequently, the quality of the encoded video is not affected. This strategy can be especially useful to prevent network congestion or violation of peak rate constraints.

Contrary to what happens in CBR networks, in VBR networks the transmission delay can be variable. However, in a real-time display scenario, the information corresponding to a given video frame must reach the decoder within a certain time interval in order to be useful. Therefore, strategies involving retransmission of lost packets are of little usefulness. Thus, video schemes for VBR transmission need to be designed under the assumption that some information may be lost, which means that they must be robust to packet losses. This may be achieved in part through the use of scalable encoding schemes where each layer is sent with different priorities (if supported by the network) and appropriate error concealment techniques to mask the perceptual effects resulting from loss of information.

Relatively to the CBR scenario, VBR encoding allows extra variability in terms of the number of bits per coded frame since, besides delaying the transmission through buffering, it is also possible, in critical conditions, to exceed the average bit rate in order to process more “difficult” frames. The greater the ratio tolerated between peak bit rate and average bit rate, the greater the variability that can be achieved. A trade-off between peak bit rate, maximum end-to-end delay, and subjective quality variation must be achieved in order that the “naturally” variable bit rate compressed video be accommodated into the variable bit rate channel without violating the buffer and channel rate constraints. Thus, in a VBR encoding scenario, extra variability in terms of the number of encoded bits per frame can be achieved by allowing peaks of bit rate in critical conditions.

Because the output rate of video encoders is inherently variable, the periods of low activity of one video source can be re-used by other sources. For example, for N sources, each requiring a CBR channel of R bits/s, it might be possible to transmit them together over a single channel with rate less than $R \cdot N$ bit/s. This reduction in capacity is the so-called statistical multiplexing gain [89].

Examples of VBR transmission include most packet or cell-switched networks such as best-effort networks like IEEE 802.3 (Ethernet) and networks with quality of service (QoS) like ATM or Ethernet with QoS, where the information is split into packets or small cells that are then routed through the network and reassembled at the destination point.

BEST-EFFORT VIDEO TRANSMISSION

Typical packet-switched networks are often referred as *Best Effort* networks because they provide no guarantees on the maximum end-to-end delay or available bit rate. As a result, in a best effort environment, the received video quality may change significantly over time [90].

In this scenario, there is no previous negotiation between the user and the network about the traffic to be carried, and thus there are no guarantees that the network can accommodate the traffic. Video is carried as any other type of data, i.e., video data is packetized and routed

through the network, sharing the available transmission resources with other services. This results in the efficient use of resources since the network resources are used only when necessary but has the drawback of not providing performance guarantees.

The performance of video transmission over best-effort networks can, however, be improved if some feedback from the network is available to the encoder, e.g., the case of the real-time transport protocol (RTP) and the RTP control protocol (RTCP) [91].

A typical example of best-effort video transmission is video streaming over the Internet, which involves video transmission from a server to a client over the Internet. Different from file transfer, video can be viewed while being transmitted, i.e., without waiting for the whole file to be downloaded. Since there is usually no admission control to the network, nor traffic policing, if too many client-server connections are established, congestion may occur and the bandwidth available for each connection may be drastically decreased with the inevitable loss of data and consequent QoS decrease.

A very important aspect in best-effort video transmission, such as video streaming over the Internet, is the capability to gradually decrease the QoS of each connection, as the available resources have to be shared among an increased number of users. This can be achieved if the encoders produce scalable bitstreams and thus the available resources are allocated among several layers that together can be combined to achieve the maximum spatio-temporal quality. In critical congestion conditions, it is still possible to achieve a minimum acceptable quality by just transmitting and decoding the lower quality layers.

QoS VIDEO TRANSMISSION

Contrary to the best-effort scenario, networks with guaranteed QoS can allocate resources according to the type of data to be transmitted, enabling this way also reliable transmission of real-time services. In this scenario, either the client establishes a traffic contract with the network, as in the ATM case, or, as in the case of Ethernet with QoS, some “privileges” to access the transmission medium are given to real-time traffic sources in order to guarantee bounded access delays.

While a circuit-switched network would guarantee R bits/s for as long as the connection is active, a network with QoS may guarantee a bound on the probability that a connection requiring on average R bits/s exceeds some maximum end-to-end transmission delay (which implies some allowable loss of data). Very important aspects of QoS video transmission are:

- **QoS parameters** – Before establishing a connection, the user and the network may exchange and/or negotiate a set of QoS parameter values that each has to conform with, e.g., bit rate, delay, and data loss parameters.
- **Admission control** – Part of the negotiation between the user and the network to set up a new connection that decides if a new connection with a given set of QoS parameters can be allowed into the network with the required QoS, without degrading the QoS of the ongoing connections.
- **Policing mechanism** – The mechanism that the network uses, after a connection has been established, to monitor if a given source traffic is satisfying the negotiated parameters. The goal of this mechanism is to prevent sources from maliciously or unwillingly exceeding the traffic parameters negotiated at call set up.

3.3.3 Complexity Constraints

Although the rate-control problem can most of the times be simply seen as a rate-distortion problem, as referred in Section 3.2, typically, the theoretical rate-distortion bounds depend on arbitrarily long data blocks and, eventually, unlimited computational power. Therefore, actual rate-distortion characteristics of practical encoding systems are obtained under restricted conditions, particularly, small data block sizes, aiming at reducing the end-to-end delay and the computational power required at the encoder and, especially, at the decoder.

In this context, either implicitly or explicitly [92, 93, 94], the rate control mechanism is responsible for obtaining the best complexity-rate-distortion trade-off. This can be generically described as: minimize D , subject to $C \leq C_T$ and $R \leq R_T$, where D , C , and R are, respectively, the distortion, computational complexity, and bit rate, and C_T and R_T are, respectively, the target computational complexity, and target bit rate.

The computational complexity of typical transform-based video encoders is closely related to the type of operations involved, e.g., motion estimation, transformation, quantization, and entropy coding, and the amount and type of data to encode (see Section 4.6).

For example, in [93, 94] a feedback rate control mechanism is used to decrease the processing time of the DCT, aiming at reducing the encoding complexity to a specified target. In this case, after coding a given frame, the complexity of the frame is given as feedback to the control mechanism to take appropriate actions (e.g., decrease or increase the number of DCT coefficients sent to the decoder) in order to guarantee that the target computational complexity is kept above the specified level along the sequence. This is accomplished by skipping the DCT and quantization of blocks that are “likely” to contain all zero coefficients after quantization for the target bit rate. Notice that, since this decision is based on the pixel data and not on the transformed data, the reduction of the computation complexity is obtained at the expense of increased distortion.

The MPEG-4 Visual standard [29] also addressed explicitly the problem of bounding the bitstream decoding complexity by defining a video complexity verifier mechanism that specifies a set of rules and limits to guarantee that a given decoder compliant with a given profile@level has the necessary computational power for the decoding any bitstream or set of bitstreams compliant with that profile@level. The rate control should use this mechanism to control the encoder in order that the computational complexity bounds of a given profile@level are not violated.

3.4 Frame-based Rate Control: The YUV Dimensions

To reach the objectives presented in Section 3.2, the rate control mechanism must exploit the relevant characteristics of the input video data and conduct appropriate and timely actions that lead to the specified objectives. In the following sections, the degrees of freedom of a frame-based rate control mechanism and the corresponding actions are analyzed.

3.4.1 Rate Control Dimensions

In a frame-based video coding framework, the input data is basically organized as a sequence of rectangular matrices of luminance and chrominance samples, with a certain spatial resolution at a certain temporal rate. Moreover, each element of the luminance and chrominance matrices is represented with certain accuracy, typically using 8 bits precision –

the texture data. During the coding process, all the available texture data, or a spatially and temporally sub-sampled version, is coded with the minimum possible distortion. In this context, it is clear that the final subjective impact of the coded video sequence is related to the coded spatial resolution, the coded temporal resolution, and to the introduced distortion, i.e., the coding errors. In this context (frame-based video coding architectures), the basic dimensions of rate control are intimately related to the characteristic dimensions of the input data model, namely:

- The spatial resolution (for texture).
- The temporal resolution (for texture).
- The luminance and chrominance sample values – texture data.

In conclusion, the rate control dimensions are associated to the parameters characterizing the resolution of the data as well as the data itself, in this case pure YUV dimensions. These dimensions that fully characterize the input data model correspond to the degrees of freedom that can be exploited by the rate control mechanisms, defining different rate control strategies.

3.4.2 Rate Control Strategies

Frame-based rate control strategies are designed within the boundaries provided by the three previously identified rate control dimensions. These strategies consist usually in:

- **Spatial resolution control** – Changing the spatial resolution of the coded video.
- **Temporal resolution control** – Skipping pictures, i.e., locally changing the temporal resolution/rate of the coded video.
- **Texture data distortion control** – Introducing more or less distortion on the texture data by coding the texture values with more or less quantization error.

Each one of these rate control strategies exploits one of the rate control dimensions, each one associated to a characteristic dimension of the input data model.

In frame-based coding architectures, the rate control mechanism has the task to choose the best trade-off in terms of spatial and temporal resolution as well as to introduce distortion in the texture data, maximizing the global subjective quality. This means combining the three strategies presented above.

SPATIAL AND TEMPORAL RESOLUTION CONTROL

Reducing the spatial resolution of the input video is usually considered a “brute-force” strategy to be used in extreme conditions, such as when the output buffer is near full due to a scene change or a significant change in picture complexity, if no better alternatives are available. A frame-based video encoder can reduce its output data rate by switching the coded spatial resolution to a lower resolution, e.g., from CIF to QCIF spatial resolution. This is usually done before starting a video session (video sequence level in MPEG-1 and MPEG-2), although some standards, like H.261 and H.263, allow changing the spatial resolution at the picture level. In MPEG-4 Visual [29] this is also possible in profiles supporting the reduced resolution tool (see Section 2.4).

Changing the temporal resolution is another option available to a rate control mechanism to adapt the output data rate of the video encoder. In situations where there is not enough space left in the encoder buffer to code a given picture with “enough” quality, the rate controller

may decide to skip its coding, saving bits to better code the next pictures. In this case, at the receiver side, the display mechanism usually repeats the last decoded picture but some interpolation tools may be applied if the corresponding delay is acceptable for the application in question.

These two strategies for controlling the output data rate of a given video encoder are usually extreme solutions that can cause annoying artifacts, such as too smoothed pictures due to the spatial sub-sampling of the original data or flickering motion due to frame skipping. These two rate control strategies have a low granularity (for example, the spatial resolution is typically CIF or QCIF and nothing in between) and a high reaction delay since actions can only be taken at the picture level. The choice of the adequate spatio-temporal resolution trade-off is very important for the final subjective impact and this decision is many times left to the rate control which has to find the best temporal rate for a certain chosen spatial resolution (which is rarely changed after an initial choice). This trade-off depends, of course, on the application in question.

TEXTURE DATA DISTORTION CONTROL

The remaining dimension of rate control regards the control of the amount of distortion introduced when coding the texture data. Since in most frame-based coding schemes, the control of the introduced distortion can be made many times within a frame (e.g., for each MB), this rate control strategy provides a fast feedback in terms of coding control, although typically at the price of a few bits. Moreover, it has typically a high granularity in terms of the introduced distortion, very much depending on the way by which the distortion is controlled, e.g., number of quantization steps for DCT-based coding.

Following the brief analysis made above, it is not surprising that the most powerful, and popular, rate control strategy for frame-based video is related to the controlled introduction of distortion in the YUV values. The simultaneous use of temporal resolution and texture distortion strategies is also very popular. For hybrid coding schemes, this means that changing the quantization parameter of the DCT coefficients is the most typical way for achieving a fine control of the output bit rate.

3.4.3 Rate Control Architecture

Effective strategies for dealing with the rate control dimensions are a key tool in any frame-based rate control mechanism, e.g., choosing the spatial and temporal resolutions and the coding parameters related to the texture distortion. Figure 3.1 presents a generic architecture for frame-based rate control, highlighting three major building blocks.

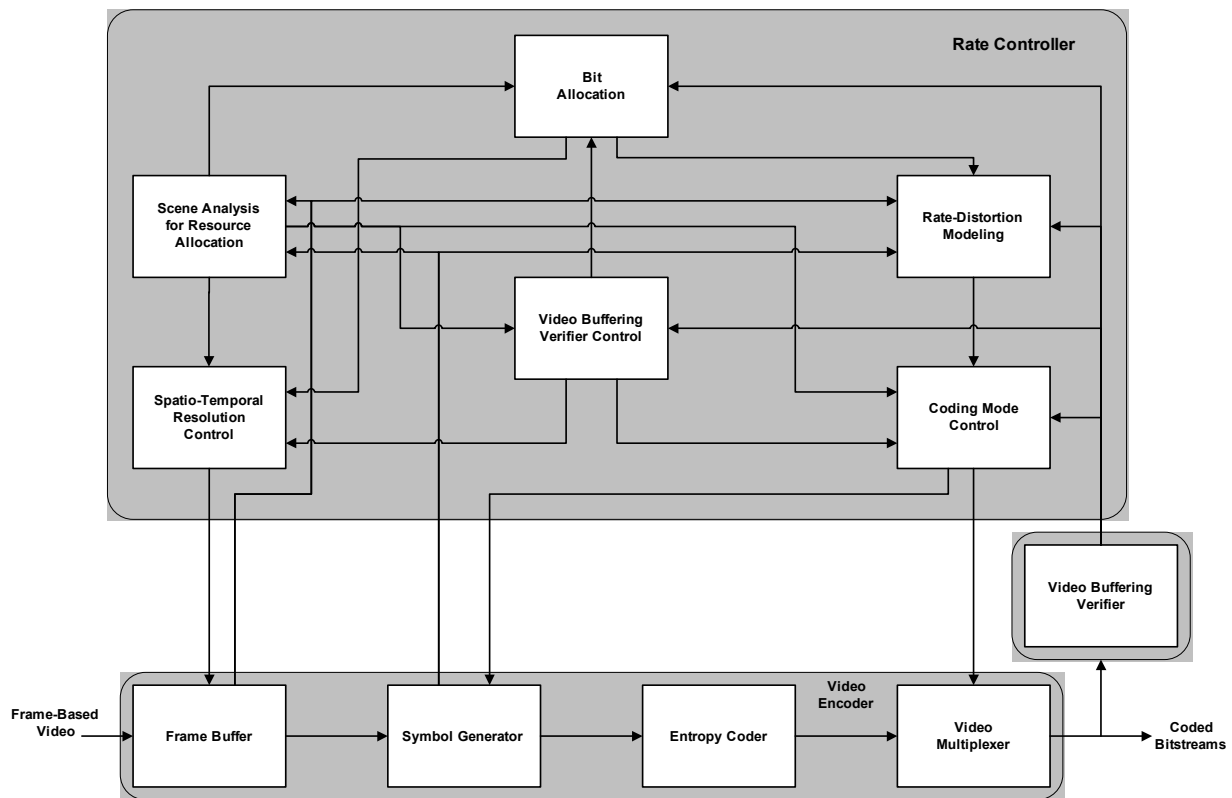


Figure 3.1 – Frame-based rate control architecture

VIDEO ENCODER

The video encoder is the mechanism responsible for encoding the original video content (i.e., the sequence of frames representing the video) into coded bits (i.e., into a bitstream or set of bitstreams³) under certain constraints, typically set by the user or by the application environment (e.g., channel bit rate and end-to-end delay). This mechanism, guided by the rate controller, can be divided into the following four modules (see Figure 3.1):

- **Frame Buffer** – Responsible for storing all the original video frames before encoding; it may store only the frame for each time instant (low-delay encoding), or it may store more frames until storing all the frames before coding any frame (off-line encoding).
- **Symbol Generator** – Responsible for reducing the redundancy and the irrelevancy to achieve compression, eventually introducing degradation in a controlled and adequate way. Converts the frame texture information into representation symbols (syntactic elements), such as motion vectors, quantized DCT coefficients, coding modes, etc. Receives as input original video frames and outputs symbols in predefined alphabets.
- **Entropy Coder** – Responsible for achieving further compression by efficiently encoding the representation symbols into bit codes exploiting their statistical redundancy, e.g., Huffman or binary arithmetic coding (BAC).
- **Video Multiplexer** – Responsible for organizing the coded symbols (and thus the associated bits) according to the video coding syntax. In case of imminent violation of the video buffering verifier mechanism, it may receive an indication from the rate

³ If scalability is used, the video encoder generates one bitstream for each layer.

controller to perform bit stuffing or skipping some coded information. Receives symbols and outputs a coded stream. In some scenarios, the output of the video multiplexer is connected to a buffer that acts as an elastic memory, being responsible for adapting the variable output data rate of the video multiplexer to the characteristics of the available transmission/storage medium (this buffer receives as input a variable rate flow and outputs a flow fulfilling the restrictions imposed by the transmission/storage medium).

VIDEO BUFFERING VERIFIER

The video buffering verifier is, typically, a normative model defining rules and limits to verify if the coded bitstreams produced by the video encoder do not violate the relevant constraints imposed by a profiling mechanism (e.g., decoder bitstream memory, and relative decoding time instants of each picture). The video buffering verifier mechanism is, therefore, responsible for defining the restrictions on the decoding complexity of a given bitstream or set of bitstreams produced by the video encoder.

Generically, the complexity of the encoded video is directly related to the encoded bit rate and to the decoded video data (YUV samples) rate that the decoder needs to process, e.g., measured in terms of the number of MB/s. For frame-based video coding, e.g., MPEG-1 Video [9] and MPEG-2 Video [10], the decoded video data rate is typically constant since the frames have fixed dimensions and are usually encoded at fixed frame rates. Therefore, in traditional frame-based video coding standards, such as MPEG-1 Video [9] and MPEG-2 Video [10], the major purpose of this mechanism was to set some restrictions on the maximum variability of the number of bits per picture, especially in the case of constant bit rate operation, and thus on the complexity of the encoded video streams.

The rate controller must use the rules and limits set by this mechanism to define the control actions that will efficiently drive the video encoder without violating this mechanism.

RATE CONTROLLER

The rate controller mechanism is responsible for controlling the video encoder aiming at efficiently encoding the original video data while producing bitstreams that meet the relevant video buffering verifier constraints. Therefore, the rate controller is responsible for driving the encoder in defining the coding parameters in order to maximize the quality of the decoded video under the constraints imposed by the application scenario (e.g., profiling mechanism constraints). This block can be further divided in several modules, performing the following tasks (see Figure 3.1):

- **Scene Analysis for Resource Allocation** – Responsible for extracting relevant information for the control process, notably the complexity/perceptual importance of each frame or subdivisions of the frame (e.g., slice or MB), in order to appropriately allocate the available resources (e.g., bit rate). Receives as input original video frames from the frame buffer and previous reconstructed encoded frames from the video encoder prediction memory.
- **Spatio-Temporal Resolution Control** – Responsible for deciding the adequate instantaneous spatio-temporal resolution for the encoded video according to some subjective quality trade-off, e.g., motion-smoothness versus spatial quality; takes into account both video characteristics, such as the motion information and texture complexity, as well as the available coding resources such as the available bit rate.

- **Rate-Distortion Modeling** – Responsible for providing a way to predict in advance the behavior of the video encoder when a certain combination of coding parameters is used (e.g., coding modes and quantization parameters). This objective is achieved through parameterized mathematical models relating the number of bits to code a given picture, the picture distortion, and the coding parameters. In order to adapt these models to the actual encoding results, after encoding⁴, the model parameters are estimated based on the recent history of encoding results, notably, the coding parameters used, the number of bits produced, and the distortion of the reconstructed frames. Prior to encoding, the coding mode control block uses these models and, eventually, some measures taken from the actual frames to code, to determine the best coding parameters, fulfilling the constraints/objectives imposed to the video encoder.
- **Bit Allocation** – Responsible for properly allocating the available bit rate; estimates by some means the number of bits to code the next frame⁵ or group of frames, according to the available bit rate and to the video buffering verifier status and control strategies. If more than one frame is involved in the estimation, it distributes the bits among them according to their relative complexity, perceptual importance, and coding constraints (e.g., I-, P-, or B-frame). Receives as input statistical data related to the scene analysis of the current picture and the video buffering mechanism status, and provides an estimation of the number of bits to code one or more frames, properly adjusted to cope with the video buffering verifier control constraints.
- **Video Buffering Verifier Control** – Responsible for controlling the video encoder in such a way that the video buffering verifier mechanism is not violated and, therefore, the bitstreams produced by the video encoder can be considered compliant with the profiling mechanism constraints selected. The main purpose of this module is to provide guidance to the other rate controller modules, notably the spatio-temporal resolution control, the bit allocation, and the coding mode control modules regarding the status of the video buffering verifier model, and consequently assist these modules in their respective tasks.
- **Coding Mode Control** – Responsible for computing the coding parameters that allow to code a frame with a given target number of bits or a given target picture quality. It must also take into account the status of the video buffering verifier mechanism: in extreme cases, where there is an imminent violation of this mechanism, the coding mode control may need to adopt extreme strategies, such as skip coding (for imminent encoder buffer overflow) or introduce stuffing data (for imminent encoder buffer underflow). Usually, this model has to deal with conflicting goals, such as avoiding to change frequently the coding parameters to favor smooth quality and not violating the video buffering verifier constraints.

3.5 Object-based Rate Control: The Semantic Dimension

The object-based video coding architecture adopted by MPEG-4 opens news dimensions and, consequently, new strategies to the rate control problem. This section analyzes the degrees of

⁴ For off-line encoding scenarios, these models can be estimated prior to encoding the video data through several encoding passes; however, for low-delay video encoding, this approach is usually unacceptable due to the computational complexity involved.

⁵ Bit allocation can be conducted at lower levels, such as MB, Block, or DCT coefficient.

freedom of an object-based rate control mechanism and the corresponding strategies and actions.

3.5.1 Rate Control Dimensions

In an object-based coding framework, such as the MPEG-4 Visual architecture [29], a scene is no longer seen as a set of rectangular frames with a given spatial and temporal resolution but, instead, as a composition of visual objects of natural and synthetic origin, typically with different characteristics and semantic relevance. Each object is independently coded, generating an elementary stream that can be independently accessed, thus providing the user the capability to access and interact with semantically meaningful objects.

In this new video representation framework, the input data model for video is characterized by some additional dimensions, in comparison with the frame-based video data model, that constitute the new dimensions of rate control, namely:

- The spatial resolution (for texture and shape).
- The temporal resolution (for texture and shape).
- The luminance and chrominance values – texture data.
- The shape data.
- The scene description data.
- The semantic relevance of each object.

In terms of data, and relatively to the frame-based scenario, there is additionally the shape data, which defines the shape of each object, and the scene description data that specifies which objects are in the scene and the way that the scene is organized, this means the way to compose the objects to re-create the visual scene. Since, typically, the objects in the scene have semantic relevance, the major novelty here is the semantic dimension of the data model and, consequently, of the rate control since it becomes possible to perform actions such as not transmitting a less relevant object to save bits for the most semantically relevant objects.

As before, the input data model dimensions are both associated to the parameters characterizing the resolution of the data – spatial and temporal resolution of the texture and shape data for each object – as well as to the data itself – texture and shape for each object and scene description.

3.5.2 Rate Control Strategies

Since the various objects in a scene are now independent entities in terms of coding, although building together a scene, the rate control dimensions presented above are dealt with by using two levels of action (see Figure 3.2):

- **Scene-level rate control** – Responsible for allocating the available resources between the objects in the scene, i.e., between the different encoding time instants and the different VOs to encode in each encoding time instant.
- **Object-level rate control** – Responsible for allocating the resources attributed to each object (in a rigid or dynamic way) among the various types of data to code (for that object), notably texture and shape, and for computing the best encoding parameters to achieve the target bit allocations while maintaining smooth quality fluctuations.

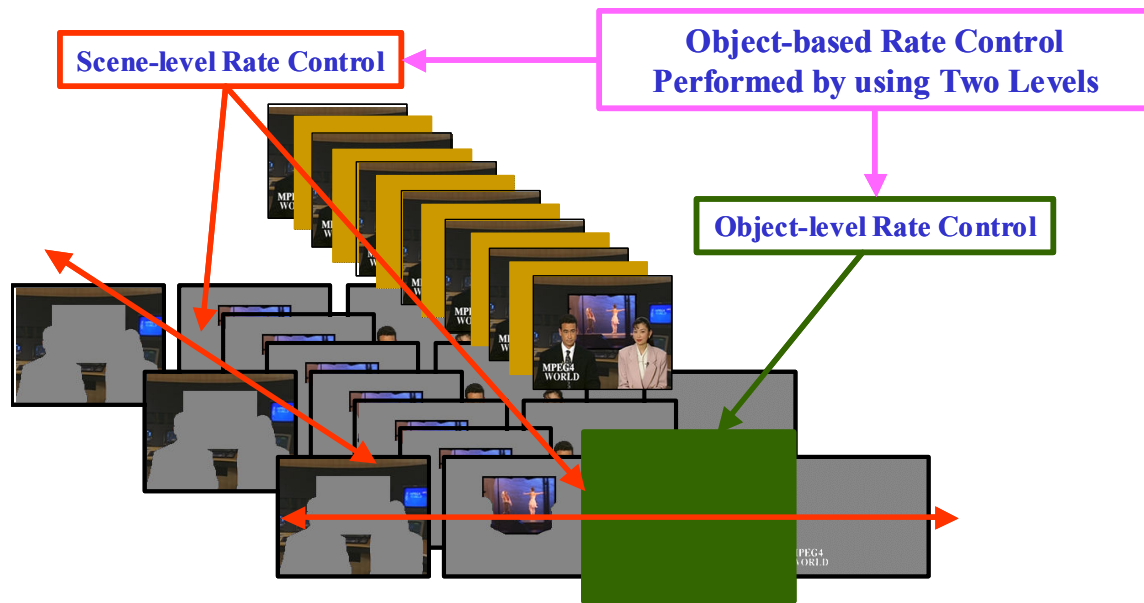


Figure 3.2 – Object-based rate control levels

To code a video scene composed by several objects, two types of scene-level rate control strategies are available:

- **Independent scene-level rate control** – Each object in the scene is coded independently of the other objects, e.g., at constant rate or constant quality, without sharing resources among objects. The total rate for the scene is the sum of the rates for each object.

For each object, a certain object-level rate control strategy is applied, with strict allocation of resources for each encoding time instant, i.e., independently of the varying characteristics of the several objects in the scene.

This is the typical situation when a scene is composed by several objects previously available as coded data, possibly even coded outside the context of any scene. In this case, no sharing of resources is possible and thus the objects have to be coded as standalone.

- **Joint scene-level rate control** – Each object in the scene is coded while dynamically sharing, along time, the total amount of resources available, without a rigid allocation of resources to each object. For each relevant time instant, the resources are distributed among the objects depending on relevant criteria, notably the object relevance, complexity, size, etc. The total rate for the scene is the sum of the rates for each object.

For each object, a certain object-level rate control strategy is applied to manage the resources dynamically distributed at each coding time instant. These resources are typically not constant in time since they are dynamically allocated among the various objects in the scene, depending on their varying characteristics.

This is the typical situation when the various objects in a scene are simultaneously coded, trying to maximize the subjective impact of the composed scene.

It is clear that all the rate control dimensions already used for frame-based architectures can now be used in the context of object-based architectures; this means that, for each object, the most adequate spatial and temporal resolutions as well as the texture distortion is looked for,

taking into account the available resources. Additionally also the best shape distortion level is looked for, keeping in mind the particular artifacts of lossy shape coding.

The division of tasks among the scene-level rate control and the object-level rate control depends on the relevant scenarios in terms of scene-level rate control: while for independent scene-level rate control all the rate control dimensions are dealt with by the object-level rate control, for joint scene-level rate control it is typically the case that the scene-level rate control decides the spatial and temporal resolutions for each object while the object-level rate control deals with the texture and shape distortion.

While it is true that frame-based rate control strategies are still useful in the context of object-based coding, there are new strategies intrinsically related to the semantic dimension of object-based representations.

In fact, if the rate controller has the task to find the best subjective impact matching between the channel constraints and the output flow, some new rate controls strategies are now possible, if a wider understanding of what subjective impact for an application means is used. The new rate control strategies are closely related to the scene description rate control dimension and can be described as:

- **Semantic resolution control** – The semantic resolution of a certain scene is related to the semantic detail provided. This means to the number of objects in which a certain amount of video data is organized (for the same amount of pixels, more objects mean higher semantic resolution and vice-versa).

Since having the texture data corresponding to a scene structured as more or less objects may have a significant impact on the resources need to code a scene, notably due to the increase of shape data, certain applications may allow the rate controller to use the scene description dimension to match the output rate to the channel constraints by merging adjacent objects in the scene, notably those less semantically relevant. In an off-line coding environment, it is even possible to imagine having the rate controller somehow signaling that the addition of more semantic resolution (more objects for the same pixels) is acceptable or not, allowing a content author to adjust the semantic resolution of a scene in order to fulfill the channel limitations while simultaneously achieving in the best way the creation purposes of the author.

An example is a remote surveillance codec using a low bandwidth channel, and providing more than two objects for coding, one of them of high priority and the others of low priority. In this case, it may be acceptable that the rate controller, in case of dramatic need, be allowed to decrease the semantic resolution to cope with the channel limitations by merging some of the lower priority objects (of course, this does not have to be performed at every encoding time instant). In this context, the semantic resolution plays a role similar to the spatial and temporal resolutions, depending on the application to accept or prevent that the rate control mechanism exploits certain rate control dimensions.

- **Amount of content control** – The amount of content in a scene is related to the number of objects and to the corresponding amount of data. Decreasing the amount of content means reducing the number of objects in the sense that the pixels corresponding to a certain object are removed (not in the sense that objects are merged). This approach is more adequate for scenes with overlapped objects, since for segmented scenes it can lead to annoying artifacts, if no additional scene processing is done, due to the mismatches in the shape of the objects.

Since having more or less content (objects) in a scene may have a significant impact on the resources need to code a scene, notably due to the increase of texture and shape data, certain applications may allow the rate controller to use the scene description dimension to match the output rate to the channel constraints by not coding the less important objects in the scene.

An example is a mobile videophone using a GSM channel, and providing two objects for coding, one of them of high priority and the other of low priority (foreground and background). In this case, it may be acceptable that the rate controller, in case of dramatic need, be allowed not to code the low priority object, saving resources for the high priority object (and by this maximizing the subjective impact).

This is a completely new approach in terms of rate control since it implies for the first time that the scene content is not coded at all and not just coded with less quality. However, for object-based coding frameworks, this rate control dimension is not only conceptually possible but also useful for certain type of applications.⁶

Notice that, for object-based coding, the relevant criteria to be used for rate control are not only related to the texture and shape characteristics of each object but also to their semantic dimension, this means to the priority and relevance of each object in the context of the scene and the application.

In object-based coding architectures, the rate control mechanism has the task to choose the best trade-off in terms of the amount of content to be coded, the corresponding semantic resolution, the spatial and temporal resolution for each object, and the amount of distortion in the texture and shape data for each object, in order that the global subjective impact in terms of the relevant requirements is maximized.

It does not require a lot of analysis to conclude that object-based rate control is, in principle, much more complex than frame-based rate control due to the new rate control dimensions involved.

3.5.3 Rate Control Architecture

The analysis made above in terms of the rate control dimensions and strategies for object-based coding architectures highlights that the rate controller may have to use new approaches to reach its objectives, notably implying interfacing with new modules besides the symbols generator (source encoder) as was the case for frame-based coding.

Figure 3.3 presents an object-based coding framework, presenting the interfaces of the rate control module. This framework shows that the rate controller, besides including the typical interfacing with the symbols generator and the video multiplexer, interfaces with the video analysis and with the scene authoring to impact on the semantic resolution and the amount of content to code. Depending on the applications, two relevant cases in terms of rate control interfacing may be relevant:

- **Real-time applications** – In terms of the semantic rate control dimensions, the rate control only interfaces with the automatic video analysis module, for example to ask for the most convenient merging of two objects to decrease the needed coding

⁶ Those that do not like the idea of having the rate control touching the semantic dimensions, should remind that if, for example, a scene has too many objects (think just about 10), it may much better to merge a few of them or even to delete a few of them than having everything coded with an incredibly poor quality.

resources. The authoring block in Figure 3.3 does not exist in this case.

- **Off-line applications** – In terms of semantic dimensions, the rate control interfaces both with the automatic or semi-automatic video analysis and the authoring modules. The interface with the authoring module provides feedback about the coding of the current scene (notably what is happening in terms of the various rate control dimensions), allowing the author to decide if that is according to his/her purposes or if something different should be done, e.g., to improve the quality of the most relevant objects.

The reasoning here is that it may not make sense to allow the rate control to proceed in an automatic way, if an author to whom it is possible to provide rate control feedback is available, notably when more drastic decisions in terms of rate control are involved. The author will then be able to take the decisions that best fit his/her content creation purposes.

This type of interfacing may be extremely useful when creating content for a specific MPEG-4 video profile and level combination where the bit rate is limited. Providing the author some feedback before acting on the semantic dimension or taking very drastic actions on the YUV and shape dimensions may be a wise decision since he/she is the one that should know better what prefers to do (for example, more objects with low quality or less objects with better quality).

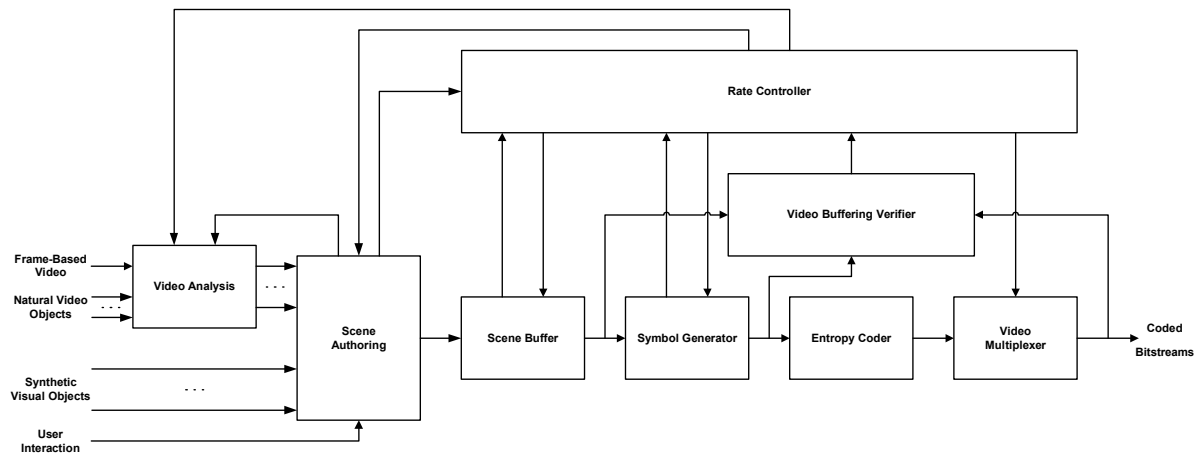


Figure 3.3 – Object-based rate control framework

3.6 Review of Object-based Bit Rate Control Methods

The purpose of this section is to review the object-based bit rate control techniques proposed in the literature. Up to the time of writing this Thesis, the most well known object-based video coding framework is provided by the MPEG-4 Visual standard [29]; therefore it is natural that most of the rate control techniques proposed in the literature for object-based rate control are essentially targeting this video coding standard.

Since there is a vast literature available on frame-based video coding rate control and this is not the main topic of this Thesis, this section will be mainly devoted to review the rate control techniques that have been proposed in the context of object-based rate control, with special emphasis on multiple video object (MVO) video encoding rate control, although some single video object (SVO) techniques are also reviewed due to their historic or technological

relevance.

3.6.1 Telenor⁷ SVO Rate Control

The problem of rate control in the MPEG-4 video coding framework started earlier during the preparation for the 1995 MPEG tests [95]. In this context, Bjontegaard, Lillevold, and Danielson proposed a simple rate control technique for generating the frame-based anchor test sequences to which the technical proposals were to be compared [96, 97]. This technique has been adopted in the MPEG-4 Video Verification Model 4.0 (VM4) [98] for SVO encoding.

Essentially, this technique aims at controlling the output bit rate of the video encoder by increasing or decreasing the VOP quantization parameter for each encoding time instant based on the deviations of the number of bits produced in the last encoding time instant relatively to a given target bit allocation:

- If too many bits have been spent in the previous encoding time instant, the quantization parameter is increased by a given amount.
- If too few bits have been spent in the previous encoding time instant, the quantization parameter is decreased by the same amount.

The number of bits to encode a given video object sequence of duration t_{SEQ} is given by the following equation

$$T_{SEQ} = R \times t_{SEQ} \quad (3.1)$$

where R is the target channel bit rate.

The target number of bits for each encoding time instant is given by

$$T[i] = \frac{T_{SEQ} - \sum_{k=1}^{i-1} S[k]}{N - i + 1} \quad (3.2)$$

Based on the deviation between the target number of bits for the current time instant $T[i]$ and the number of bits spent in the previous encoding time instant $S[i-1]$, the VOP quantization for the current time instant is updated according to the following equation

$$Q[i] = \begin{cases} \min[31, Q[i-1] + \Delta Q[i]] & \Leftarrow S[i-1] > T[i] \times \beta_T \\ Q[i-1] & \Leftarrow T[i]/\beta_T \leq S[i-1] \leq T[i] \times \beta_T \\ \max[1, Q[i-1] - \Delta Q[i]] & \Leftarrow S[i-1] < T[i]/\beta_T \end{cases} \quad (3.3)$$

where

$$\Delta Q[i] = \max[1, \beta_Q \times Q[i-1]] \quad (3.4)$$

with $\beta_T = 1.15$ and $\beta_Q = 0.1$.

This simple algorithm is able to roughly control the output bit rate for a single video object encoder, although not very accurately since there is no MB-level adaptation of the quantization parameter. Moreover, no buffer control is performed.

⁷ Telenor Research & Development, Oslo, Norway.

Although this algorithm was proposed only for single video object encoding, it was already recognized in [97] that the problem of multiple video object encoding was complex, notably because the different VOs composing the scene could change their sizes along time and could be encoded at different temporal resolutions.

3.6.2 Sarnoff⁸ SVO and MVO Rate Control

The problem of rate-distortion modeling for accurate bit allocation, in the context of MPEG-4 video encoding, was first addressed at David Sarnoff Research Center, by Chiang⁹ and Zhang¹⁰ [99, 100, 101]. In this work, the authors propose the quadratic rate-quantization model that has been adopted in MPEG-4 Video VM5 [102] for the frame and multiple video-object rate control algorithms (see Section 2.5).

Using a different set of model parameters for each picture coding type (i.e., I, P, and B) and based on the previous encoding time instants, the authors propose to use this rate-quantization model at the picture and MB level to allocate the available bit rate before really encoding the video data.

Lee¹¹ *et al.* [103, 104] use this rate-quantization model for SVO and MVO rate control. To tackle the problem of MVO encoding, notably, to distribute the target number of bits allocated for each encoding time instant among the several VOs to encode, the authors use the following complexity weight

$$\omega[i] = \frac{MAD^2[i]}{\sum_{k=1}^{N_{VO}} MAD^2[k]}, \quad i = 1, \dots, N_{VO} \quad (3.5)$$

where MAD is the mean absolute difference of the luminance component between original and the reference VOP.

This is a simple criterion for bit allocation among the several VOs composing a scene, since it does not take into account the size and activity of the VOs as the work presented by Nunes and Pereira in [14] and Vetro *et al.* in [105].

In [103, 104], Lee *et al.* also proposed a rate control method for the shape data, where, basically, if the target number of bits for the current encoding time instant is higher than the estimated bits used for coding the syntax, motion, and shape information, then the threshold for controlling the shape accuracy (see Section 2.4) is increased by α_{step} ; otherwise, it is decreased by the same amount, i.e.,

$$\alpha_{TH}[i] = \begin{cases} \min[\alpha_{\max}, \alpha_{TH}[i-1] + \alpha_{\text{step}}] & \Leftarrow T[i] \leq E[S_{\text{syntax}}[i] + S_{\text{motion}}[i] + S_{\text{shape}}[i]] \\ \max[0, \alpha_{TH}[i-1] - \alpha_{\text{step}}] & \Leftarrow T[i] > E[S_{\text{syntax}}[i] + S_{\text{motion}}[i] + S_{\text{shape}}[i]] \end{cases} \quad (3.6)$$

In [103, 104], the authors do not mention the typical values of α_{\max} , and α_{step} , although proposing a conservative approach, recognizing that the lossy encoding of shape information

⁸ David Sarnoff Research Center (DSRC), Princeton, NJ, USA; today Sarnoff Corporation.

⁹ T. Chiang is now with National Chiao Tung University, Taipei, Taiwan, R.O.C.

¹⁰ Y.-Q. Zhang is now with Microsoft Research, USA.

¹¹ H.-J. Lee is now with Multimedia Technology Laboratory, Sarnoff Corporation, Princeton, NJ, USA.

can be subjectively very annoying, as already pointed by Nunes *et al.* in [106]. As referred in Section 2.4, in the context of MPEG-4 video rate control, $\alpha_{\max} = 64$ and $\alpha_{\text{step}} = 4$.

3.6.3 Mitsubishi¹² MVO Rate Control and Related Work

In 1998, Vetro¹², Sun¹³, and Wang¹⁴ [105, 107] proposed a multiple video object rate control algorithm extending the work of Chiang and Lee [101, 103] to the MVO case. This algorithm (multiple video-object rate control algorithm described in Section 2.5) was adopted by MPEG in the MPEG-4 Video VM8 [108].

Although object-based video encoding architectures can easily and naturally support the coding of multiple video objects with different temporal resolutions, the Telenor [96, 97], Sarnoff [101, 103], and Mitsubishi [105, 107] rate control algorithms assume that all VOs in the scene are encoded at the same temporal resolution. In fact, the problem of encoding multiple video objects with different temporal resolutions was first tackled in the context of this Thesis [26].

In [109], Lee¹⁵, Vetro, Wang, and Ho¹⁴, also address this problem, considering explicitly the rate-distortion characteristics of coded and skipped VOPs in order to enable a trade-off between spatial and temporal qualities. In this context, the idea is to minimize the average pixel distortion over a time interval $[t_{i+1}, t_{i+f_s}]$, $\bar{D}_{[t_{i+1}, t_{i+f_s}]}(Q_{i+f_s}, f_s)$, considering the distortion of the coded VOP at time instant t_{i+f_s} , $D_C(Q_{i+f_s}, f_s)$, and the distortion of the $f_s - 1$ skipped VOPs, $D_S(Q_i, k)$, i.e.,

$$\bar{D}_{[t_{i+1}, t_{i+f_s}]}(Q_{i+f_s}, f_s) = \left[D_C(Q_{i+f_s}, f_s) + \sum_{k=i+1}^{i+f_s-1} D_S(Q_i, k) \right] \quad (3.7)$$

The distortion of the coded VOPs is modeled according to (3.8)

$$D_C(Q_i) = a \cdot 2^{-2R(t_i)} \cdot \sigma_{z_i}^2 \quad (3.8)$$

where a is a model parameter, $R(t_i)$ is the average bit rate per sample at time instant t_i , and $\sigma_{z_i}^2$ is the variance of the input signal.

The distortion of skipped VOPs is modeled according to (3.9)

$$D_S(Q_i, k) = D_C(Q_i) + E[\Delta_{z_{i,k}}^2] \quad (3.9)$$

where $D_C(Q_i)$ is the distortion of the last coded VOP and $E[\Delta_{z_{i,k}}^2]$ is an estimate of the interpolation mean square error for the skipped VOP at time instant t_k .

¹² Mitsubishi Electric Information Technology Center America (MEITCA), New Providence, NJ, USA.

¹³ A. Vetro and H. Sun were formerly with the Advanced Television Laboratory of MEITCA, New Providence, NJ, USA, and are now with Mitsubishi Electric Research Laboratories, Cambridge, MA, USA.

¹⁴ Y. Wang is with the Department of Electrical Engineering, Polytechnic University, Brooklyn, NY, USA.

¹⁵ J.-W. Lee and Y.-S. Ho are with the Department of Information and Communications, Kwangju Institute of Science and Technology, Kwangju, Korea.

After each encoded VOP, the algorithm computes the number of skipped encoding time instants, f_s , that minimizes (3.7) by setting the trade-off between spatial and temporal quality. Notice that, by skipping one or more VOPs after each encoding time instant, the number of available bits for the next encoding time instant increases allowing improving the spatial quality of the next coded VOP.

In [109], the authors consider two forms of the spatio-temporal quality trade-off:

1. **Constrained temporal rates** – All VOs are encoded at the same temporal resolution, i.e., for each target encoding time instant, the algorithm decides between coding or skipping all VOPs of the current time instant.
2. **Unconstrained temporal rates** – The different VOs composing the scene can be encoded at different temporal resolutions, i.e., first the algorithm selects the next encoding time instant, t_{i+f_s} , based on buffer occupancy restrictions and, afterwards, selects the set of VOPs to be encoded for that time instant, minimizing the distortion for that encoding time instant.

This work exhibits the interesting idea, of taking into account also the rate-distortion characteristics of skipped VOPs. A possible drawback is the efficient estimation of the skipped frames distortions.

3.6.4 Sharp MB Rate Control

Low-delay video encoding requires a fine control of the encoder bit production since the buffering delay needs to be kept low. In this context, Ribas-Corbera¹⁶ and Lei¹⁷ proposed a MB rate control that allows a fine control of the encoder output bit rate through a fine control of the MB quantization parameter based on rate-distortion modeling at the MB level [110, 111]. This rate algorithm (described in Section 2.5) was adopted by the MPEG-4 Video VM8 [108].

In [112], the authors extend their work proposing a frame-level bit allocation, in the context of the H.263+ developments [113], which intends to balance bit allocations between P- and B-frames in order to achieve uniform quality between these types of frames.

For a group of pictures with M_B B-frames and one P-frame, the target number of bits for each frame coding time is given as follows:

$$T_B = \frac{T - (H_P - \beta H_B)}{\beta + M_B} \quad (3.10)$$

$$T_P = T - M_B T_B$$

where T is the target number of bits for the current group of pictures; H_P and H_B are the number of syntax and motion bits for each frame coding type estimated from the previous encoded frames of the same coding type; and β is a control parameter defined by the ratio between the frame coding type energies, i.e.,

¹⁶ J. Ribas-Corbera was formerly with the Department of Digital Video, Sharp Labs of America, Camas, WA, USA, and is now with Microsoft Corporation, Redmond, WA, USA.

¹⁷ S. Lei is with the Department of Digital Video, Sharp Labs of America, Camas, WA, USA.

$$\beta = (0.9 + 0.1F) \frac{S_P}{S_B} \quad (3.11)$$

where S_P and S_B are the sum of the MB variances, respectively, for the P- and B-frames computed from the last encoded P- and B-frame, and F is an empirically set adjustment parameter that intends to favor the spatial quality of P-frames relatively to B-frames (since P-frames are used as references for B-frames, increasing the quality of P-frames typically allows to increase the average quality of each group of pictures). In [112], F is computed as

$$F = \frac{1.35}{\sqrt{Bpp}} - 0.4 \quad (3.12)$$

where Bpp is the average target number of bits per pixel; F given by (3.12) is clipped between 1 and 5.

Although, the Sharp MB rate control algorithm can achieve accurate bit allocations, it is based on the frequent change of MB quantization parameter. Therefore, large variations of the quantization parameter inside each VOP may occur with consequent large variations in the VOP spatial quality.

3.6.5 Universidad Politécnica de Madrid¹⁸ MVO Rate Control

Ronda¹⁹ *et al.* addressed the problem of multiple video object encoding as the optimization of a given cost criterion based on video quality metrics [114, 78]. In their work, the authors propose a rate control algorithm able to fulfill different rate control goals, operating under the following restrictions:

- Synchronous VOs with the same temporal resolution, i.e., with the same VOP period T_0 , and with each encoding time instant $t_n = n \cdot T_0$, $n = 0, \dots, N-1$.
- VOP-level quantization parameter, i.e., no MB quantization parameter adjustment.
- Real-time constant bit rate encoding.

In this context, the authors proposed three rate control variants with different goals and associated different cost criteria, namely:

1. **Weighted distortion rate control** – Aims at minimizing the weighted distortion of the various VOs in the scene, embedding VO priorities in the optimization process, i.e., the rate control goal becomes the minimization of the weighted sum of the individual VO distortions without imposing any relation between the different VOs in the scene.
2. **Priority-based rate control** – Aims at satisfying a prioritized list of VO target distortions, specifying minimum qualities for the various VOs, ordered according to their priority, i.e., the rate control goal is to achieve the minimum target quality specified by the user for each VO (for each encoding time instant), respecting the prioritized order.
3. **Constant distortion-ratio rate control** – Aims at achieving a constant distortion ratio

¹⁸ Universidad Politécnica de Madrid (UPM), Madrid, Spain.

¹⁹ The authors are with the Grupo de Tratamiento de Imágenes, UPM, Madrid, Spain.

for each VO, relatively to a reference VO, i.e., given a reference VO and a set of distortion ratios specified by the user, the rate control goal is to minimize the average VOs distortion, respecting these distortion ratios.

For each of the above rate control goals, the authors define a cost function that should be minimized.

WEIGHTED DISTORTION COST FUNCTION

In this case, for each encoding time instant, t_n , the rate control goal is to minimize the following function

$$\bar{D}[n] = \sum_{i=1}^{N_{VO}} \alpha[i] \cdot D[n][i] \quad (3.13)$$

under the constraint

$$\sum_{i=1}^{N_{VO}} R[n][i] \leq R_T[n] \quad (3.14)$$

where $\alpha[i]$ is the VO i weight; $D[n][i]$ and $R[n][i]$ are, respectively, the estimated distortion and estimated number of bits of VO i for time instant t_n computed through the rate-quantization and distortion-quantization models of VO i ; and, $R_T[n]$ is the target number of bits for the given encoding time instant. Finally,

$$D[n][i] = p[n][i] \cdot d[n][i] \quad (3.15)$$

$$R[n][i] = p[n][i] \cdot r[n][i] \quad (3.16)$$

where $p[n][i]$, $d[n][i]$, and $r[n][i]$ are, respectively, the number of non-transparent pixels, the average pixel distortion, and the average number of bits per pixel of VO i in time instant t_n .

The constrained problem defined by (3.13) and (3.14) can be solved using the Lagrange multiplier technique (see Section 5.2) with the following formulation²⁰

$$\text{minimize } \sum_{i=1}^{N_{VO}} \alpha[i] \cdot D[i] + \lambda \sum_{i=1}^{N_{VO}} R[i] \quad (3.17)$$

Solving the problem specified by (3.13) and (3.14) requires iteratively changing λ and solving (3.17) for each λ , regarding the VOP quantization parameter for each VO, until (3.14) is met.

PRIORITY-BASED COST FUNCTION

In this case, the rate control goal is specified through a prioritized list of target distortions of the following form

$$L_P = \{(id_1, \bar{d}_1), (id_2, \bar{d}_2), \dots, (id_k, \bar{d}_k)\} \quad (3.18)$$

where id_k represent VO ids and \bar{d}_k represents the target distortion for the given VO.

²⁰ For simplicity, from this point onwards, the time instant index will be dropped.

In order to accomplish the rate control goal specified by (3.18), given the target number of bits for the corresponding the coding time instant, R_T , the rate control algorithm should process sequentially each element of the list, performing the following operations:

- Allocate to each VO, for the given encoding time instant, the minimum number of bits to achieve the target quality, if such number of bits is available.
- If the required number of bit is not available, allocate the remaining number of bits to the given VO and zero to the remaining VOs.

It is important to notice that this rate control approach requires the user to specify an adequate list of target distortions since otherwise the objectives specified may be impossible to fulfill.

CONSTANT DISTORTION-RATIO COST FUNCTION

In this case, the rate control goal is to minimize the distortion of a reference VO (for simplicity, it may be considered that VO 1 is the reference VO) under the constraints of maintaining an approximate constant distortion-ratio for each other VO and not exceeding the target number of bits for the given encoding time instant, i.e.,

$$\text{minimize } D[1] \quad (3.19)$$

under the constraints

$$d[i] = \beta[i] \cdot d[1] \quad (3.20)$$

$$\sum_{i=1}^{N_{VO}} R[i] \leq R_T \quad (3.21)$$

where VO 1 is the VO selected as reference and $\beta[i]$ are the user selected distortion-ratios, which become the main goals of the rate control algorithm.

To overcome the impossibility to meet (3.20) under a discrete set of coding parameters, (3.20) can be converted in

$$d[i] \leq \beta[i] \cdot d[1] \quad (3.22)$$

In order to fulfill simultaneously (3.19), (3.20), and (3.21), the rate control algorithm needs to perform the following operations for each encoding time instant and for each possible quantization parameter value of the reference VO in decreasing order:

- Obtain the distortion and number of bits for the reference VO (VO 1).
- For each of the other VOs, compute the quantization parameter, which provides a distortion most closely approaching $\beta[i] \cdot D[1]$ and the corresponding number of bits $R[i]$.
- Sum the number of bits for all VOs; if it is lower than the global target number of bits, R_T , the optimum assignment has been obtained.

Since it is not always possible to achieve the target distortion $D[1]$, an alternative formulation can be defined as

$$\text{minimize } k_1 \sum_{i=1}^{N_{VO}} D[i] + k_2 \sum_{i=2}^{N_{VO}} |d[i] - \beta[i] \cdot d[1]| \quad (3.23)$$

under the constraint (3.21) where k_1 and k_2 are nonnegative constants that allow balancing the importance of (3.19) relatively to (3.20).

Similarly to the priority-based control, the problem specified by (3.19), (3.20), and (3.21) can be converted into an unconstrained minimization problem of the following form

$$\text{minimize } k_1 \sum_{i=1}^{N_{VO}} D[i] + k_2 \sum_{i=2}^{N_{VO}} |d[i] - \beta[i] \cdot d[1]| + \lambda \sum_{i=1}^{N_{VO}} R[i] \quad (3.24)$$

In this case, finding a solution for (3.24) involves a higher degree of complexity than for the priority-based control, since an exhaustive search of the quantization parameter for the reference VO needs to be done for each value of λ .

Notice that, similarly to the previous algorithm formulation, it is not easy for the common user to specify adequate values of $\beta[i]$, k_1 , and k_2 , for a given set of VOs composing a given scene, which makes this algorithm not straightforwardly usable.

RATE-DISTORTION MODELING

In order that the minimization of the above cost functions can be done in real-time with reasonable computational power (i.e., avoiding multiple encoding passes for each VOP), the rate and distortion for each possible quantization parameter for each VO are estimated through the following rate-quantization and distortion-quantization models:

$$r(Q) = \left(a_0 + a_1 \frac{1}{Q} + a_2 \frac{1}{Q^2} \right) \cdot m \quad (3.25)$$

$$d(Q) = b_0 + b_1 Q + b_2 Q^2 \quad (3.26)$$

where a_0 , a_1 , a_2 , b_0 , b_1 , and b_2 are the model parameters (a different set for each VO) estimated from past encoding results, at the end of VOP encoding, and m is the MAD between the original and the prediction VOPs.

Notice that (3.25) and (3.26) represent, respectively, the average number of texture bits per pixel and the average pixel distortion obtained when encoding a given VO VOP with quantization parameter Q .

BIT ALLOCATION

Independently of the rate control goal, the UPM rate control algorithm allocates a given number of bits, $R_T[n]$, for each encoding time instant, t_n , according to a buffer control mechanism targeting a buffer occupancy of $B_s/2$ before each encoding time instant, where B_s is the encoder rate buffer size. This bit allocation task involves the following five steps:

Step I – Skipping Control

Before allocating bits for a given encoding time instant, the UPM rate control algorithm assesses the occupancy of the encoder rate buffer. If $B \geq 0.8 \times B_s$, the encoding time instant under consideration is skipped; otherwise, the rate control algorithm proceeds to the next step.

Notice that, since there is no MB quantization parameter adjustment in this algorithm, the overflow margin needs to be relatively high (20 % of the buffer size) as a less adequate choice of the VOP quantization parameter (i.e., too low quantization parameter) for some VOs can

lead to imminent encoder buffer overflow without possible correction. This is one of the main reasons why the rate control proposed in this Thesis (see Chapter 6) provides quantization parameter adjustment and VBV control at the MB level for a more accurate control of the VBV occupancy.

Step II – Target Number of Bits for each VOP Coding Type Computation

Still before really allocating the number of bits for the next encoding time instant, the UPM algorithm computes the target number of bits for each VO VOP coding type according to the following equations

$$R_I[i] = \eta \bar{R}_I[i], R_P[i] = \eta \bar{R}_P[i], R_B[i] = \eta \bar{R}_B[i] \quad (3.27)$$

where $\bar{R}_I[i]$, $\bar{R}_P[i]$, $\bar{R}_B[i]$ are the average number of bits used for each VOP coding type in the previous encoding time instants computed over a sliding window, $w = 5$, for VO i , and η is an adjusting parameter computed by the following equation

$$B[n] + \eta(1 - p_{\text{skip}}) \sum_{k=n}^{n+M-1} \sum_{i=1}^{N_{VO}} \bar{R}_{T[k][i]}[i] - M \cdot R_C = \frac{B_S}{2} \quad (3.28)$$

where p_{skip} is the frame skipping probability estimated from recent observations, $\bar{R}_{T[k][i]}$ is the target number of bits for the corresponding VOP coding type of VO i at time instant t_k , M is a look ahead window (typically, coincident with the GOV size), and R_C is the number of bits drained from the encoder rate buffer to the channel during successive target encoding time instants.

Notice that, although after I-VOP encoding the occupancy of the encoder rate buffer is typically higher than after P- or B-VOP encoding, (3.28) does not take into account the VOP coding type relative positions for setting the target buffer occupancies, forcing typically the encoder rate buffer occupancy to be around $B_S/2$, which tends to penalize the VOPs immediately after the I-VOPs.

Step III – Buffer Overflow Control

If the value of η obtained through (3.28) leads to imminent encoder rate buffer overflow (i.e., $B[n] \geq 0.8 \times B_S$ after the bits from the current time instant have been added to the encoder rate buffer) for any encoding time instant $t \in [t_n, t_{n+M-1}]$, η is recomputed such that

$$B[n] + \eta \sum_{k=n}^{n+l} \sum_{i=1}^{N_{VO}} \bar{R}_{T[k][i]}[i] - l \cdot R_C < 0.8 B_S, l = 0, \dots, M-1 \quad (3.29)$$

Step IV – Target Number of Bits for Current Encoding Time Instant Computation

The target number of bits for the current encoding time instant is, therefore, computed as

$$R_T[n] = \sum_{i=1}^{N_{VO}} \eta \bar{R}_{T[n][i]}[i] \quad (3.30)$$

Step V – Buffer Underflow Control

If the target number of bits for the time instant t_n results in encoder buffer underflow (i.e.,

$B[n] + R_T[n] - R_C < 0$), then the target number of bits for the current time instant is adjusted as follows

$$R_T[n] = R_C - B[n] \quad (3.31)$$

VOP QUANTIZATION PARAMETER COMPUTATION

After computing the bit allocation for the given encoding time instant, the UPM rate control algorithm computes the VOP quantization parameter for each VO through the cost functions defined for each rate control goal.

I) Weighted Distortion Rate Control

Requires solving equation (3.17), which means finding the set of quantization parameters $Q[i]$, $i = 1, \dots, N_{VO}$ minimizing (3.17) subject to the restriction (3.14), i.e.,

$$\arg \min_{Q[i] \in \{Q_{\min}, Q_{\max}\}} \left[\sum_{i=1}^{N_{VO}} \alpha[i] \cdot p[i] \cdot d_i(Q[i]) + \lambda \sum_{i=1}^{N_{VO}} p[i] \cdot r_i(Q[i]) \right] \quad (3.32)$$

II) Priority-based Rate Control

In this case, the rate control algorithm successively assigns bits to each VO (from the target bits allocated for the given encoding time instant) in order that the objectives specified through (3.18) are achieved respecting the prioritized order, or the bit allocation for that encoding time instant is exhausted. The algorithm starts by assigning the highest quantization parameter to each VO. If the estimated total number of bits is higher than the target number of bits for the current time instant, the VOP quantization parameter computation is finished and all VOPs are encoded with this quantization parameter.

III) Constant Distortion-Ratios Cost Function

Requires solving equation (3.24), which means finding the set of quantization parameters $Q[i]$, $i = 1, \dots, N_{VO}$ minimizing (3.24) subject to the restriction (3.21), i.e.,

$$\arg \min_{Q[i] \in \{Q_{\min}, Q_{\max}\}} \left[k_1 \sum_{i=1}^{N_{VO}} p[i] \cdot d_i(Q[i]) + k_2 \sum_{i=2}^{N_{VO}} |d_i(Q[i]) - \beta[i] \cdot d_1(Q[1])| + \lambda \sum_{i=1}^{N_{VO}} p[i] \cdot r_i(Q[i]) \right] \quad (3.33)$$

The UPM proposals provide a useful framework for priority-based rate control, however they lack an efficient mechanism for compensating the deviations between the rate-distortion models and the actual encoding results. This is one of the main topics also addressed in the context of this Thesis, i.e., the compensation mechanisms for dealing with this type of deviations (see Chapter 6).

3.6.6 University of California at Santa Barbara²¹ SVO and MVO Rate Control via ρ -Domain Source Modeling

He²² and Mitra²³ proposed a new framework for modeling the rate-distortion characteristics of

²¹ University of California, Santa Barbara (UCSB), Santa Barbara, CA, USA.

²² Z. He was formerly with the Department of Electrical and Computer Engineering, UCSB, Santa Barbara, CA, USA, and he is now with Sarnoff Corporation, Princeton, NJ, USA.

DCT video encoders where the coding rate and the pixel distortion are modeled as functions of the percentage of zeros in the quantized DCT coefficients, ρ [115, 116, 117]. Based on these models, the authors estimate the rate-quantization and distortion-quantization functions before encoding, using these functions for rate control [118, 119, 120, 121].

In this context, the number of bits, R , and the pixel distortion, D , for a given VOP is modeled as follows

$$R(\rho) = \theta \cdot (1 - \rho) \quad (3.34)$$

$$D(\rho) = \sigma^2 e^{-\alpha(1-\rho)} \quad (3.35)$$

where θ is and α are model parameters, σ^2 is the picture variance, and ρ is the percentage of zero DCT coefficients on the current VOP as a function of the VOP quantization parameter. Below the algorithms for SVO and MVO rate control, based on these models, proposed by He and Mitra are briefly reviewed.

SVO BIT ALLOCATION

For constant bit rate single VO encoding, assuming a target channel bit rate, R , and a VO temporal rate, VR , the target number of bits, $R_T[n]$, for each encoding time instant, t_n , is computed as follows

$$R_T[n] = R_p - B[n] - \beta B_s \quad (3.36)$$

where $R_p = R/VR$, $B[n]$ is the encoder rate buffer occupancy for the current encoding time instant, B_s is the encoder rate buffer size, and β is a parameter defining the target encoder rate buffer occupancy before encoding each VOP.

SVO MB QUANTIZATION PARAMETER COMPUTATION

Given the target number of bits for encoding each VOP (3.36), the UCSB rate control algorithm computes the quantization parameter for each MB following the subsequent steps:

Step I – Initialization

Generate the histograms of the DCT coefficients for all MBs in the given VOP, $h_0(x)$ and $h_1(x)$, respectively, for the Intra and Inter coded MBs. After, compute the $\rho(q)$ function as follows

$$\rho(q) = \frac{1}{L} \left[\sum_{|x| < 2q} h_0(x) + \sum_{|x| < 2.5q} h_1(x) \right] \quad (3.37)$$

where L is the number of DCT coefficients in the current VOP, and q is the quantizer step²⁴.

Step II – MB Quantization Parameter Computation

For each MB m , compute $\rho[m]$ as follows

²³ S. K. Mitra is with the Department of Electrical and Computer Engineering, UCSB, Santa Barbara, CA, USA.

²⁴ In the MPEG-4 Video case [29], the second quantization method is assumed (see Section 2.4).

$$\rho[m] = 1 - \frac{1}{\theta[m-1]} \cdot \frac{R_T - \sum_{k=1}^{m-1} b_{MB}[k]}{\#MB_{\text{coeff}} \cdot (N_{MB} - m + 1)}, \quad m = 1, \dots, N_{MB} \quad (3.38)$$

where $b_{MB}[k]$ is the number of bits generated by MB k , $\#MB_{\text{coeff}}$ is the number of DCT coefficients in a MB ($\#MB_{\text{coeff}} = 384$ for MBs of 16×16 pixels and 4:2:0 chroma subsampling), N_{MB} is the number of MBs in the VOP, and $\theta[0] = 7$ (the typical average value of θ).

Using (3.37), compute the quantization step to encode the given MB.

Step III – Model Update

After encoding each MB, compute the number of zero coefficients in the current MB, $\rho_{MB}[m]$, and the MB bit count, $b_{MB}[m]$. If $m \geq 10$, estimate parameter θ as follows

$$\theta[m] = \frac{\sum_{k=1}^m b_{MB}[k]}{\#MB_{\text{coeff}} \cdot m - \sum_{i=1}^m \rho_{MB}[m]}, \quad m = 1, \dots, N_{MB} \quad (3.39)$$

MVO BIT ALLOCATION

He and Mitra proposed also to use the ρ -domain source modeling framework for MVO bit allocation [118, 121] adopting a different set of models (3.34) and (3.35) for each VO in the scene. In this case, the optimum bit allocation problem can be formulated as follows

$$\min_{\rho[i]} \left[\sum_{i=1}^{N_{VO}} p[i] \cdot \sigma^2[i] \cdot e^{-\alpha[i](1-\rho[i])} \right] \quad (3.40)$$

under the constraint

$$\sum_{i=1}^{N_{VO}} p[i] \cdot \theta[i] \cdot (1 - \rho[i]) = R_T[n] \quad (3.41)$$

where $p[i]$ is the size of VO i , and $R_T[n]$ is the total number of bits allocated for encoding time instant t_n .

The constrained problem specified by (3.40) and (3.41) can be converted into the following unconstrained problem

$$\min_{\rho[i]} \left[\sum_{i=1}^{N_{VO}} p[i] \cdot \sigma^2[i] \cdot e^{-\alpha[i](1-\rho[i])} + \lambda \left[\sum_{i=1}^{N_{VO}} p[i] \cdot \theta[i] \cdot (1 - \rho[i]) - R_T[n] \right] \right] \quad (3.42)$$

Solving (3.42) leads to the following optimal bit allocation for each VO

$$R[i] = \xi[i] \cdot p[i] \cdot \ln \left(\frac{\sigma^2[i]}{\xi[i]} \right) + \frac{\xi[i] \cdot p[i]}{\sum_{k=1}^{N_{VO}} \xi[k] \cdot p[k]} \left[R_T[n] - \sum_{k=1}^{N_{VO}} \xi[k] \cdot p[k] \cdot \ln \left(\frac{\sigma^2[k]}{\xi[k]} \right) \right] \quad (3.43)$$

where $\xi[i] = \theta[i]/\alpha[i]$.

Therefore, from (3.43), the target number of bits for each VOP is derived before encoding the given VOP.

MVO MB QUANTIZATION PARAMETER COMPUTATION

Using the VOP target number of bits for the current encoding time instant computed through (3.43), the UCSB rate control algorithm computes the MB quantization parameters for each VOP following the steps I to III of the SVO MB quantization parameter computation, where R_T is now the VOP target, $R[i]$, computed through (3.43).

MVO RATE-DISTORTION PARAMETERS ESTIMATION

After encoding each VOP, the rate and distortion model parameters are updated from the encoding results as follows

$$\theta[i] = \frac{S_{VOP}[i]}{1 - \rho_{VOP}[i]} \quad (3.44)$$

$$\alpha[i] = \frac{1}{1 - \rho_{VOP}[i]} \ln \frac{\sigma^2[i]}{D_{VOP}[i]} \quad (3.45)$$

where $S_{VOP}[i]$ is the number of texture bits, $\rho_{VOP}[i]$ is the percentage of zero DCT coefficients, $\sigma^2[i]$ is the VOP variance, and $D_{VOP}[i]$ is the VOP distortion. In [121], the authors also proposed a method for estimating $\alpha[i]$ before encoding with a given quantization step, q , assuming that the DCT coefficients have a Laplacian distribution, and estimating $\rho_{VOP}[i]$ and $D_{VOP}[i]$ from the input data and the distortion quantization function $D(q)$ for a Laplacian source.

Notice that the UCSB MVO rate control algorithm assumes that all VOs are synchronous, i.e., encoded with the same VOP temporal rates and at the same encoding time instants, and the rate control goal is the minimization of the VOs distortions without any constraint on the distortion between the different VOs in the scene; this may lead to large differences in terms of quality among the different VOs composing the scene. Moreover, smooth temporal quality is also not taken into account in this algorithm.

3.6.7 University of Texas at Arlington²⁵ SVO and MVO Rate Control

Sun²⁶ and Ahmad²⁷ addressed the problem of CBR rate control for single and multiple synchronous video objects proposing a rate control algorithm based on a feedback buffer controller using a proportional-integral-derivative (PID) technique [122, 123]. Recently, the authors also addressed the problem of rate control for multiple asynchronous video objects proposing some modifications to the original algorithm [124, 125].

²⁵ University of Texas at Arlington (UTA), Arlington, TX, USA.

²⁶ Y. Sun was formerly with the Department of Computer Science and Engineering, UTA, Arlington, TX, USA, and is now with the Department of Computer Science, University of Central Arkansas, Conway, AR, USA.

²⁷ I. Ahmad is with the Department of Computer Science and Engineering, UTA, Arlington, TX, USA.

SYNCHRONOUS MVO RATE CONTROL

The UTA rate control algorithm [122, 123] can be described by the following seven steps.

Step I – Initialization

Initialize the rate control and encoder parameters.

Step II – Initial Frame²⁸ Bit Allocation for Each Encoding Time Instant

For each encoding time instant, t_n , the initial frame target number of bits is computed as follows

$$\bar{T}_F[n] = \alpha_{T[n]} \cdot \frac{R_R[n]}{\alpha_I[n] \cdot N_I[n] + \alpha_P[n] \cdot N_P[n] + \alpha_B[n] \cdot N_B[n]} \quad (3.46)$$

where $R_R[n]$ is the number of bits left for the remaining encoding time instants; $\alpha_I[n]$, $\alpha_P[n]$, $\alpha_B[n]$, $N_I[n]$, $N_P[n]$, $N_B[n]$ are, respectively, the I-, P-, and B-VOP coding type weights and number of frames of each type left to be encoded; and $\alpha_{T[n]}$ is the coding type of the current frame (the same for all VOPs).

Step III – Frame Bit Allocation Adjustment

The initial frame target number of bits computed through (3.46) is first adjusted based on the frame complexity according to the following equation

$$T_F^0[n] = \bar{T}_F[n] \frac{C_F[n]}{\bar{C}_F[n]} \quad (3.47)$$

where $C_F[n]$ is the current frame complexity and $\bar{C}_F[n]$ is the average frame complexity of the previous n_P or n_B frames, depending on the current frame coding type:

$$C_F[n] = \sum_{i=1}^{N_{VO}} (NW_{VO}[n][i] \cdot C_{VO}[n][i]) \quad (3.48)$$

where $NW_{VO}[n][i]$ and $C_{VO}[n][i]$ are, respectively, the normalized VO i weight and coding complexity for encoding time instant t_n .

$$NW_{VO}[n][i] = \frac{W_{VO}[n][i]}{\sum_{k=1}^{N_{VO}} W_{VO}[n][k]} \quad (3.49)$$

where $W_{VO}[n][i]$ is the VO i weight for encoding time instant t_n , dynamically adjusted during the encoding process (initially $W_{VO}[i] = 1.0$, $i = 1, \dots, N_{VO}$).

$$C_{VO}[n][i] = SIZE_{VO}[n][i] \cdot (VAR_{VO}[n][i])^k \quad (3.50)$$

where $SIZE_{VO}[n][i]$ and $VAR_{VO}[n][i]$ are, respectively, the number of non-transparent MBs

²⁸ In the context of UTA rate control algorithm, a frame is defined as the set of VOPs of the various VOs for each encoding time instant.

and the luminance variance of VO i in encoding time instant, t_n , and $k = 1/4$.

The complexity-adjusted target (3.47) is further adjusted based on the occupancy of the encoder rate buffer through a PID controller as follows

$$T_F^1[n] = (1 + PID[n]) \cdot T_F^0[n] \quad (3.51)$$

where

$$PID[n] = K_p \left(E(t_n) + K_I \cdot \int_0^{t_n} E(\tau) d\tau + K_D \cdot \frac{dE(t_n)}{dt} \right) \quad (3.52)$$

$$E[n] = \frac{B_s/2 - B[n]}{B_s/2} \quad (3.53)$$

where B_s and $B[n]$ are, respectively, the encoder rate buffer size and occupancy, and $K_p = 1.0$, $K_I = 0.25$, and $K_D = 0.3$ are, respectively, the proportional, integral and derivative constants.

Notice that the control error defined by (3.53) sets the target encoder rate buffer occupancy to $B_s/2$, which means that the PID controller forces the encoder to reach an approximate constant encoder rate buffer occupancy independently of the frame coding type and treats equally buffer overflows and underflows.

Step IV – VO Bit Allocation

The target bits (3.51) are distributed among the different VOs as follows

$$T_{VOP}[n][i] = \frac{NW_{VO}[n][i] \cdot NSIZE_{VO}[n][i] \cdot NVAR_{VO}[n][i]}{\sum_{k=1}^{N_{VO}} (NW_{VO}[n][k] \cdot NSIZE_{VO}[n][k] \cdot NVAR_{VO}[n][k])} \cdot T_F^1[n] \quad (3.54)$$

where $NW[n][i]$, $NSIZE[n][i]$, and $NVAR[n][i]$ are, respectively, the normalized VO i VOP weight, size and variance.

Step V – VOP Quantization Parameter Computation

To compute the VOP quantization parameter for each VO, the UTA rate control algorithm uses the MPEG-4 Video VM8 [108] approach, estimating the number of bits to encode the VOP texture information by subtracting from the target number of bits for the current encoding time instant the number of bits used to encode the shape, motion, and header data in the previous encoding time instant, i.e.,

$$T_{VOP}^{\text{text}}[n][i] = T_{VOP}[n][i] - H_{VOP}[n-1][i] \quad (3.55)$$

where $H_{VOP}[n-1][i]$ is the number of bits used to encode the motion, shape and header of VO i in the previous encoding time instant.

For P-VOPs, the VO i VOP quantization parameter, $Q[i]$, is then computed through the quadratic rate-quantization model, solving the following equation

$$T_{VOP}^{\text{text}}[i] = \left(X_1[i] \frac{1}{Q[i]} + X_2[i] \frac{1}{Q^2[i]} \right) \cdot MAD[i] \quad (3.56)$$

For I-VOPs, the UTA rate control algorithm computes the VO i VOP quantization parameter, $Q[i]$, based on the average quantization parameter of previous P-VOPs and an adjustment parameter, i.e.,

$$Q_l[n][i] = \bar{Q}[n][i] + \beta_l[i] \quad (3.57)$$

where $\bar{Q}[n][i]$ is the average quantization parameter of previous l P-VOPs of VO i , and $\beta_l[i]$ is an adjustment parameter computed as

$$\beta_l[i] = \beta_l^{\text{prev}}[i] + \frac{PSNR_l^{\text{prev}}[i] - \overline{PSNR}^{\text{prev}}[i]}{\lambda} \quad (3.58)$$

where $\beta_l^{\text{prev}}[i]$ is the adjustment parameter for the last I-VOP of VO i , $PSNR_l^{\text{prev}}[i]$ is the PSNR of the last I-VOP, $\overline{PSNR}^{\text{prev}}[i]$ is the average PSNR of the l P-VOPs before the last I-VOP of VO i , and $\lambda = 16$ is a correction parameter.

Step VI – Encoding

Encode each VO VOP with the quantization parameter computed through (3.56) or (3.57) as appropriate. Notice that this algorithm does not provide any MB quantization parameter adjustment.

Step VII – Parameter Adjustment

The UTA rate control algorithm needs to update three types of parameters: rate-quantization parameters, coding type weights, and VO weights.

The rate-quantization parameters X_1 and X_2 are updated through linear least squares estimation as described in the MPEG-4 Video VM8 [108] frame rate control (see Section 2.5).

The coding type weights α_l and α_B are updated as follows

$$\alpha_l[n] = \frac{\bar{b}_l[n]}{\bar{b}_p[n]} \cdot e^{(\overline{PSNR}_p[n] - \overline{PSNR}_l[n])/\gamma} \quad (3.59)$$

$$\alpha_B[n] = \frac{\bar{b}_B[n]}{\bar{b}_p[n]} \cdot e^{(\overline{PSNR}_p[n] - \overline{PSNR}_B[n])/\gamma} \quad (3.60)$$

where $\bar{b}_l[n]$, $\bar{b}_p[n]$, and $\bar{b}_B[n]$ are the average number of bits per frame computed over previous n_l I-frames, n_p P-frames, and n_B B-frames, and $\overline{PSNR}_l[n]$, $\overline{PSNR}_p[n]$, and $\overline{PSNR}_B[n]$ are the corresponding average PSNRs (in [123] $n_l + n_p + n_B = 30$).

Notice that in this Thesis [26] it was already proposed to use the concept of coding type weights to better balance the bit allocation between the different VOP coding types. However, it was found that these parameters should be different for each VO in the scene due to their possible different characteristics.

The VO weights W_{VO} are updated similarly to the coding weights as follows

$$W_{VO}[n][i] = W_{VO}[n-1][i] \cdot e^{(PSNR[n-1][1] - PSNR[n-1][i])/\theta}, \quad i = 2, \dots, N_{VO} \quad (3.61)$$

Notice that in (3.61) VO 1 is the reference for the other VOs.

Step VIII – Frame Skipping Control

As in MPEG-4 Video VM8 [108], for each target encoding time instant, if the encoder rate buffer occupancy is above 80% of the buffer size, the current frame is skipped and the buffer occupancy is updated for the next target encoding time instant.

In this context, at the end of each frame encoding, the UTA rate control algorithm updates the encoder rate buffer occupancy as follows

$$B[n] = B[n-1] + A[n] - \bar{T}_F[n] \quad (3.62)$$

where $A[n]$ is the total number of bits used for encoding the set of VOPs at encoding time instant t_n , and $\bar{T}_F[n]$ is the number of bits drained from the encoder rate buffer between consecutive encoding time instants given by (3.46).

Notice that $\bar{T}[n]$ depends on the frame coding type and the encoder bit production and not only on the elapsed time from the last encoding time instant and the channel bit rate as in real CBR encoding scenarios. Therefore, although (3.62) is adequate for guaranteeing an average target encoded bit rate, it cannot prevent encoder rate buffer overflows and underflows in CBR encoding scenarios, where the drain rate of the encoder rate buffer is constant.

ASYNCHRONOUS MVO RATE CONTROL

Subsequently to the work presented in this Thesis [26], Sun and Ahmad proposed some modifications to their initial rate control algorithm in order to encode multiple video objects at different temporal resolutions [124, 125].

I) Initial VO Bit Allocation for each Encoding time Instant

In this context, the initial VO i VOP bit allocation for encoding time instant t_n is computed as follows

$$\bar{T}_{VOP}[n][i] = \frac{\alpha_{T[n][i]}}{\alpha_I[n][i] \cdot N_I[n][i] + \alpha_P[n][i] \cdot N_P[n][i] + \alpha_B[n][i] \cdot N_B[n][i]} \times L[n][i] \times R_R[n] \quad (3.63)$$

where $L[n][i]$ is the temporal bit-ratio of VO i for time instant t_n computed as follows

$$L[n][i] = \frac{VR[i] \cdot \bar{A}[n][i]}{\sum_{k=1}^{N_{VO}} (VR[k] \cdot \bar{A}[n][k])} \quad (3.64)$$

where $VR[i]$ is the temporal rate of VO i and $\bar{A}[n][i]$ is the average number of bits per VOP of VO i over previous $N_C[i]$ encoding time instants, corresponding to a period of t_A seconds, i.e.,

$$\bar{A}[n][i] = \frac{1}{N_C[i]} \sum_{k=1}^{N_C[i]} A[n-k][i] \quad (3.65)$$

where $A[n-k]$ is the number of bits used by the k^{th} previous encoding time instants of VO i and $N_C[i]$ is the number of VOPs of VO i encoded during t_A , i.e.,

$$N_C[i] = t_A \cdot VR[i] \quad (3.66)$$

II) VO Bit Allocation Adjustment

The initial VO i VOP target number of bits computed through (3.63) is first adjusted based on the VO complexity according to the following equation

$$T_{VOP}^0[n][i] = \frac{NC_{VO}[n][i]}{\bar{C}_{VO}[n][i]} \cdot \bar{T}_{VOP}[n][i] \quad (3.67)$$

where

$$NC_{VO}[n][i] = NW_{VO}[n][i] \cdot C_{VO}[n][i] \quad (3.68)$$

with $NW_{VO}[n][i]$ computed as in (3.49) and where $C_{VO}[n][i]$ is the complexity of VO i at encoding time instant t_n computed, in this version of the UTA rate control algorithm, as the sum of non-transparent MB luminance variances.

Averaging (3.68) over the previous $N_C[i]$ encoding time instants gives the average coding complexity of VO i , for encoding time instant t_n , $\bar{C}_{VO}[n][i]$, i.e.,

$$\bar{C}_{VO}[n][i] = \frac{1}{N_C[i]} \sum_{k=1}^{N_C[i]} NC_{VO}[n-k][i] \quad (3.69)$$

As in the original UTA rate control algorithm [122, 123], the VO i target number of bits (3.67) is further adjusted through a PID controller as in (3.51).

III) VOP Quantization Parameter Computation

The same method presented above for UTA synchronous MVO rate control is used.

IV) Parameter Adjustment

The rate-quantization parameters are updated as in the original algorithm. The VO coding weights α_I and α_B are now updated as follows

$$\alpha_I[n] = \bar{b}_I[n] / \bar{b}_P[n] \quad (3.70)$$

$$\alpha_B[n] = \bar{b}_B[n] / \bar{b}_P[n] \quad (3.71)$$

where $\bar{b}_I[n]$, $\bar{b}_P[n]$, and $\bar{b}_B[n]$ are the average number of bits per frame computed over previous n_I I-frames, n_P P-frames, and n_B B-frames.

The VO weights W_{VO} are now updated as follows

$$W[n][i] = W[n-1][i] \times \left(\frac{\overline{PSNR}[n-1]}{PSNR[n-1][i]} \right)^2 \quad (3.72)$$

where

$$\overline{PSNR}[n-1] = \frac{\sum_{i=1}^{N_{VO}} (SIZE_{VO}[n-1][i] \cdot PSNR[n-1][i])}{\sum_{i=1}^{N_{VO}} SIZE_{VO}[n-1][i]} \quad (3.73)$$

V) Frame Skipping Control

Frame skipping control is performed as in the original UTA rate control algorithm; however, in this context, at the end of each VOP encoding, the encoder rate buffer occupancy is updated by $A[n][k] - \bar{T}_{VOP}[n][k]$, which means that at the end of each encoding time instant the encoder rate buffer is updated as follows

$$B[n] = B[n-1] + \sum_{k=1}^{N_{VO}} (A[n][k] - \bar{T}_{VOP}[n][k]) \quad (3.74)$$

Updating the encoder rate buffer occupancy according to (3.74) is a workaround for dealing with non-uniform bit allocation when the various VOs in the scene are encoded at different temporal resolutions since (3.74) assumes that the number of bits drained from the encoder rate buffer to the channel fluctuates according to the encoding time instant. In practice, what should have been considered, as proposed in the context of Thesis (see Section 6.4), is to adopt different target buffer occupancies for each encoding time instant according to the number and complexity of the VOs being encoded; otherwise, in a constant bit rate encoding scenario, there are no guarantees that encoder rate buffer overflows and underflows are avoided.

3.6.8 Other Related Work

Several other research groups addressed the problem of rate control for MPEG-4 video coding proposing slight changes to the original MPEG-4 rate control algorithms described in Section 2.5, or addressing particular issues related to MPEG-4 video coding rate control. Since these proposals do not include complete rate control algorithms, but only parts of existing ones, they are not included as separate sections, but are referred here by the specific problem they addressed. This section reviews some of these proposals.

VARIABLE BIT RATE MPEG-4 VIDEO ENCODING

Jagmohan [126, 127] proposed a method for single-pass SVO MPEG-4 VBR video encoding where the first step in encoding a given frame is to select its spatial resolution²⁹ based on the spatio-temporal complexity of the frame estimated based on the amount of deviation from the target bit rate, the prediction error, and the average quantization parameter of the last encoded frame.

In this context, the frame bit allocation is performed over sliding windows taking into account the long-term target average bit rate and the amount of excess bits used up to the given encoding time instant. This frame bit allocation is further adjusted based on the proportion of bits used up to the given encoding time instant, the motion complexity of the current frame relatively to the average motion complexity of previous frames, and the average quantization parameter of the last encoded frame relatively to the average quantization parameter of previous encoded frames.

JOINTLY ENCODING OF MULTIPLE VIDEO SEQUENCES

Hung [128] and Yang [129] addressed the problem of jointly encoding multiple video sequences, respectively for H.263 [8] and H.264/Advanced Video Coding (AVC) [61, 37]. Although in this case there is no single video buffering verifier mechanism for controlling the

²⁹ This method is applied to the ARTS Profile that supports the reduced resolution coding tool (see Section 2.4).

encoding of the multiple video programs, this approach has some similarities with MVO rate control, though, in this case, all video sequences have fixed spatial resolutions, which reduces this problem to a typical statistical multiplexing problem [130, 131, 132, 133].

In [128], the authors introduced the concept of super frames corresponding to the set of frames of all video sequences in a given time period. These super frames are jointly encoded computing the rate-distortion characteristics of each super frame and perform optimal bit allocation for each frame of the super frame assuming a constant bit rate channel and a constant number of bits per super frame.

In [129], Yang *et al.* used the concept the concept of super frames, but now applied for each encoding time-instant. In this case, a super frame is the set of frames of all video sequences to be encoded in that particular encoding time instant. The bit allocation between super frames is similar to the MPEG-2 Video TM5 [134] bit allocation method, while inside each super frame the super frame target number of bits is distributed among the different frames according to the different frames weights based on the MAD of each frame estimated from the previous encoding time instants.

IMPROVEMENT OF THE MPEG-4 VM5 RATE CONTROL MECHANISM

Pan *et al.* [135, 136] proposed some changes to the MPEG-4 Video VM5 rate control algorithm (see Section 2.5), which, according to the authors, allow reducing the occurrence of frame skipings and increasing the average PSNR. These changes can be summarized as follows:

- **Lower bound on the frame bit allocation** – A new frame bit allocation lower bound of $R_p/3$ bits per frame, where R_p is the nominal target number of bits per frame, while originally this lower bound was set to $R_s/30$, where R_s is the target average channel bit rate.
- **Target encoder rate buffer occupancy** – Set the target buffer occupancy, B_T , according to the frame coding type and favor the prevention of overflows instead of underflows by defining a target buffer occupancy at the end of each GOP $B_T = B_S/6$ instead of $B_T = B_S/2$ as in the original MPEG-4 Video VM5 rate control algorithm.
- **Quantization parameter range** – Restrict the minimum quantization parameter value to $Q_{\min} = 5$, if $B > B_T$, and $Q_{\min} = 3$, if $B \leq B_T$, in order to avoid bit allocation oscillations, notably for low bit rate encoding (while originally $1 \leq Q \leq 31$).
- **Sliding-window size adaptation** – Force the sliding window, w (see Section 2.5), to increase smoothly after scene changes, i.e., $w[i] \leq w[i-1] + 1$, in order to favor the use of recent data for rate-distortion parameter estimation after scene changes.
- **Quantization parameter after frame skipping** – Increase by 25% the quantization parameter of P-frames after a frame skipping in order to avoid bit allocation oscillations after frame skipping.
- **Adjust bit allocation and target buffer occupancy according to the frame position in the GOP** – According to the authors, P-frames closer to the I-frame should get a higher bit allocation since they will be used as predictions for other future frames. Therefore, the bit allocation of P-frames is changed, relatively to the MPEG-4 Video VM5 rate control algorithm, by the amount ΔT_p that depends on the frame relative

position inside the GOP as follows

$$\Delta T_p(n) = \frac{\bar{T}_p}{5} \frac{N-2n}{N-2}, \quad n=1, \dots, N-1 \quad (3.75)$$

where \bar{T}_p is the average number of bits for P-frames, and N is the GOP size.

The target buffer occupancy for each encoding time instant should also be adjusted to reflect the different nominal bit allocations for each P-VOP.

- **Adaptive quantization parameter for I-frames** – Set the quantization parameter for I-frames according to the following expression

$$Q_I = \frac{16.34}{b_I^{2.05}} \times MAV_{DCT}^{1+0.29 \times \ln(b_I)} \quad (3.76)$$

where b_I is the target number of bits for the I-frame in kbit and MAV_{DCT} is the mean absolute value of the DCT coefficients in the frame. In addition, $5 \leq Q_I \leq 25$.

With these changes, the authors report PSNR gains between -0.1 dB and $+0.6$ dB for typical MPEG-4 test material.

Chen and Ngan [137, 138, 139], also proposed some changes to the MPEG-4 Video VM5 rate control algorithm, notably to overcome some problems in the quantization parameter computation through the quadratic rate-quantization model (see Section 5.2). In this context, they proposed to specify constraints for the validity of the quadratic rate-quantization model, as also already proposed by Nunes and Pereira in [26]. Additionally, Chen and Ngan also proposed to restrict the quantization parameter variation between consecutive encoding time instants based on the buffer occupancy variation and VOP complexity variation.

In [140, 141], the same authors also addressed the problem of shape bit rate estimation through a linear model where the shape bits are estimated through a linear function of the number of boundary MBs in the corresponding VOP.

3.7 Final Remarks

This chapter investigates the new dimensions and strategies of rate control when an object-based coding architecture is used. Special emphasis was put on the study of the semantic dimension, notably the new rate control strategies associated to the semantic resolution and to the amount of content. In this context, a new framework for object-based video coding rate control is proposed, where this important function of any video encoder is performed by using two levels: the scene-level rate control and the object-level rate control. Additionally, a review of the existing object-based rate control techniques is presented.

The following chapters will be devoted to tackle some problems opened by this new object-based video representation. Therefore, Chapter 4 considers mechanisms for guaranteeing interoperability among devices/applications targeting compliance with a given MPEG-4 Visual profile@level; Chapter 5 addresses the problem of rate and distortion modeling for Intra and Inter coding in the context of object-based MPEG-4 video encoding; and finally, Chapter 6 proposes an efficient rate control algorithm for single and multiple video object encoding.

Chapter 4

MPEG-4 Video Buffering Verifier

Mechanism: Analysis and Alternatives

4.1 Introduction

One of the main reasons for the existence of the MPEG-4 profiling mechanism is the need to bound the complexity of decoders and bitstreams while guaranteeing interoperability. As pointed out previously, it is not viable that all MPEG-4 enabled devices/applications support the whole MPEG-4 toolbox since many tools are rather complex and moreover useless in certain application domains. Thus profiles have been defined, basically targeting maximum interoperability at minimum complexity. However, even for a given profile, it is not reasonable that all decoders have to be designed to support the full range of processing complexities that a certain profile may cover. Thus levels have to be defined for each profile in order to bound the decoding resources required by a given visual-clip bitstream collection¹.

Profile and level information indicate which restrictions have been applied to the several syntactic and semantic² bitstream elements. The restrictions defined for a given profile@level

¹ The term “bitstream” means in this Thesis, an ISO/IEC 14496 (MPEG-4) video bitstream. A bitstream is the coded representation of one layer of a single visual object. A “visual-object bitstream collection” is the set of bitstreams representing all the layers of one VO. A “visual-clip bitstream collection” is a set of bitstreams representing all the layers of all the visual objects making a video clip [31].

² By syntax it is meant the set of rules that the coded representation of a video object using the MPEG-4 Visual standard has to follow, notably the sequence of the symbols, while semantics is related to the meaning of the symbols in the coded representation and thus to the operations that have to be performed to decode them.

combination aim at reducing the cost of encoder and decoder implementations while providing interoperability. Interoperability can only be achieved if bitstreams and decoders comply with what has been specified in the standard, i.e., a compliant set of video bitstreams shall be decodable by any compliant MPEG-4 video decoder that supports the relevant profile and level combination. Thus, compliance plays a major role in the successful deployment of the MPEG-4 standard and may be divided into [31]:

- **Bitstream compliance** – A set of bitstreams (visual-clip bitstream collection) compliant with a given video profile@level shall not contain any disallowed syntactic element for that profile, neither the parameter values in any video bitstream shall exceed the allowed values for that profile@level. Additionally the set of bitstreams shall not violate the complexity restrictions defined for the profile@level in question.
- **Decoder compliance** – A decoder compliant with a particular profile@level shall be able to interpret all allowed values of all allowed syntactic elements for that profile@level (*static compliance*) and to perform all decoding operations according to the decoding semantics for the syntax supported by that profile@level at the required pace (*dynamic compliance*).

Although MPEG-4 does not directly address encoder compliance, since this is not essential for interoperability, any encoder claiming compliance with a given MPEG-4 visual profile@level shall produce sets of bitstreams compliant to the corresponding profile@level in the sense defined above [31].

The problem of how to control a video encoder in order to produce bitstreams compliant with a given MPEG-4 visual profile@level is the main motivation of this chapter, which proposes and analyses a method for the implementation of some natural visual profiles and levels. In the context of this Thesis, the focus is put on the visual profiles suited to encode video data – here called video profiles.

Each video profile@level combination defines a sub-set of the syntax and semantics of the MPEG-4 Visual standard [29] and establishes lower limits on the decoder resources required to decode any set of elementary bitstreams building a video scene and compliant to the relevant profile@level. These resources are the bitstream memory – the memory where the bits wait to be decoded, the computational power, and the picture memory – the memory where the decoded pixels are stored until they are no longer needed. Annex D of the MPEG-4 Visual standard [29] specifies a mechanism based on virtual buffers – Video Buffering Verifier – to constrain the set of bitstreams produced by an MPEG-4 video encoder, in the context of a scene encoded complying to a given MPEG-4 visual profile@level, in order that those bitstreams can be considered compliant with the selected profile@level.

Any set of elementary bitstreams building a video scene can only be considered profile@level compliant if it does not violate the video buffering verifier constraints for the chosen profile@level. This requirement makes the enforcement of this mechanism a very important tool in any MPEG-4 video encoder, since it has the task to guarantee that the profile@level constraints are never violated.

This chapter provides a detailed analysis of the MPEG-4 video buffering verifier mechanism, discussing its major features and drawbacks, notably in comparison with alternative solutions. Furthermore, this chapter proposes a model for the integration of the MPEG-4 video buffering verifier mechanism into a generic video encoder rate control mechanism in order to produce bitstreams that comply to a chosen MPEG-4 video profile@level.

The chapter is organized as follows: after this introduction, Section 4.2 offers a description of the MPEG-4 video buffering verifier mechanism; afterwards, Section 4.3 briefly compares this mechanism with the equivalent mechanisms in the most widely known video coding standards, MPEG-2 Video [10] and ITU-T H.263 [8]; Section 4.4 proposes a model for the integration of the MPEG-4 video buffering verifier mechanism into an MPEG-4 encoder; Sections 4.5 to 4.7 study the implementation of each of the video buffering verifier models in a video encoder, discussing alternative solutions and presenting some implementation results obtained by encoding some well known MPEG-4 test sequences with the MPEG-4 Visual Simple and Core Profiles; finally Section 4.8 summarizes the main conclusions of the chapter.

4.2 The MPEG-4 Video Buffering Verifier Mechanism

The idea of using a video buffering verifier mechanism to bound the decoding complexity of a given set of bitstreams is not new, and was already adopted in previous MPEG video coding standards, MPEG-1 [9] and MPEG-2 [10]. In these standards, the major purpose of the video buffering verifier mechanism was to set some restrictions on the maximum variability of the number of bits per picture, especially in the case of constant bit rate operation, and thus on the complexity of the encoded video streams.

Generically, the complexity of the encoded video is directly related to the encoded bit rate and to the decoded video data rate that the decoder generates, e.g., measured in terms of the number of MB/s. For frame-based video coding, e.g., MPEG-1 and MPEG-2, the decoded video data rate is typically constant since the frames have fixed dimensions and are usually encoded at fixed frame rates. This is not the general case for object-based video coding, e.g., MPEG-4, since the several video objects composing a scene may vary in size along time and may be encoded at different VOP rates. Therefore, the amount and type³ of MB/s that a given object-based video decoder has to process may largely vary over time in comparison with frame-based coding solutions.

In the MPEG-4 context, to limit the decoding complexity of a set of bitstreams corresponding to a video scene it is then necessary to set some limits on the variability of the number of decoded MB/s, and their complexity, and also on the picture memory required to store the decode data. This constitutes the major novelty of the MPEG-4 video buffering verifier mechanism, relatively to the previous MPEG standards, since it does not only bound the bitstream buffer memory but also the MB decoding capacity and the MB picture memory.

The MPEG-4 video buffering verifier mechanism consists of three normative models, each one defining a set of rules and limits to verify if the amount required for a specific type of decoding resource is within the values allowed by the corresponding profile and level specification:

1. **Video Rate Buffer Verifier (VBV)**⁴ – This model is used to verify that the bitstream memory required at the decoder(s) does not exceed the values specified for the corresponding profile and level. The model is defined in terms of the VBV buffer sizes for all the VOLs corresponding to the objects building the scene. Each VBV buffer size

³ For an arbitrarily shaped video object, three types of MBs may exist: transparent, opaque, and boundary.

⁴ In MPEG-4 Visual Annex D, the VBV is also referred as video buffering verifier; however since the entire video verifier mechanism is also referred by the same name, it was decided to use here the term video rate buffer verifier for the rate buffer model and the term video buffering verifier for the complete mechanism in order to avoid ambiguity.

corresponds to the maximum amount of bits that the decoder can store in the bitstream memory for the corresponding VOL; there is, however, also a limitation on the sum of the VOL VBV buffer sizes. The bitstream memory is the memory where the decoder puts the bits received for a VOL while waiting to be decoded.

2. **Video Complexity Verifier (VCV)** – This model is used to verify that the computational power (processing speed), defined in terms of MB/s, required at the decoder does not exceed the values specified for the corresponding profile and level. The model is defined in terms of the VCV MB/s decoding rate and VCV buffer size and is applied to all MBs in the scene. If arbitrarily shaped VOs exist in the scene, an additional VCV buffer and VCV decoding rate is also defined, to be applied only to the boundary MBs.
3. **Video Reference Memory Verifier (VMV)** – This model is used to verify that the picture memory required at the decoder for the decoding of a given scene does not exceed the values specified for the corresponding profile and level. The model is defined in terms of the VMV buffer size, which is the maximum number of decoded MBs that the decoder can store during the decoding process of all VOLs corresponding to the scene.

In order that the set of visual elementary streams corresponding to a given scene may be considered compliant with a given profile and level, the encoder must guarantee that none of the above-mentioned buffers overflows and, additionally, it must also guarantee that, in certain circumstances, the VBV buffer never underflows.

4.2.1 Video Rate Buffer Verifier (VBV) Definition

The MPEG-4 VBV model defines a set of rules and limits for examining a video elementary bitstream with a delivery rate function, $R(t)$. This model simulates the occupancy of the decoder bitstream buffer in order to control the amount of bitstream memory required at the decoder. Its purpose is to guarantee that the bitstream memory required is less than the specified buffer size, i.e., to verify that the decoder bitstream buffer occupancy never goes beyond the limits of the specified buffer size for the relevant profile@level. In the case of visual scenes composed by multiple VOs, each with one or more VOLs, the MPEG-4 Visual standard specifies that the video rate buffer model shall be applied independently to each VOL (using a particular buffer size and rate function for each VOL). Additionally, the maximum total bitstream buffer size (defined as the sum of all VOL bitstream buffer sizes) for the given profile and level shall not be exceeded [29]. Notice that the bit rate and buffer size allocation, among the several VOs and, for each VO, among the several VOLs, is a non-normative issue that will be addressed in the following chapters of this Thesis. This is a very important issue because it can significantly determine the performance of object-based video encoders, and thus deserves careful attention.

The VBV applies to video data encoded as a combination of I-, P-, B-, and S-VOPs, using several coding tools organized in terms of video object types (see Chapter 2). Face animation, still texture, and mesh objects are not constrained by the VBV model. The coded video bitstreams shall be constrained to comply with the requirements of the VBV specified in the following sections.

VBV MODEL PARAMETERS

The VBV model for a given elementary stream (ES) is defined by the three following

parameters: *vbv_buffer_size*, *vbv_occupancy*, and *bit_rate*. These parameters have to be defined for all the ESs corresponding to the various objects in a scene.

These parameters can be specified at video level, this means through the video ES, or by means of systems level configuration information [28]. In the first case, the VBV model parameters are specified in the VOL header, when the one-bit flag *vbv_parameters* is set to “1”. In the second case, the VBV model parameters are conveyed to the video decoder through the Object Description Information, more precisely through the *DecoderConfigDescriptor* field of the *ES_Descriptor* associated to the ES in question.

When the *vbv_buffer_size* and *vbv_occupancy* parameters are specified by systems level configuration information, the bitstream shall be constrained according to the specified values, and these values shall not be part of the video ES. It may happen, however, that these parameters are not explicitly specified; in this case, it is assumed that the ES is constrained according to the default values of the corresponding profile and level combination⁵.

VBV Buffer Size

The VBV buffer size for a VOL specifies the minimum bitstream memory required at the decoder to properly decode the corresponding VOL ES. The VBV buffer size for a VOL is defined by the 18-bit *vbv_buffer_size* field in units of 16384 bits (the value zero is forbidden). The maximum VBV buffer size in bits, vbv_{BS} , is then given by

$$vbv_{BS} = 16384 \times vbv_buffer_size \text{ [bit]}$$

The *vbv_buffer_size* value is bounded by *Max VOL VBV buffer size* in Table 2.6, which specifies the levels' constraints, and the sum of all these values for all VOLs is bounded by *Max total VBV buffer size*.

The default value of *vbv_buffer_size* for a VOL is the maximum value of *vbv_buffer_size* allowed for the profile and level combination in question (called *Max VOL VBV buffer size* Table 2.6). Still, it must be checked that the sum of the *vbv_buffer_size* default values does not exceed *Max total VBV buffer size*.

In terms of the levels specification shown in Table 2.6, there are two constraints defined: *Max VOL VBV buffer size*, which sets the limit for each VOL, and *Max total VBV buffer size*, which sets the limit on the sum of all the VOL buffer sizes.

VBV Occupancy

The VBV occupancy for a VOL specifies the initial occupancy of the VBV buffer for that VOL, this means the occupancy that the VBV buffer must reach in order that the decoding process may start with the removal of the first VOP bits following the VOL header; this parameter, together with the *bit_rate* parameter, establishes the initial decoding delay, the so-called VBV latency. The VBV occupancy is defined by the 26-bit *vbv_occupancy* field in units of 64 bits⁶.

⁵ Except for the short video header case as described in Section 4.2.1.

⁶ For basic sprites, the *vbv_occupancy* field specifies the initial VBV occupancy before decoding the first S-VOP in the elementary stream, i.e., not the very first VOP in a basic sprite, which must be an I-VOP, but the subsequent VOP, i.e., an S-VOP. Low-latency sprites, that allow the transmission of large image sprites progressively (both spatially and in terms of quality), are treated as any other VOL.

The default value of $vbv_occupancy$ for a VOL, in 64-bit units, is given by $170 \times vbv_buffer_size$ (for that VOL), where vbv_buffer_size is in 16384-bit units; of course, the maximum value of $vbv_occupancy$ is vbv_buffer_size for the corresponding VOL. This corresponds to an initial occupancy (before the removal of the first VOP from the buffer) in bits, vbv_0 , of approximately two-thirds of the defined buffer size, i.e.,

$$\begin{aligned} vbv_0 &= 64 \times vbv_occupancy \\ &= 64 \times \left(170 \times \frac{vbv_{BS}}{16384} \right) \approx \frac{2}{3} \times vbv_{BS} \quad [\text{bit}] \end{aligned}$$

Note that there is no explicit limitation on the $vbv_occupancy$ in terms of levels definition.

Bit Rate

When present for a VOL, the bit rate parameter, defined by the 30-bit bit_rate field in units of 400 bits per second (value zero is forbidden), specifies the ES peak bit rate for VOL_{ij} ⁷, such that

$$R_{VOL_{ij}}(t) \leq 400 \times bit_rate,$$

where $R_{VOL_{ij}}(t)$ is defined as the instantaneous VOL channel bit rate for VOL_{ij} (in bits per second) counting only the visual syntax.

If the channel, with a total instantaneous channel rate, $R(t)$, is a serial time multiplex of several streams (e.g., as defined by MPEG-4 Systems [28]) then $R_{VOL_{ij}}(t) = R(t)$ for the time instants where the channel is occupied by the relevant VOL_{ij} bits; otherwise it is zero [29], i.e.,

$$R_{VOL_{ij}}(t) = \begin{cases} R(t) & \Leftarrow t \in \{\text{bit time interval from } VOL_{ij}\} \\ 0 & \text{otherwise} \end{cases}$$

Notice that the purpose of the bit rate parameter is to provide an upper bound on the VOL ES bit rate rather than a precise value of the actual VOL bit rate since MPEG-4 Visual does not specify any temporal window to measure the actual ES bit rate.

In terms of the levels specification shown in Table 2.6, only the sum of the bit rate for all the VOLs for all the objects in the scene is bounded, assuming that this total bit rate can be shared among the VOLs at author's wishes (signaled using the bit_rate field for each VOL).

VBV OCCUPANCY DYNAMICS

The VBV occupancy dynamics specifies when the bitstream bits enter the VBV buffer and when they are removed from it to be decoded, i.e., the process by which the VBV buffer is filled and drained. This process is mainly driven by the time instants at which the VOP bits are removed from the VBV.

⁷ VOL_{ij} corresponds to VOL j of VO i .

VBV Buffer Filling

The VBV buffer for each ES is initially empty and filled as coded data arrives, until it reaches the value specified in the *vbv_occupancy* field, or the first VOP decoding time arrives. The first bit that is put in the VBV buffer is the first bit of the elementary stream (the VOL header bits are not taken into account since they are not considered to be part of the elementary stream data, see clause 6.2.1 of [29]).

VBV Buffer Draining

The VBV buffer is instantaneously emptied at the VOP decoding times (see Figure 4.1, which shows the VBV occupancy for a VOL, $vbv(t)$, as a function of time). This instantaneous removal property distinguishes the VBV buffer model from a real bitstream buffer. This way, the model accommodates the worst-case scenario, i.e., the case where the decoder stores all the encoded data for the current VOP in its bitstream buffer before it starts decoding it.

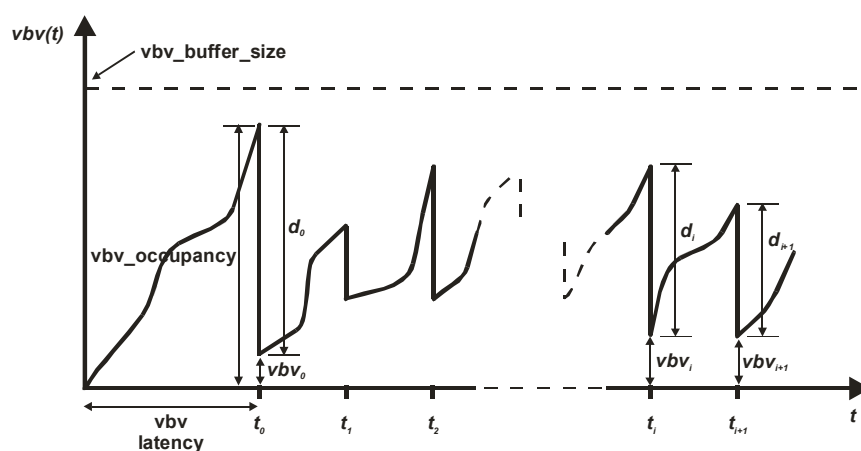


Figure 4.1 – Dynamics of the VBV occupancy for one VOL

VOP Decoding Time Computation

In order to keep a good estimate of the decoder bitstream buffer occupancy, the encoder needs to know when the encoded data shall be removed from the VBV buffer, i.e., the VOP decoding times. Since the VOP time information carried in the VOP ES is the VOP composition time, the encoder needs to compute the corresponding VOP decoding time from this information. In MPEG-4 Visual [29], the time at which each VOP must be available in the composition memory for composition is given by this VOP composition time plus a fixed delay – *VCV Latency* (see description in Section 4.2.2). This delay sets the minimum latency of the decoding process.

The usage, in some profiles, of B-VOPs, which may be coded using more than one prediction (i.e., may be predicted from preceding I- or P-VOPs – *forward prediction*, and from upcoming I- or P-VOPs – *backward prediction*, as shown in Figure 4.2), implies that the VOP decoding order and the VOP composition order are different for these cases. In fact, some VOPs must be decoded in advance, i.e., before their natural composition order, because they are needed for the prediction of other VOPs. In terms of decoder operation, this implies additional delay and VOP memory for the decoding and storage of the backward predictions.

The following example illustrates this situation showing an ES with a variable number of B-VOPs between P-VOPs and the corresponding acquisition, decoding, and presentation orders

for the several VOPs (including some necessary delay):

Acquisition order: $I_0 P_1 P_2 \mathbf{B}_4 P_3 \mathbf{B}_6 P_5 \mathbf{B}_8 \mathbf{B}_9 P_7 \mathbf{B}_{11} \mathbf{B}_{12} P_{10}$

Decoding order: $I_0 P_1 P_2 P_3 \mathbf{B}_4 P_5 \mathbf{B}_6 P_7 \mathbf{B}_8 \mathbf{B}_9 P_{10} \mathbf{B}_{11} \mathbf{B}_{12}$

Presentation order: $I_0 P_1 P_2 \mathbf{B}_4 P_3 \mathbf{B}_6 P_5 \mathbf{B}_8 \mathbf{B}_9 P_7 \mathbf{B}_{11} \mathbf{B}_{12} P_{10}$

MPEG-4 Visual [29] clearly defines the time instants at which a given VOP has to be available at the bitstream buffer (all its bits) for decoding; these time instants have to be computed by the encoder in order to track the occupancy of the decoder bitstream buffer.

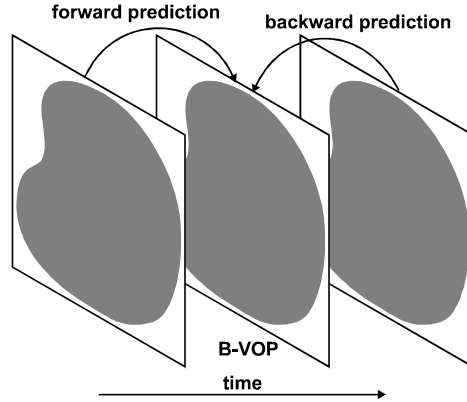


Figure 4.2 – B-VOP and corresponding forward and backward predictions

Let d_i be the size in bits of the i -th VOP plus any immediately preceding GOV header, where i is the VOP index in decoding order. Notice that a VOP includes any trailing stuffing code words before the next start code and, due to start code byte alignment, its size is always a multiple of 8 bits.

Let t_i be the decoding time of the i -th VOP in decoding order. As shown in Figure 4.1, all bits of VOP i , d_i , are removed from the VBV buffer instantaneously at instant t_i . Notice that the VBV buffer is considered to underflow whenever, at decoding time, t_i , not all the bits for the corresponding VOP are in the buffer.

Let τ_i be the composition time (or presentation time in a no-compositor decoder⁸) of the i -th VOP, defined by the timing information in the VOP header.

The relation between the decoding time and the composition time depends on the following factors:

- VOP coding type, e.g., I, P or B
- Low-delay indication⁹
- Scalability usage
- Sprite usage

⁸ This happens when the output of the decoder is sent directly to the presentation, thus there is no compositor needed.

⁹ The *low_delay* parameter is a flag which when set to “1” indicates that the ES does not contain B-VOPs, which means that frame reordering is not required.

Depending on these factors, two different cases may appear in a video elementary stream:

1. The VOP decoding time is the same as the VOP composition time.
2. The VOP decoding must be anticipated relatively to its composition time because the VOP is needed for building the predictions for other VOPs.

Although the MPEG-4 video buffering verifier assumes that each VOP is completely decoded at the VOP composition time plus the VCV Latency, real systems may add an additional delay to this time to cope with the extra delay of the decoding operation. Since this delay is not normative and may be arbitrarily low, depending on the implementation, the MPEG-4 VBV model assumes it is zero.

- **VOP Decoding Time = VOP Composition Time**

The decoding time equals the composition time, i.e., $t_i = \tau_i$, if at least one of the following conditions is verified:

1. **VOP Coding Type = B-VOP** – Since B-VOPs are not used for the prediction of other VOPs, its composition and decoding times are equal.
2. **Low-delay = 1** – In this case, the *low_delay* flag in the VOL header indicates that the VOL contains no B-VOPs and thus no reordering of VOPs is needed at the decoder.
3. **Scalability usage = 1** – VOPs in the enhancement layer are composed in the same order as they are decoded. Even if B-VOPs are used in the enhancement layer, they are never predicted from future VOPs, in composition order, of the same layer (but from layers below).
4. **Sprite usage = 1** – Since the combination of sprite coding and B-VOPs is not allowed in MPEG-4, when the sprite tool is used in a video object no reordering of VOPs is needed at the decoder side. In this case, the decoding time equals the composition time for all VOPs of the corresponding VO.

- **VOP Decoding Time \neq VOP Composition Time**

The composition time of I- and P-VOPs is delayed until all immediately preceding B-VOPs, in composition order, have been composed, i.e., the anchor VOPs, I- or P-, have to wait to be composed until all the B-VOPs with earlier composition time have been composed.

In this case, the I- and P-VOPs decoding time is equal to the composition time of the preceding non B-VOP, i.e., $t_i = \tau_{p(i)}$, where i is the index of the VOP itself and $p(i)$ is the index of the nearest preceding non B-VOP to VOP i .

Since $\tau_{p(i)}$ has no meaning for the first decoded VOP (i.e., for $i = 0$), which shall not be a B-VOP although the timing structure is determined by the B-VOP decoding times, the decoding time t_0 is not defined in this case. Nevertheless, it is assumed that the first decoded VOP is available for composition at τ_0 .

VBV MODEL CONSTRAINTS

This section applies to all the cases considered in the VBV model except for basic sprites (see Section 2.4), which have a special treatment. The first I-VOP of a sprite VO is divided into N sections of 396 MBs and each section is treated as a different VOP. The remaining S-VOPs

are treated as any other VOP.

Constraints on VBV Occupancy

The main constraint imposed to the VBV model is that each VOL VBV buffer shall never overflow or underflow. The VBV buffer occupancy for a VOL, immediately following the removal of VOP i from the bitstream buffer, vbv_i , as shown in Figure 4.1, can be iteratively defined by equation (4.1)¹⁰

$$\begin{aligned} vbv_0 &= vbv_{0^-} - d_0 \\ vbv_{i+1} &= vbv_i + \int_{t_i}^{t_{i+1}} R_{vol}(t)dt - d_{i+1} \quad \text{for } i \geq 0 \end{aligned} \quad (4.1)$$

where vbv_{0^-} is the initial VBV occupancy just before the removal of the first VOP from the buffer, d_0 is the number of bits for the first VOP in the ES, and d_i is the number of bits for VOP i .

The conditions that the VBV buffer never overflows or underflows, can then be expressed by

$$\begin{cases} vbv_i + d_i \leq vbv_{BS} \\ vbv_i \geq 0 \end{cases} \quad \text{for all } i,$$

where vbv_{BS} is the VBV buffer size in bits for the relevant VOL.

Constraints on VOP coded size

The VBV occupancy constraints for a VOL impose that the coded VOP size must always be less than the VBV buffer size, i.e., $d_i < vbv_{BS}$ for all i .

VBV MODEL RESTRICTIONS FOR SHORT VIDEO HEADER

The short video header mode has been defined to achieve compatibility with the H.263 standard [8]. In this case, the VBV model is very similar to the H.263 Hypothetical Reference Decoder model [8].

If video with the short video header is in use (i.e., $short_video_header = 1$), then the parameter vbv_buffer_size is not present and the VBV operation for the corresponding ES is described as follows:

Buffer Inspection Instants

The buffer is initially empty at the start of the decoding operation (i.e., at $t = 0$, when the first bit of the first VOP with short header enters the bitstream buffer), and its fullness is subsequently inspected at time intervals of 1001/30000 seconds (i.e., at $t = 1001/30000$, $2002/30000$, etc.).

VOP Removal

If at least one complete VOP with short header is in the buffer at the checking time, than all

¹⁰ To avoid accumulating errors, the MPEG-4 Visual standard specifies that real-valued arithmetic should be used to compute vbv_i .

the data for the earliest VOP in the buffer is instantaneously removed.

Buffer Fullness after VOP Removal

After the removal of a VOP from the bitstream buffer, the buffer fullness, vbv_i , shall verify the following conditions

$$\begin{cases} vbv_i \geq 0 \\ vbv_i < (4 \cdot R_{\max} \cdot 1001) / 30000 \end{cases} \quad \text{for all } i,$$

where R_{\max} is the maximum bit rate, in bits per second, allowed for the relevant profile@level. Furthermore, the total VBV buffer fullness at any time shall not exceed a maximum value of $vbv_{\max} = k \cdot 16384 + (4 \cdot R_{\max} \cdot 1001) / 30000$ bits, where k is a constant that depends on the picture format used and is given in Table 4.1¹¹.

Table 4.1 – Values of k for the allowed picture formats

Picture format	K
Sub-QCIF	4
QCIF	4
CIF	16
4CIF	32
16CIF	64

Number of bits per VOP

The number of bits used to code any single VOP, d_i , shall not exceed $k \cdot 16384$ bits.

VOP Decoding Times

Since the MPEG-4 short video header tool aims at minimizing the end-to-end delay, no restrictions are imposed regarding the decoding start times; this means that it is up to the decoder designer to decide when decoding starts. However, if a decoder starts to decode too early, it may have to wait more frequently for the arrival of complete pictures at the decoder bitstream buffer.

4.2.2 Video Complexity Verifier (VCV) Definition

The MPEG-4 VCV model defines a set of rules and limits for examining a set of ESs building a visual scene to control if the required amount of decoder processing power is less than the maximum complexity specified for the given profile and level, both measured in MBs per second. This model is applied to all MBs of all ESs of the scene together.

The VCV applies to video objects encoded as a combination of I-, P-, B- and S-VOPs¹². A separate VCV model applies to still texture objects [29]. Face animation and mesh objects are not constrained by this model.

¹¹ MPEG-4 Visual refers the possibility of specifying larger values of k ; however, this is not currently supported by the syntax [29].

¹² For sprites, a hypothetical number of MBs is defined for each S-VOP.

The coded video bitstreams for a certain scene shall be constrained to globally comply with the requirements of the VCV defined in the following sections.

VCV MODEL PARAMETERS

The VCV model consists in two virtual buffers accumulating the number of MBs in the encoded data:

1. The **VCV Buffer** accumulates all MBs of all VOLs for the scene.
2. The **Boundary MB VCV Buffer (B-VCV)**¹³ accumulates only the boundary MBs of all VOLs for the scene.

Notice that boundary MBs (i.e., MBs including shape information which is not totally transparent or totally opaque) are included in both the VCV and the B-VCV buffers.

The VCV model is defined by the size of the buffers mentioned above, the corresponding draining rates (i.e., the VCV and B-VCV decoding rates), and the latency of the VCV model (which depends on the VCV buffer size and VCV decoding rate).

VCV Buffer Sizes and VCV Decoding Rates

Each VCV buffer can be seen as a queue, instantaneously filled with all the MBs of each VOP at the VOP decoding time, and delivering MB encoded data to the decoding process at a constant rate.

The size of each VCV buffer, respectively *vcv_buffer_size* and *boundary_vcv_buffer_size*, defines the maximum number of MBs that a given decoder can instantaneously have in the decoding queue to process, i.e., the maximum occupancy of the VCV buffers in MB units. In the current MPEG-4 Visual specification [29], the two buffers have always the same maximum dimension for all profile@level combinations.

These MBs are consumed by the decoder, from each buffer, at a given VCV decoding rate, in MB/s, as specified for each profile@level. The VCV decoding rate, H , specifies the draining rate of the VCV buffer while the B-VCV decoding rate, H_B , specifies the draining rate of the B-VCV. Together they define the maximum speed of the decoding process. As can be seen in Table 2.6, the B-VCV decoding rate, H_B , is typically half the VCV decoding rate, H , i.e., $H = 2H_B$.

For each profile@level combination, MPEG-4 Visual defines the maximum VCV buffer size (the same for the VCV and B-VCV buffers) and the draining rates for the VCV and B-VCV buffers.

VCV Latency

The VCV Latency, L , is defined as the time it takes to decode a full VCV buffer, and thus is given by the following equation

$$L = \frac{vcv_buffer_size}{H} \quad (4.2)$$

This parameter imposes a minimum latency in the decoding process, as explained in Section 4.2.1. Notice that, by definition, the latency of the VCV model is imposed by the VCV buffer

¹³ The B-VCV is only defined for profiles supporting arbitrarily shaped video objects.

not by the B-VCV buffer. Since the B-VCV decoding rate, H_B , is typically half the VCV decoding rate, H , this means that it is not possible to decode a full B-VCV during a time interval of L , since the two buffers have the same size. This implies that at full decoding rate, the amount of boundary MBs in the scene cannot exceed 50 % of the total number of MBs.

VCV OCCUPANCY DYNAMICS

The VCV dynamics simulates the VOP decoding process. At the VOP decoding times, the VOP encoded data is added to the VCV buffers and is removed from these buffers as the decoding process progresses. The time instant at which a given VOP is completely decoded depends on the amount and type of MBs to be decoded, the occupancy of the VCV buffers at the VOP decoding time, and the maximum decoding speed specified through the VCV decoding rates for the profile@level in question.

VCV Buffer Filling

Let M_i be the total number of MBs in VOP i , and M_{Bi} the number of boundary MBs in the same VOP. For S-VOPs, M_i is given by the hypothetical number of MBs in a S-VOP, MB_{S-VOP} , as specified in Annex D of [29].

The VCV buffer is empty at the start of decoding and is filled instantaneously with encoded data at VOP decoding times as the decoding process advances. At the VOP decoding time, t_i , defined above, M_i is added to the VCV buffer occupancy, $vcv(t)$, and simultaneously M_{Bi} is added to the B-VCV buffer occupancy, $vcv_B(t)$.

VCV Buffer Draining

The VCV buffers occupancies decrease linearly at rates H and H_B , respectively for the VCV and B-VCV buffers, until they are zero or until the next VOP decoding time, t_{next} , where t_{next} is the earliest VOP decoding time greater than t_i for any VOP of any ES of the scene.

If the occupancy of the VCV buffers becomes zero, the VCV model decoder becomes idle and remains idle until t_{next} , as exemplified in Figure 4.3.

VOP Decoding Duration

In order to avoid the violation of the VCV model, each VOP must be decoded in time. The interval of time where VOP i is being decoded extends from s_i to e_i which are defined by equation (4.3)

$$\begin{aligned} s_i &= t_i + \max \left[\frac{vcv(t_i)}{H}, \frac{vcv_B(t_i)}{H_B} \right] \\ e_i &= t_i + \max \left[\frac{(vcv(t_i) + M_i)}{H}, \frac{(vcv_B(t_i) + M_{Bi})}{H_B} \right] \end{aligned} \quad (4.3)$$

where $vcv(t_i)$ is the VCV occupancy before the MBs representing VOP i , M_i , are added to $vcv(t)$, H is the VCV decoding rate, $vcv_B(t_i)$ is the B-VCV occupancy before the boundary MBs of VOP i , M_{Bi} , are added to $vcv_B(t)$, and H_B is the B-VCV decoding rate. Notice, that according to equation (4.3), the VOP decoding only starts after the two VCV buffers become

empty, which means that the decoder only starts processing a given VOP after having completely finished the previous one in the decoding queue.

VCV MODEL CONSTRAINTS

Compliance regarding the VCV model can only be guaranteed if the set of ESs building a scene fulfills the constraints imposed by the VCV model relatively to the occupancy of the VCV buffers and the VOP decoding duration defined as follows:

Constraints on VCV Occupancy

A given set of visual ESs building a scene conforms with a given profile@level with respect to the VCV model if they never overflow the VCV buffers.

When the VCV buffers become empty, the decoder simply remains idle and the VCV buffer occupancies, $vcv(t)$ and $vcv_B(t)$, remain unchanged during the idle period; this is illustrated in Figure 4.3, which shows the occupancy of a VCV buffer, $vcv(t)$, as a function of time.

Constraints on VOP Decoding Duration

In addition to not overflowing the VCV buffer, the decoding of each VOP i must be completed by $\tau_i + L$ (composition time plus the latency of the VCV decoding process). Notice that the latency L of the VCV decoding process is constant for all VOPs.

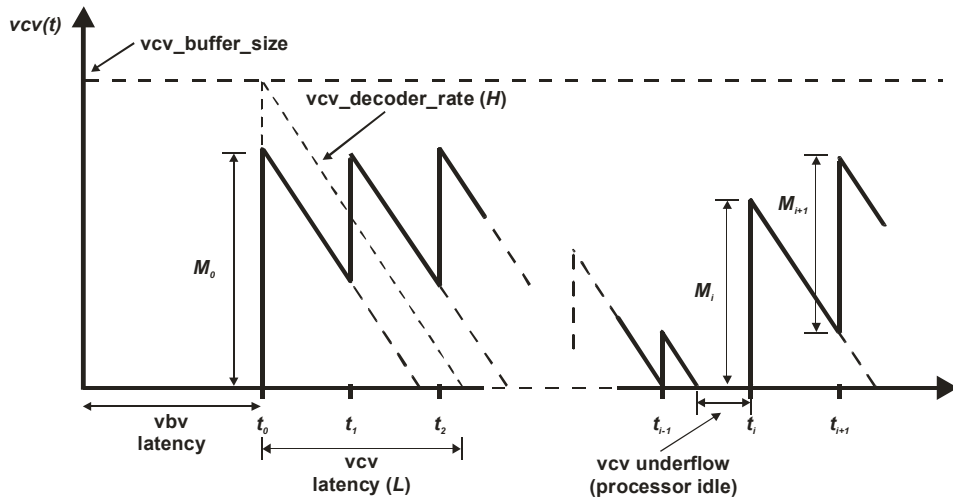


Figure 4.3 – Dynamics of the VCV occupancy

4.2.3 Video Reference Memory Verifier (VMV) Definition

The MPEG-4 VMV model defines a set of rules and limits for examining the set of ESs building a visual scene to control if the required amount of decoder picture memory, measured in MB units, is less than the maximum memory specified for the chosen profile and level. The VMV models the memory requirements of all VOLs of all VOs in the scene (this model assumes a common memory space, shared by all VOLs of all VOs).

The VMV applies to video objects encoded as a combination of I-, P-, B-, S-VOPs, and still texture objects. Face animation, mesh objects, and I-VOPs in basic sprite sequences are not constrained by this model.

The coded video bitstreams shall be constrained to comply with the requirements of the VMV defined in the following sections.

VMV MODEL PARAMETERS

The VMV model consists of a MB buffer that accumulates all the decoded MBs of all VOPs and stores them until they are no longer needed for the prediction of other VOPs. The VMV model is defined by the size of this buffer, the *vmv_buffer_size*, specifying the maximum amount of decoded MBs that the decoder can store at any time instant, see Table 2.6.

VMV OCCUPANCY DYNAMICS

The VMV dynamics simulates the decoded VOP memory allocation and de-allocation process. As each VOP is being processed, the decoder needs to allocate memory to store the decoded data. This data remains in the decoder memory until it is no longer needed, e.g., for prediction. At this point in time, the memory allocated to store this data is instantaneously released and can be used again.

VMV Buffer Filling

The VMV buffer is initially empty and is filled with decoded data as each MB is decoded (see Figure 4.4). For I-, P-, and B-VOPs, the amount of picture memory required for the decoding of the i -th VOP is defined as the number of MBs in the VOP, M_i . This memory, called reference memory in the MPEG-4 Visual standard [29], is consumed at the same constant rate specified for the VCV buffer (i.e., H MB/s) as the decoding process takes place. This solution contemplates the worst case scenario in terms of memory consumption since the VCV has the highest decoding rate (consumes memory faster than the B-VCV) and accumulates all the MBs (consumes all the needed memory). As illustrated in Figure 4.5, the VMV buffer reaches its maximum occupancy when the VCV becomes idle; however, the VOP is only completely decoded at e_i , i.e., when the B-VCV becomes idle.

For S-VOPs, the amount of picture memory required for the decoding of the VOP is defined as the number of MBs in the reconstructed VOP. The memory used for storing the sprite is not constrained by the VMV model.

The decoding duration of VOP i , T_i , is identical for the VCV and VMV models and starts at s_i and ends at e_i , as defined in Section 4.2.2 and illustrated in Figure 4.5.

VMV Buffer Draining

The VMV draining depends on the coding type of the VOP being decoded, as explained in the following:

- **I- and P-VOPs** – At the VOP composition time (or presentation time for a no-compositor decoder) plus VCV latency, $\tau_i + L$, the total memory allocated to the previous I- or P-VOP in the decoding order is instantaneously released [29].
- **B-VOPs** – At the VOP composition time (or presentation time for a no-compositor decoder) plus VCV latency, $\tau_i + L$, the total memory allocated to the current B-VOP is instantaneously released [29].

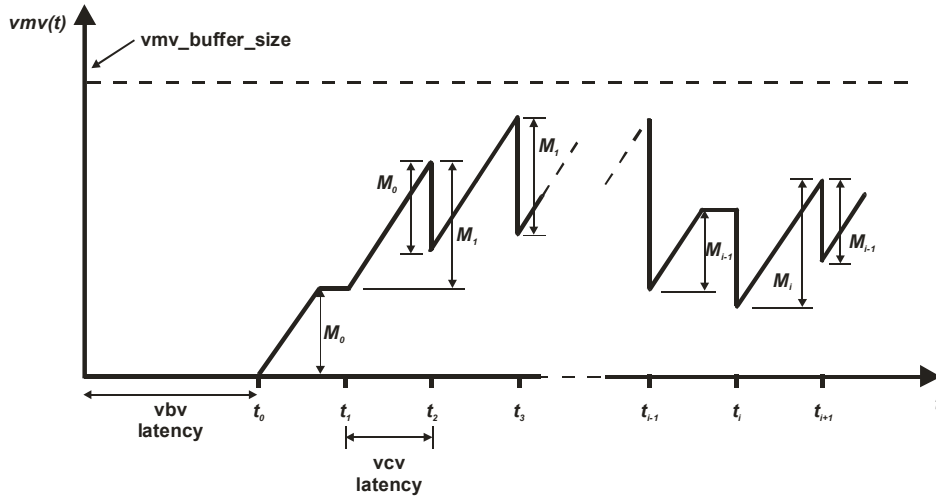


Figure 4.4 – Dynamics of the VMV occupancy

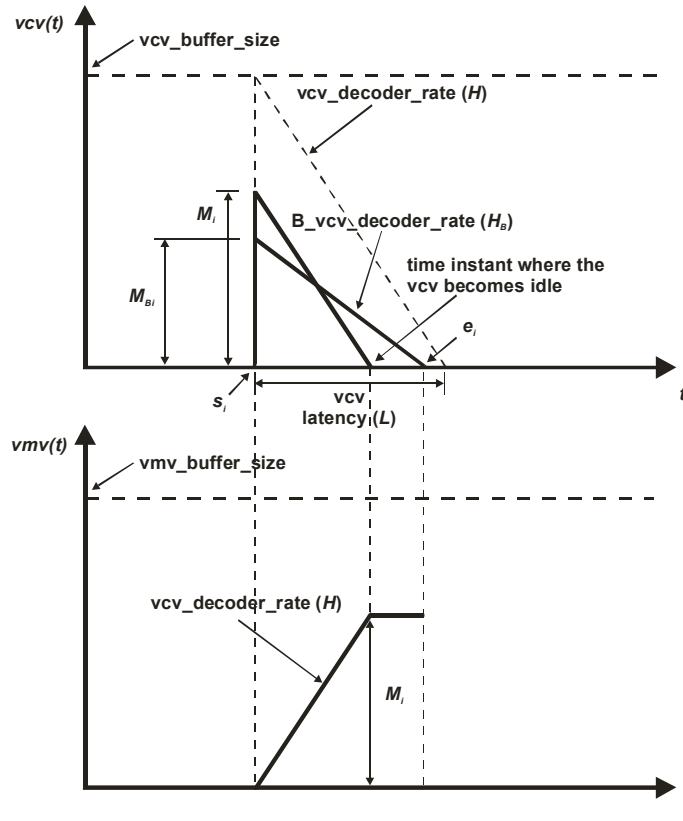


Figure 4.5 – Relation between VCV and VMV occupancies

VMV MODEL CONSTRAINTS

A given set of visual ESs building a scene conforms with a given profile@level, with respect to the VMV model, if it never overflows the VMV buffer.

4.2.4 Interaction between the VBV, VCV, and VMV Models

A given set of ESs building a visual scene is considered compliant with a given profile and level if it fulfills all the constraints defined by the several video buffering verifier models. Bitstream compliance with a given profile@level guarantees that the resources required at the decoder do not exceed a certain pre-defined amount corresponding to the relevant profile@level. Moreover compliance defines strict timing for the completion of decoding and composition of VOPs as explained in the following:

- The VBV model defines the time at which the coded bits for each VOP are available for decoding and the time at which they should be removed from the VBV buffer – *the coded bits for each VOP should be removed from the VBV buffer at the VOP decoding times, t_i , computed from the composition time information in the video ES or conveyed by systems decoding time stamps.*
- The VCV model defines the decoding speed of the MB data, and, thus, the time at which each VOP is available for composition – a given VOP should be available for composition, at most, at the VOP composition time plus the VCV latency, i.e., at the time it is supposed to be available to the compositor.
- The VMV model defines the amount of picture memory allocated at each time instant and the time it should be released – a given VOP should be removed from the VMV buffer at its composition time plus the VCV latency (B-VOP) or at the composition time plus the VCV latency of the next P- or I-VOP (I- or P-VOPs).

The various models are independent but interact with each other in the following way:

The Decoder Cannot Decode Arbitrarily in Advance

From a decoding point of view, it could be advantageous to process the incoming data as far in advance as possible; this is however constrained by two factors:

- The decoder can only start decoding if the bits are available for decoding – *constraint imposed by the VBV model.*
- As the decoder decodes the incoming data, it generates macroblocks that consume picture memory; if the decoder decodes too fast, it may not have enough picture memory to store the decode data – *constraint imposed by the VMV model.*

The Decoder Cannot Decode Too Late

If the decoder starts decoding too late, then it may not be able to complete the decoding on time, and the following situations may occur:

- The VOP bits may be removed from the bitstream buffer before they could be decoded.
- The composition time for the current VOP may arrive without the VOP being completely decoded.
- The time to release the picture memory required for the prediction of the current VOP may arrive before the VOP could be decoded.

In order to avoid these situations, the video buffering verifier mechanism imposes strict times for starting and ending any VOP decoding – *constraint imposed by the VCV model.*

The video buffering verifier models provide the mechanism allowing any encoder to produce bitstreams that will be decodable by any decoder compliant with the selected profile@level. This mechanism allows to simultaneously limit the amount of decoding resources needed at the receiving terminals as well as to ensure the timely reconstruction of the encoded information.

It is important to highlight that it is a major task of the encoder to simulate each of the video buffering verifier models in order to produce bitstreams compliant with the intended profile and level. If any of these models tends to be violated, the encoder has to take appropriate countermeasures to avoid it. Although the video buffering verifier is defined for the decoders, it is in fact a major module of any encoder generating compliant sets of bitstreams.

4.3 MPEG-2 and H.263 Video Verification Mechanisms

As referred above, the idea of using a video buffering verifier mechanism to monitor and control the decoding resources needed by encoded video information is already present in previous video coding standards. In this section, the major differences and similarities between the MPEG-4 solution and two of the most widely know video buffering verifier mechanisms are presented: the MPEG-2 video buffering verifier (VBV) [10] and the H.263 hypothetical reference decoder (HRD) [8].

4.3.1 MPEG-2 Video Buffering Verifier

In MPEG-2, video data is typically encoded with fixed spatial and temporal resolutions, which sets *a priori* the complexity of the encoded data with respect to the amount of decoder memory and the maximum number of MB/s that a given decoder has to process. In this case, the uncertainty associated to the complexity of the encoded data was simply related to the variation of the number of bits per picture this means the bit rate.

The MPEG-2 Video solution to constrain the encoded data produced by the encoder is based on a virtual buffer approach similar to the MPEG-4 video rate buffer verifier, simply known as video buffering verifier [10].

The MPEG-2 VBV model consists on a hypothetical decoder, conceptually connected to the output of the encoder via an input buffer known as the VBV buffer, whose size is specified in the sequence header through the *vbv_buffer_size* field. Encoded data enters the VBV buffer at a piecewise constant rate, for each picture, and is instantaneously removed from the buffer at the corresponding picture decoding times.

Contrary to what happens in the MPEG-4 VBV, the MPEG-2 model assumes instantaneous picture decoding at the picture decoding time. This is a reasonable approach for the MPEG-2 video scenario since decoder implementations can add an arbitrary, non-normative, fixed delay to the decoding process to cope with the necessary margin to completely decode the incoming encoded data. Faster decoders will need a shorter delay while slower decoders will require a higher safeguard delay at the expense of some additional buffering space.

For MPEG-4 video, such approach is not suitable since the large variation of the incoming encoded data, in terms of decoding complexity, mainly due to a higher variation of the number of MB/s, the more or less large choice of object types for each object in the scene depending on the profile, and the fact that MBs can be opaque, transparent, or boundary, would lead to decoders designed to cope with worst-case scenarios, which in average would result in very over dimensioned decoders. For this reason, MPEG-4 specifies a fixed decoding delay (the

VCV latency) in order to set some minimum acceptable bounds in terms of the decoding capacity of the decoder.

A bitstream conforming to a given MPEG-2 Video profile@level shall never overflow the VBV buffer, and additionally, when *low_delay* is zero, the bitstream shall not cause the VBV buffer to underflow also.

When *low_delay* is one, buffer underflow is allowed. Buffer underflow occurs if not all picture data is available at the bitstream buffer at the normally expected picture decoding time. In this case, the VBV buffer shall be re-inspected at later time instants until all picture data is present in the VBV buffer, resulting in a picture decoding delay.

MPEG-2 VBV MODEL PARAMETERS

The MPEG-2 VBV model is defined by the four following parameters: *bit_rate*, *vbv_buffer_size*, *low_delay*, and *vbv_delay*. Together they determine the behavior of the VBV model.

Bit Rate

For variable bit rate encoding, the *bit_rate* parameter specifies the maximum input data rate to the MPEG-2 VBV buffer. For constant bit rate encoding, it specifies the actual VBV filling rate.

VBV Buffer Size

The *vbv_buffer_size* specifies the size in bits of the MPEG-2 VBV buffer, and thus sets the minimum bitstream memory required at the decoder.

Low-delay

The *low_delay* parameter is a flag that when set to zero, indicates that the bitstream may contain B-pictures and the VBV buffer is not allowed to underflow; when set to one, this flag indicates that the bitstream contains no B-pictures, thus frame reordering is not needed. In this case, the VBV is allowed to underflow, which means that at the picture decoding time it is possible that not all the data for the next picture to be decoded is in the decoding buffer; these pictures are called *Big Pictures*. When a *Big Picture* occurs, the VBV buffer shall be re-inspected at later time instants until all picture data is present in the VBV buffer for decoding.

VBV Delay

The *vbv_delay* is a parameter sent in the picture header of each picture indicating the time that the picture should stay in the bitstream buffer before being extracted for decoding.

MPEG-2 VBV OCCUPANCY DYNAMICS

The MPEG-2 VBV dynamics specifies the process by which the VBV is filled and drained. As in MPEG-4, this process is mainly driven by the time instants at which the pictures are removed from the VBV. In MPEG-2, however, the time instant at which the decoding process starts is slightly different, as described below.

VBV Filling and Decoding Start Time

The VBV buffer is initially empty and is filled with encoded data at a piecewise constant rate. Decoding starts at the time instant specified by the *vbv_delay* of the first picture in the

sequence or when the VBV is completely full, depending on whether the *vbv_delay* field carries information or not.

A) With *vbv_delay*

The *vbv_delay* value, when different from 0xFFFF, indicates the amount of time a picture should stay in the VBV buffer before being extracted for decoding. The *vbv_delay* field indicates the number of periods of the 90 kHz system clock to wait, after receiving the last bit of the picture start code, before extracting the picture data from the buffer for decoding. The systems time, TS_i , at which picture i arrives at the decoder (VBV buffer) is then given by

$$TS_i = DTS_i - vbv_delay_i \quad (4.4)$$

where DTS_i is the systems picture decoding time stamp. In this case, the model assumes that each picture is delivered to the VBV buffer at a bit rate, R_i , given by

$$R_i = \frac{d_i}{TS_{i+1} - TS_i} \quad (4.5)$$

where d_i is the number of bits in picture i , and $R_i \leq R_{\max}$.

If R_i is constant throughout all pictures of the sequence then the transmission is considered to be constant bit rate; otherwise it is considered variable bit rate.

The *vbv_delay* for the first picture in the bitstream defines the amount of time the decoder has to wait before removing the first picture from the decoding buffer.

Notice that, contrary to what happens in the MPEG-4 VBV, the MPEG-2 VBV model defines a temporal window for measuring the bitstream bit rate at which the VBV buffer is filled. This temporal window is given by the time intervals $[TS_i, TS_{i+1}]$.

Additionally, in MPEG-2, the decoding start time is derived from the initial *vbv_delay* that defines the “VBV latency”, while in MPEG-4 this latency is specified through the initial occupancy of the VBV buffer conveyed through the *vbv_occupancy* parameter.

B) Without *vbv_delay*

A *vbv_delay* value equal to 0xFFFF indicates that the VBV buffer is filled at the maximum data rate, R_{\max} , specified in the *bit_rate* field of the sequence header. When the VBV buffer is full after being filled at R_{\max} during some time, no more data can be accommodated in the VBV buffer. In this case, the encoder is responsible for waiting until some data is removed from the VBV buffer to avoid overflowing it.

VBV Inspection Time Instants

After decoding has started, the VBV buffer is inspected at successive time instants determined by the picture rate and temporal coding structure, notably the occurrence of B-pictures. At these time instants, the number of bits in the VBV buffer shall be less than the buffer size and greater than zero. The process of removing encoded pictures from the VBV buffer depends, however, on the *low_delay* flag value:

A) With $low_delay = 0$

When low-delay is zero, the bitstream may contain B-pictures and picture reordering is necessary. In this case, the VBV buffer shall never underflow, requiring that all picture data be present at the buffer at the picture decoding time.

Each time the buffer is inspected, all the data for the earliest picture in the buffer shall be removed instantaneously. If the buffer does not contain all that picture data, the bitstream is considered non-compliant with the profile@level in question and the behavior of the decoder is undefined.

B) With $low_delay = 1$

When operating in low-delay mode, there may exist situations where the VBV buffer needs to be inspected several times before all the data for a picture is in the buffer. This type of pictures is called *Big Picture*. Nevertheless, the number of bits for a big picture shall never cause the VBV to overflow otherwise data may be lost.

It can happen in real systems that the *DTS* and the *vbv_delay* values are incorrect (i.e., they do not reflect the actual size in bits of the incoming picture) for some big pictures since real-time encoders may realize too late that they are producing a big picture. Nevertheless, equations (4.4) and (4.5) are still valid for big pictures.

Whenever an encoder sends a big picture, the receiver repeats the display of the previously decoded picture until the big picture is decoded. This can be seen as an increase in the end-to-end delay between picture acquisition and picture display. To reduce this delay to its normal value, the encoder usually skips the encoding of a few pictures following the big picture.

Notice that the concept of big pictures (VOPs) does not exist in MPEG-4 since it involves an underflow of the VBV model which, by definition, is not allowed in the MPEG-4 VBV model.

4.3.2 H.263 Hypothetical Reference Decoder

As MPEG-2 Video [10], the H.263 standard [8] specifies a video verification mechanism aiming at limiting the variability of the number of bits per picture. This mechanism, similar to the MPEG-4 VBV model for the short video header mode described in Section 4.2.1, is called Hypothetical Reference Decoder (HRD) and consists on a virtual buffer filled with the encoded data output from the encoder and inspected at regular times for picture removal. Like in MPEG-2, the uncertainty associated to the complexity of the encoded data is simply related to the variation of the number of bits per picture this means the bit rate.

HRD MODEL PARAMETERS

The HRD model consists of a receiving buffer whose size depends on the following parameters, negotiated between the two terminals involved at the beginning of the communication:

- The maximum video bit rate during the connection in bits per second – R_{\max} .
- The maximum number of bits per picture allowed in the bitstream in 1024 units – B_p .

Notice that, in MPEG-4 Visual [29], the equivalent parameters are either transmitted in the bitstream or derived from systems level configuration information.

The relation between these parameters and the HRD buffer size, B_s , is expressed by

$$B_s = B + B_p \cdot 1024 \quad [\text{bit}] \quad (4.6)$$

where $B = 4 \cdot R_{\max} / 29.97$.

H.263 HRD OCCUPANCY DYNAMICS

The HRD is initially empty and is filled with encoded data as the encoding process proceeds. At regular times – CIF intervals (1000/29.97 ms) – the HRD buffer is inspected and, if it contains at least a complete picture, then all the encoded data for the earliest picture is instantaneously removed. The HRD model requires that immediately after removing a picture, the buffer occupancy shall be less than B (defined above). To meet this requirement, the HRD model imposes that the number of bits per picture verifies the following condition

$$d_{i+1} \geq b_i + \int_{t_i}^{t_{i+1}} R(t)dt - B \quad (4.7)$$

where d_{i+1} is the number of bits in picture $i+1$, t_i is the time instant when the i -th picture is removed from the HRD buffer, b_i is the HRD buffer occupancy immediately after t_i , and $R(t)$ is the video bit rate at time t . This means that the encoder may need to use stuffing data to meet this requirement.

The main difference between the MPEG-4 VBV model without short video header and H.263 or MPEG-4 VBV model with short video header is related to the way each of the solutions deals with the end-to-end delay.

While in the MPEG-4 case (without short video header) the minimum end-to-end delay is conditioned by the syntax and semantics of the bitstreams, i.e., the minimum end-to-end delay is imposed by the latencies of the VBV and VCV models, in the H.263 case the objective is to minimize this delay and thus it is not normatively specified.

This minimization of the end-to-end delay is usually achieved by using variable frame rate encoding. Typically, an encoder will not transmit pictures that follow a picture that has been encoded with significantly more bits than the nominal number of bits (i.e., bit rate/picture rate). In this situation, the decoder will output the previously decoded picture in place of these non-transmitted pictures before the one with many bits. If the objective was to maintain the same picture intervals as at the encoder, the result would be a constant delay determined by the picture coded with the highest number of bits. However, with this more flexible strategy, it is only the (big) picture that is delayed, and the overall average delay can be kept low.

4.4 MPEG-4 Video Buffering Verifier Integration Architecture

This section proposes and discusses an architecture for the integration of the MPEG-4 video buffering verifier in an MPEG-4 video encoder.

Although the video buffering verifier mechanism is essentially described in terms of decoder operation, it is a task of the encoder to implement it and to guarantee that it is not violated. For this, the encoder has to “shape” the encoded data in a way that it does not violate the constraints imposed by this mechanism. Such task is mainly dealt with by the rate control mechanism that takes into account the status of the several video buffering verifier buffers for the best control of the encoder [18].

Figure 4.6 presents a block diagram where the video buffering verifier mechanism is integrated into a multiple object MPEG-4 video encoder showing the several types of interaction between the video encoder and the rate control mechanism. For the purpose of the current analysis, the video encoder is composed of the following blocks:

- **Scene Buffer** – Stores the VOPs of all VOLs of all VOs before encoding; it may store only the VOPs existing at each instant (low-delay encoding), or it may store all the VOPs of all VOs before coding any VOP (off-line encoding).
- **Symbol Generator** – Converts the VOP's texture and shape (when applicable) information into representation symbols (syntactic elements), such as motion vectors, quantized DCT coefficients, coding modes, etc.
- **Entropy Coder** – Efficiently encodes the representation symbols into bit codes (e.g., Huffman or binary arithmetic coding).
- **Video Multiplexer** – Combines the encoded information following the adopted syntax, producing syntactically valid video elementary streams.
- **Rate Controller** – Has the important task of controlling most of the previous blocks, given the scene characteristics, the encoding results, and the video buffering verifier feedback information, in order to produce a set of video elementary streams that do not violate the relevant profile and level constraints.

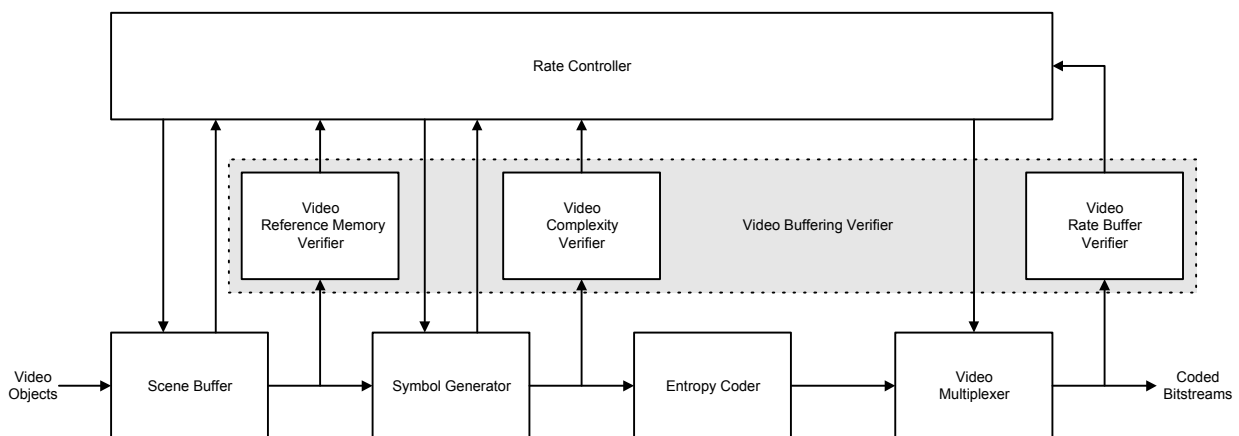


Figure 4.6 – Integration of the video buffering verifier mechanism in a MPEG-4 video encoder

The integration of the video buffering verifier mechanism with the rate control mechanism can be described as follows: each video buffering verifier model, this is the VBV, VCV, and VMV, must be sequentially verified, during the encoding process, providing feedback to the rate control module that takes this information and performs the adequate action in order to produce overall compliant bitstreams, according to the given profile and level definitions.

From an implementation point of view, the verification order of the various models is related to the degrees of freedom of the rate control mechanism to react to a possible violation of the profile and level definitions, e.g., controlling the output rate of an ES can be done during the encoding of each VOP, while controlling the amount of memory required to decode the given VOPs should be done prior to the encoding of the VOPs avoiding unnecessary encoding if the picture memory limits are not verified.

In practice, there is a verification order which is more natural, and in fact more efficient since it avoids the unnecessary performance of certain operations by only proceeding to them when certain conditions are verified. This optimal order is closely associated to the sequence of blocks in the encoder, and is presented and motivated in the following:

1. **Video Reference Memory Verification** – The first step is the verification of the amount of picture memory required to encode the scene at hand; if the picture memory available at the decoder for the selected profile and level is not enough, there is no need to verify the other models since this model would already be violated.
2. **Video Complexity Verification** – In the second step, the encoder estimates the MB decoding resources needed at the decoder given the amount of information it has to encode for a given time instant; after guaranteeing that the picture memory requirements do not exceed the profile and level definition values, the encoder must guarantee that the computational power required is also not exceed, otherwise the decoder may not be able to decode the incoming bits in time.
3. **Video Rate Buffer Verification** – Finally, the number of bits produced by the encoder is checked; the encoder must guarantee that the amount of bits produced does not violate the decoder configuration information, notably the video rate buffer size, for any of the produced ESs.

This order can be especially important in real-time encoding since, in this case, the encoder should be able to react in advance to a possible violation of any of the video buffering verifier models, otherwise compliance may be broken or a dramatic loss of quality may occur. For example, imminent overspending of memory caused by an unexpectedly large VOP should be detected before sending any data to the channel, otherwise the set of ESs being generated may no longer be considered compliant with the selected profile@level or the VOP may have to be only partially sent.

Even for offline-encoding, this verification order is important since it can avoid performing expensive optimizations to achieve the best optimal rate-distortion encoding whenever the picture memory resources or the MB decoding resources available are not enough for the time instants under consideration.

In the following sections, the various MPEG-4 video buffering verifier models will be analyzed against alternative solutions using the ideal checking order proposed above. Additionally, the major advantages and drawbacks of this mechanism are compared with some relevant alternative models.

4.5 Analysis of the Video Reference Memory Verifier

The control of the picture memory resources needed at the decoder is a major task of the encoder. However there are, at least, two major approaches for the specification and control of the video picture memory. The objective of this section is to highlight the major advantages and drawbacks of the MPEG-4 VMV approach in comparison with possible alternatives and to propose some guidelines for its implementation within an MPEG-4 video encoder.

4.5.1 Decoder Picture Memory Modeling

The VMV model provides a way to limit the amount of picture memory required at the

decoder for decoding a set of visual elementary streams building a scene. As previously referred, this picture memory is also called reference memory¹⁴ in the context of the video buffering verifier mechanism. The memory constraints are determined by the VMV buffer size and the dynamics of the VMV model, i.e., the process by which the VMV model buffer is filled and emptied.

The VMV buffer size defines the maximum amount of picture memory available at the decoder (maximum in the sense that it is there for sure; of course, nothing prevents that implementers put there more memory but the encoder cannot count on it since that would be additional memory with respect to the profile@level specification). The VMV occupancy due to a given VOP depends on its size in MB units and on the time interval during which the VOP stays in the VMV buffer.

The VMV buffer occupancy provides the encoder with the necessary information to monitor the (minimum) amount of picture memory needed at each time instant at the decoder. As such, it allows to bound the memory requirements at the decoder, since the encoder can use the information about the occupancy of this buffer to avoid that the encoded scene exceeds the memory capabilities of the decoder during any period of time.

The real picture memory used by a given MPEG-4 player is highly dependent on its implementation. It is possible, however, to identify, at least, two major types of picture memory required by any receiving terminal:

- **Decoding Memory** – To store the decoded VOPs and the corresponding predictions (reference VOPs), during the decoding process.
- **Composition Memory** – To store the decoded VOPs (composition units) and the resulting composed scene, during the composition and presentation process.

Real implementations do not really need/use different areas of memory for decoding and for composition. The decoding and composition processes can share memory among them since the decoder and the compositor may simultaneously need the decoded VOPs during some periods of time (a given decoded VOP may be simultaneously being presented to the end-user and being used as prediction for the decoding of upcoming VOPs).

4.5.2 VMV Model Approaches

Monitoring the allocated picture memory at the decoder is an essential task of the encoder to guarantee that the selected memory boundaries are not overrun. However, depending on how the previous two types of picture memory are considered in the VMV model there are several alternatives for the modeling of this capability. The following sections present two alternative VMV model approaches and compare the adopted MPEG-4 VMV model with these approaches.

GLOBAL MEMORY APPROACH

In this approach, the VMV model takes into account both the memory needed for the decoding and for the composition of the decoded VOPs. In this case, a given VOP consumes picture memory from the instant it starts being decoded until it has been presented and is no

¹⁴ The MPEG-4 specification does not provide a definition for reference (picture) memory, but only defines reference VOPs as the reconstructed VOPs that were encoded in the form of I- or P-VOPs and are used for forward or backward prediction in the decoding of P- or B-VOPs.

longer needed for the prediction of other VOPs.

The Global Memory approach potentially models more closely the memory usage in real terminals; however, since VOP composition is non normative in MPEG-4, different terminals can implement different composition methods with different memory management strategies, and thus such model would count memory resources intentionally left out of the standard.

DECODING MEMORY APPROACH

In this approach, the VMV model takes only into account the memory needed for reconstructing the encoded VOPs. Here, a given VOP consumes picture memory from the instant it starts being decoded until it has been decoded and is no longer needed for the prediction of other VOPs.

The Decoding Memory approach, although not considering all the possible picture memory needed at the receiving terminal, considers all the normative memory requirements for the decoding of a given scene, thus being more in line with the boundaries of the standard.

MPEG-4 VMV MODEL APPROACH

At a first glance, the MPEG-4 VMV model approach could be expected to follow a decoding model approach since the composition process is left out of the normative parts of the standard. However, the memory constraints imposed by the MPEG-4 VMV do not follow a pure Decoding Memory approach. For this analysis, it is convenient to consider two different situations: video ESs without and with B-VOPs.

I) Video Reference Memory Verification without B-VOPs

When the encoder does not use B-VOPs, the decoded VOPs are presented in the same order as they are decoded; moreover, the decoding time is by definition equal to the composition time, i.e., $t_i = \tau_i$. In this case, the MPEG-4 VMV specifies that a given VOP (I or P) should stay in the VMV buffer until the composition time of the following I- or P-VOP in decoding order, imposing a longer memory occupation than the one that would be necessary in a pure Decoding Memory approach, where the reference VOPs are released from the VMV buffer after the decoding of the current VOP is finished. This situation is illustrated in Figure 4.7, showing the relation between the time instants for VOP decoding, composition, and release for each of the VMV model approaches under study, when B-VOPs are not used.

Figure 4.8 shows an example with the occupancy of the VCV and VMV buffers for the encoding of a single VO with a VOP rate equal to $1/T$, without using B-VOPs. The plot of the VCV occupancy is also shown to illustrate how the VMV occupancy evolves with the VCV occupancy. In this example it is assumed that the number of boundary MBs in each VOP is always less than 50 % of the total number of MBs in the VOP. This way the decoding duration of each VOP is not constrained by the B-VCV.

It is possible to see that, in the case of the MPEG-4 VMV model, extra memory stays allocated during some periods of time in comparison with the pure Decoding Memory management approach because some VOPs are not released immediately after they become useless in terms of decoding process. In this case, the MPEG-4 VMV model follows a Global Memory approach, as can be seen in Figure 4.7.

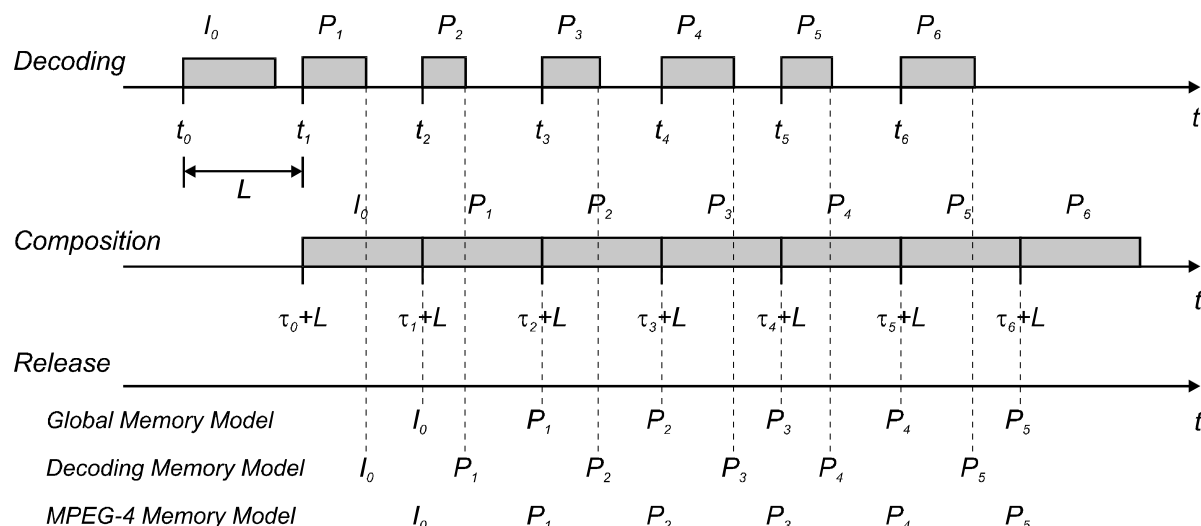


Figure 4.7 – Relation between the VOP decoding, composition, and release times for the VMV models under study, when no B-VOPs are used

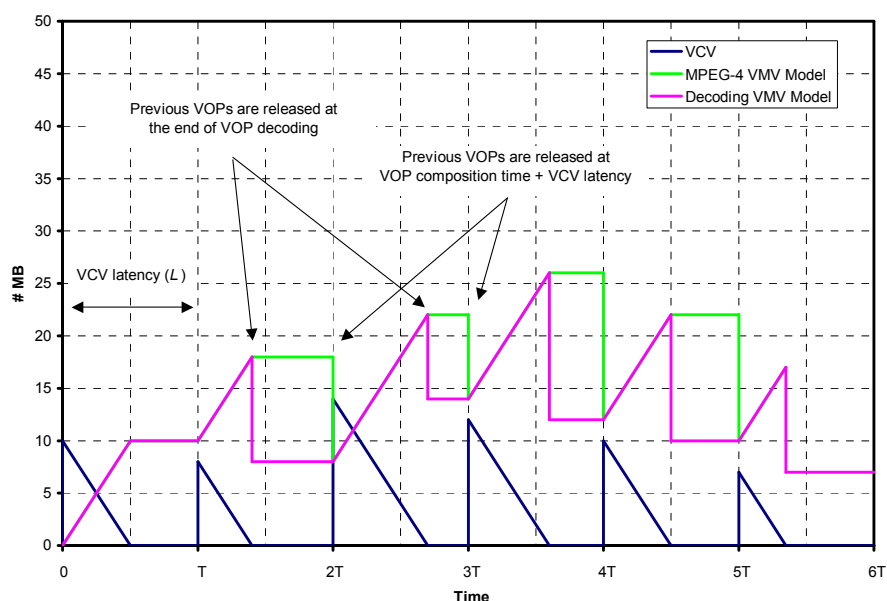


Figure 4.8 – VMV occupancy for the MPEG-4 and decoding memory approaches for a scene with 1 VO, when B-VOPs are not used

The longer memory occupancy of the MPEG-4 VMV model does not influence the encoding of single VO scenes since the maximum occupancy of the VMV buffer remains the same. However, when encoding scenes with multiple VOs, this solution may lead to an overestimation of the picture memory requirements and consequently to a possible waste of resources, resulting in some scenes being conservatively considered too demanding for a given profile and level. This situation is illustrated in Figure 4.9 showing an example with the occupancy of the VCV and VMV buffers for the MPEG-4 VMV model and the pure Decoding Memory approach considering the encoding of a generic multiple VO scene containing 2 VOs with VOP rates equal to $1/T$. For better understanding Figure 4.9, Table 4.2 shows the VOP sizes in MB units, for each VO.

As can be seen in Figure 4.9, the MPEG-4 VMV model requires more picture memory than it is really needed (if only the decoding process is taken into account), leading to an overestimation of the picture memory requirements. The earlier release of VOPs in the Decoding Memory approach has the advantage of allowing encoding more demanding scenes, in terms of picture memory, while keeping the profile and level limits untouched.

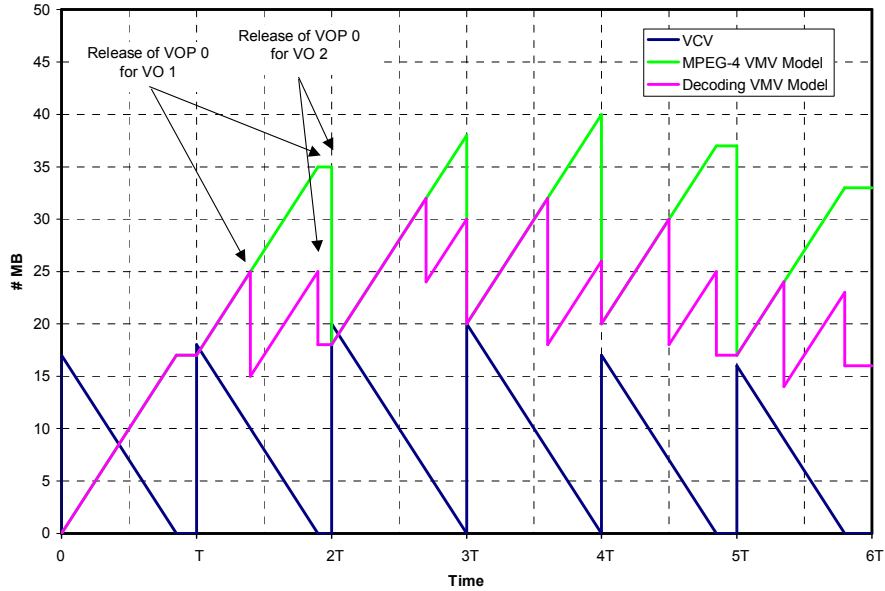


Figure 4.9 – VMV occupancy for the MPEG-4 and decoding memory approaches for a scene with 2 VOs, when B-VOPs are not used

Table 4.2 – VOP Size in MB units

VOP Index	VO 1	VO 2
0	10	7
1	8	10
2	14	6
3	12	8
4	10	7
5	7	9

II) Video Reference Memory Verification with B-VOPs

When B-VOPs are used, some of their prediction VOPs must be decoded in advance to their normal presentation order which in terms of allocated picture memory results in extra memory required to store them. In this case, composition has also to be delayed in order that the decoder is able to decode all VOPs before their composition time arrives. Figure 4.10 shows the relation between VOP decoding, composition, and release times for each of the VMV approaches under study, when using B-VOPs.

Figure 4.11 shows the occupancy of the VCV and VMV buffers for the encoding of a generic single VO with a VOP rate equal to $1/T$, using B-VOPs. For the same content, the Global Memory approach gives a higher occupancy of the VMV buffer while the Decoding Memory approach gives the lowest occupancy. As for the previous case (without B-VOPs), the MPEG-4 VMV model requires that extra memory stays allocated during some periods of time,

relatively to the Decoding Memory approach, because some VOPs are not released immediately after they become useless for the decoding process.

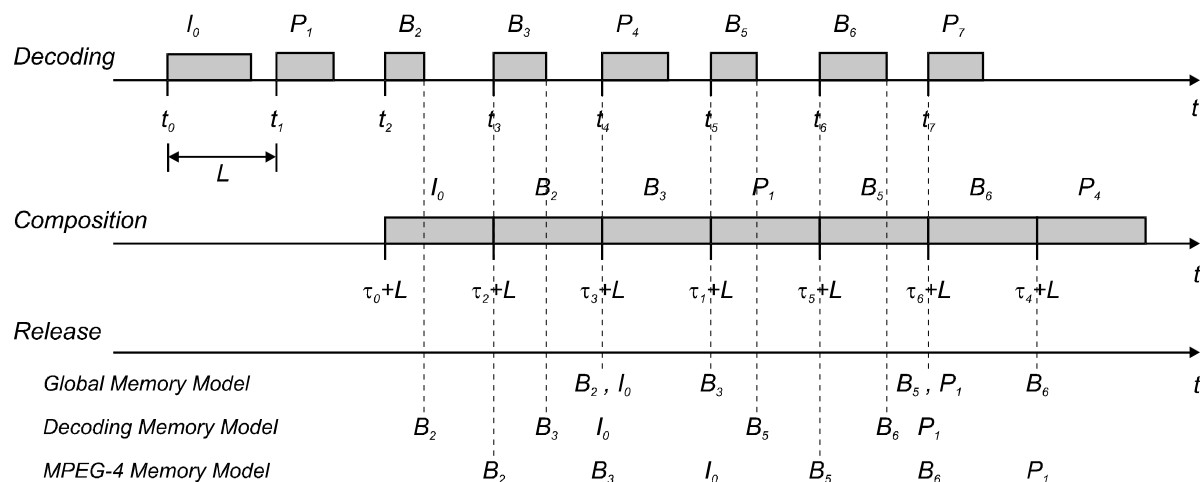


Figure 4.10 – Relation between the VOP decoding, composition, and release times for the VMV models under study when B-VOPs are used

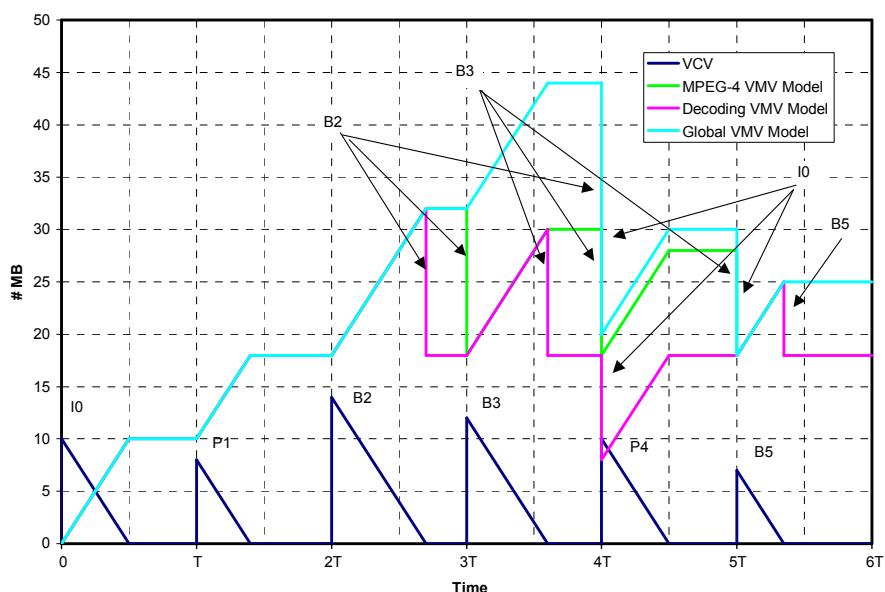


Figure 4.11 – VMV occupancy for the VMV models under study for a scene with 1 VO, when B-VOPs are used

The analysis presented above clearly shows that the adopted MPEG-4 VMV solution does not take into account all the picture memory needed at the decoder, notably the memory needed for the composition of B-VOPs, but overestimates the picture memory needed only for the decoding process. Since composition is not normative in MPEG-4, it may be seen as normal that the adopted MPEG-4 VMV solution does not follow the Global Memory approach; in this case, the Decoding Memory approach should have been precisely adopted to clearly reflect that situation.

4.5.3 VMV Encoder Implementation

This section describes the main steps of the VMV model implementation in the context of an MPEG-4 video encoder. For each VOL, the video encoder needs to allocate picture memory to store each VOP being encoded (decoded when the process will move to the decoder) and the VOPs used as predictions. Moreover, when it detects a possible violation of the VMV model, it has to take the adequate actions. For that, the encoder needs to keep track of the picture memory needed at the decoder side for storing the relevant decoded VOPs.

In order to properly schedule the encoding time instants for each VO, the precise memory allocation and de-allocation time instants at the decoder must be known by the encoder¹⁵ for each encoded VOP. The relation between decoding and composition times was described in Section 4.2.1.

For each target encoding time instant, the encoder verifies if the amount of picture memory required for decoding the given VOP or set of VOPs, during its lifetime at the decoder does not exceed the maximum amount of memory available for the selected profile and level. If a given VOP or set of VOP requires too much picture memory, the encoder has to take some action in order to keep valid the corresponding ES(s), e.g., it may skip the encoding of one or more VOPs for the current time instant. For this, the encoder has to estimate the picture memory usage during the decoding period of each VOP, which can be given by the following expression:

$$vmv(t) = vmv(s_i) + \min[(t - s_i) \cdot H, M_i] - \sum_{k \in A} M_k \cdot u(t - t_k), \quad s_i < t \leq e_i \quad (4.8)$$

where s_i and e_i are, respectively, the start and end decoding times of the given VOP, H is the VCV decoding rate, M_i is the total number of MBs in the VOP, $u(t)$ is the unit step function, A represents the set of VOPs to be released from memory in the interval $[s_i, e_i]$, and t_k their corresponding releasing time instants.

The first term in (4.8), $vmv(s_i)$, denotes the VMV occupancy immediately before starting decoding VOP i . The second term, $\min[(t - s_i) \cdot H, M_i]$, denotes the allocation of picture memory as the decoding process progresses (the VCV occupancy decreases at decoding rate H , while the VMV occupancy increases at the same rate). Finally, the picture memory released during the decoding period of the given VOP is expressed by $\sum_{k \in A} M_k \cdot u(t - t_k)$.

In order to ensure that the set of VOPs to be encoded at a given time instant does not exceed the profile and level picture memory limits, the encoder has to compute the local maximums for equation (4.8). This situation is exemplified in Figure 4.12 for the MPEG-4 VMV, Encoder Memory Estimation I, and for the Decoding Memory approach, Encoder Memory Estimation II, in the same conditions of Figure 4.9. The incoming VOPs require more picture memory than what is available if any of the determined local maxima exceeds the VMV buffer size.

Equation (4.8) plays a fundamental role in the VMV verification since it is very important that the encoder is able to estimate in advance the local maxima of the picture memory usage

¹⁵ The decoder may add some delay to the VOP decoding times provided that it accommodates the necessary additional buffering space.

process. By estimating in advance these local maxima, i.e., before encoding, the encoder is able to take preventive actions to avoid possible violations of the VMV.

Whenever the rate control mechanism detects a possible violation of the VMV model, it can take one of the following actions:

- Skip the encoding of one or more of the larger VOPs for that time instant.
- Avoid B-VOPs (when applicable).
- Use VOPs with less MBs in the bounding box, by merging/splitting VOs, or reducing its number (if this is an acceptable solution for the application in question; this action impacts on the “authoring” of the scene).
- Signal the impossibility to encode the given scene with the amount of picture memory provided by the chosen profile@level combination (this may lead to the choice of a more powerful profile@level in the case of off-line encoding).

An overflow of the VMV buffer may lead to a situation where the decoder is not able to correctly decode part or all MBs in one or more VOPs (due to the lack of picture memory to store the decoded VOPs) and thus to a coding desynchronization between encoder and decoder.

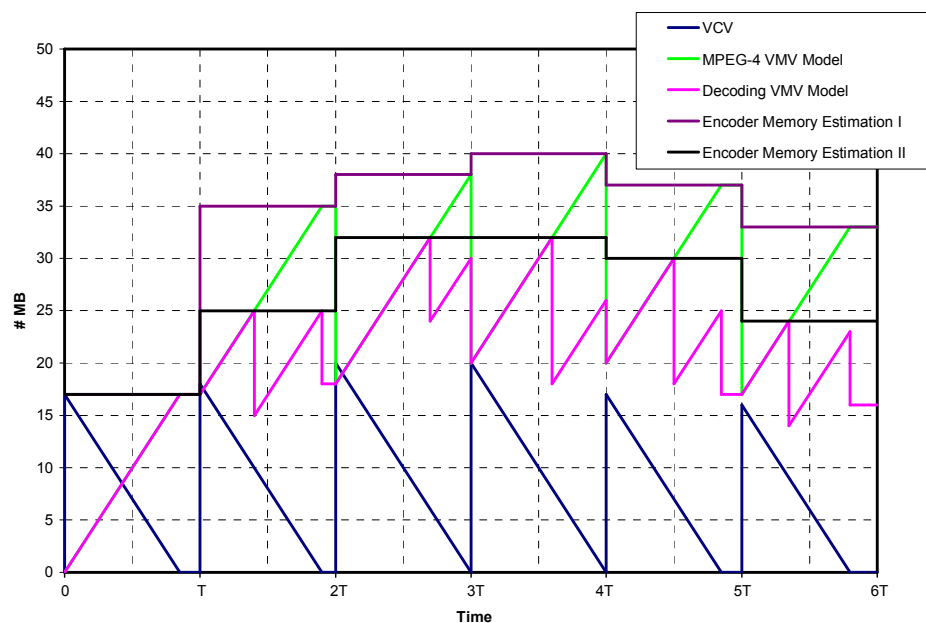


Figure 4.12 – Decoder memory allocation estimation for the MPEG-4 (I) and the decoding memory (II) VMV models

The video reference memory verifier mechanism plays a very important role in MPEG-4 video coding to ensure that the flexibility opened by the object-based approach does not prevent interoperability between different terminals at a reasonable cost, notably in terms of picture memory. This interoperability can only be achieved if the VMV model is not violated.

4.6 Analysis of the Video Complexity Verifier

The control of the computational power resources needed at the decoder is a major task of the

encoder. However there are several approaches for the specification and control of this type of decoding resources. The objective of this section is to highlight the major advantages and drawbacks of the MPEG-4 VCV approach in comparison with possible alternatives, both in terms of the encoded data complexity evaluation models and the virtual buffer models. Moreover some guidelines for the implementation of the VCV model within an MPEG-4 video encoder are proposed.

4.6.1 Encoded Data Complexity Modeling

The VCV model provides a way for the encoder to constrain the amount of computational power required at the decoder for decoding a set of visual elementary streams building a scene. In the MPEG-4 VCV model, the load on the decoder is measured by the occupancy of the VCV model buffers; the higher the occupancy of these buffers, the higher the computational load on the decoder.

In a first approach, the decoding complexity of the incoming data can be related to the amount of data that the decoder has to process, thus it can be related to the number of MBs per second that the decoder has to process. However, the computational power required to decode each MB may largely vary due to the diversity of MB types (e.g., in terms of shape: transparent, opaque, and boundary) and coding tools (e.g., in terms of texture coding tools: Intra, Inter, and Inter4V), associated to the various object types that can be used in the encoding process. Depending on the degree of precision that is required for the evaluation of the decoding complexity, more or less elaborate complexity evaluation measures can be used. However, the more close to the real complexity the model intends to go, the more difficult it is to be generic since the complexity becomes highly dependent on implementation issues which were intentionally left out of the normative scope of the standard.

In fact, there are several ways to measure the decoding complexity of the encoded data, ranging from simply measuring the MB rate to more complex measures related to the rate of machine instructions required to decode the data. These measures can be grouped into four categories [16], as illustrated in Figure 4.13:

- **Number of MBs** – In this approach the decoding complexity is simply modeled by the number of MBs per second in the VOs composing the scene. This is the straightforward approach; however it is rather weak since there are a large number of MB types that can be used with very different decoding complexities. As such this approach cannot effectively capture the real decoding complexity of the encoded scene.
- **Number of MBs per Shape Type** – This is a more realistic approach, already assuming some distinction between the different MBs in terms of decoding complexity, i.e., based on their shape information (opaque, boundary, or transparent). However even MBs with the same type of shape can exhibit very different decoding complexities, depending on the texture and shape coding tools used to encode them (e.g., for texture: Intra, Inter, etc.; for shape: IntraCAE, InterCAE, etc.). This type of approach has been adopted by MPEG-4 Visual [29] with the different MBs being discriminated into boundary and non-boundary MBs. In fact, in the MPEG-4 Visual standard two non-exclusive classes are defined: one considering only boundary MBs and another considering all MBs.
- **Number of MBs per Coding Type** – MPEG-4 Visual [29] defines a large number of coding tools that can be used to encode a given MB. Thus, an accurate decoding

complexity model shall take into account the decoding complexity of each coding tool, since this is the main source of differences between MBs, in terms of decoding complexity. In this case, the decoding complexity model can be based on the following types of tools:

1. **Texture Coding** – The MB decoding complexity is evaluated based on the texture coding tools used, e.g., Intra, Inter, or Inter4V.
 2. **Shape Coding** – The MB decoding complexity is evaluated based on the shape coding tools used, e.g., NoUpdate, IntraCAE, or InterCAE.
 3. **Texture and Shape Coding** – The MB decoding complexity is evaluated based on the combination of texture and shape coding tools used, e.g., Intra+IntraCAE, Inter+NoUpdate, Inter+InterCAE, or Inter4V+InterCAE.
- **Number of Decoding Instructions** – This is the most accurate approach since it closely takes into account the computational power required to decode a given scene. Its major drawback comes from the fact that it is highly dependent on the decoding implementation and platform used for the model definition, which makes the model valid only for similar decoding implementations.

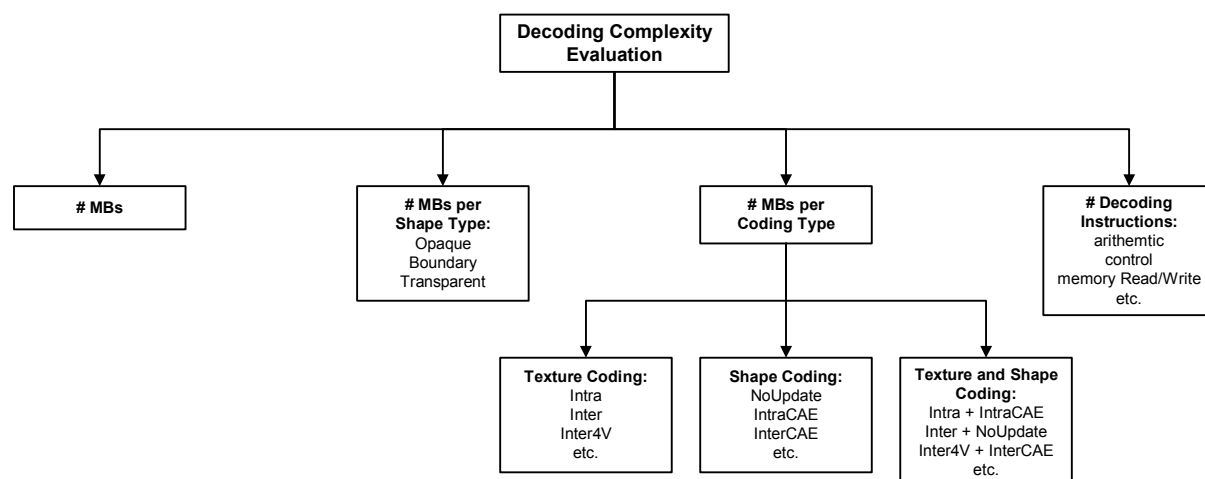


Figure 4.13 – Decoding complexity evaluation approaches for MPEG-4 video

In terms of the VCV model operation, the use of the measures proposed above is similar. At each VOP decoding time, the complexity measure for the incoming VOP is added to the VCV buffer occupancy (e.g., #MBs, #decoding instructions, etc.) which decreases at a constant rate defined in the appropriate data complexity units by the profile@level combination (e.g., #MB/s, #decoding instructions/s, etc.), until the next VOP decoding time.

The computational power constraints for a given profile@level are determined by the sizes of the VCV buffers and the draining rate of these buffers, using the appropriate complexity measure units. The size of the VCV buffers defines the maximum load that the decoder can accept, in terms of the complexity measures used, while the draining rates of the VCV buffers define the maximum decoding speed of the decoder.

The VCV model occupancy provides a way for the encoder to monitor the computational load of the decoder at each time instant. As such, it allows the encoder to bound the computational requirements on the decoder, since the encoder can use the occupancy of these buffers to

prevent that the complexity of the encoded scene exceeds the decoding speed capacity of the decoder during any period of time.

4.6.2 VCV Model Approaches

Monitoring the computational load of the decoder is an essential task of the encoder to guarantee that the selected computational boundaries are not overrun. However, there are several alternatives for the modeling of the decoder computational capability, notably depending on the number of buffers and decoding rates used. The following sections present several VCV model approaches based on virtual buffers and compare the adopted MPEG-4 VCV model with these alternative approaches.

SINGLE BUFFER WITH SINGLE DECODING RATE

The simplest VCV model using the virtual buffer approach consists in specifying a single VCV buffer that accumulates the complexity of the encoded data and a single decoding rate that defines the speed at which the decoder can decode this data. This approach assumes a first in first out (FIFO) decoding approach. There are two major possible variations for this approach:

I) Without MB Weights

In this approach, all MBs are considered to have the same weight in terms of decoding complexity, independently of the MB type and coding tools used. This approach has the advantage of being rather simple. Its major drawback comes from the fact that decoders have to be designed to deal with worst-case scenarios (heaviest MBs) in terms of decoding capacity since the decoding capacity required to decode a given set of ESs may have big fluctuations depending on the type of MBs and coding tools used.

If M_i is the number of MBs in a given VOP i and H is the VCV decoding rate, then the time that takes to decode a VOP i is given by

$$td_i = \frac{M_i}{H}$$

II) With MB Weights

In this approach, the VOP MBs are divided into classes, according to an adequate complexity criterion, e.g., MB coding type, and each MB class has an associated decoding complexity weight. The VCV occupancy due to a given VOP is a weighted sum of its MBs following the adopted complexity measure. As in the preceding case, the VCV buffer is emptied at a fixed VCV decoding rate. The main advantage of this approach is to model more closely the real decoding complexity of a given set of ESs building a visual scene. However, since decoders may be implemented in a variety of ways, from software only to general purpose or dedicated hardware, the definition of meaningful MB weights is not a straightforward task [142].

If M_{ij} is the number of MBs of VOP i in class j , and α_j is the corresponding decoding complexity weight, then the time that takes to decode VOP i is given by

$$td_i = \frac{1}{H} \left(\sum_{j=1}^N \alpha_j M_{ij} \right)$$

SINGLE BUFFER WITH MULTIPLE DECODING RATES

As for the single buffer with single decoding rate approach, this approach assumes a FIFO decoding solution; however, here, each MB class has its own decoding rate. A closer analysis shows that this approach is equivalent to the previous approach where the MB complexity weights are the ratios between the various MB class decoding rates and a reference decoding rate.

If H_j is the VCV decoding rate for class j , then the time that takes to decode VOP i is given by

$$td_i = \sum_{j=1}^N \frac{M_{ij}}{H_j}$$

or equivalently by

$$td_i = \frac{1}{H_k} \left(M_{ik} + \sum_{j=1, j \neq k}^N \frac{H_k}{H_j} M_{ij} \right) = \frac{1}{H} \left(\sum_{j=1}^N \alpha_j M_{ij} \right)$$

where $H = H_k$ is the reference rate and $\alpha_j = \begin{cases} 1 & \Leftarrow j = k \\ \frac{H_k}{H_j} & \Leftarrow j \neq k \end{cases}$.

MULTIPLE BUFFERS WITH MULTIPLE DECODING RATES

This VCV model approach assumes some degree of parallelism in the decoder that may happen in hardware-based decoders with dedicated hardware for decoding some specific types of MBs. An example would be a decoder containing a module for decoding MBs without shape information or completely opaque and another module for decoding MBs with shape information, i.e., boundary MBs.

In this case, the VCV model consists of N buffers, one for each MB class, and the associated decoding rates H_j . The decoding time for a given VOP i would be then given by

$$td_i = \max_j \left[\frac{M_{ij}}{H_j} \right]$$

In this approach, the decoding time is determined by the last buffer to be emptied.

MPEG-4 VCV MODEL APPROACH

The MPEG-4 VCV model approach follows a multiple buffers, multiple decoding rates approach considering two buffers: the VCV, accumulating all MBs of the incoming VOPs, and the B-VCV, accumulating only boundary MBs. Notice, however, that these two MB classes are not mutually exclusive since boundary MBs are counted both in the VCV and in the B-VCV.

With this approach, the MPEG-4 VCV model tries to include in the same model the parallel nature of a pure multiple buffers with multiple decoding rates with the serial nature of a single buffer with multiple decoding rates. Measuring the MB complexity based only on its shape type (in this case, boundary versus all) reflects a trade-off between the simplicity of the model

and its efficacy in characterizing the complexity of encoded scenes. However, since the MB coding types are not taken into account, this approach requires that decoders be designed to support the highest possible complexity in terms of MB coding types, denoting a worst-case scenario. This means that the decoder must be able to decode the number of MBs specified by the relevant profile@level combination, with all these MBs having the highest decoding complexity possible, with an additional restricting imposed by the B-VCV that the number of boundary MBs be less than 50% of the total for each VOP.

For the profiles and levels currently defined, the VCV has twice the decoding speed of the B-VCV reflecting the higher decoding complexity of boundary MBs [29]. It is important to refer however that, since the MBs are organized in two classes (boundary and all), opaque and transparent MBs are treated exactly in the same way. As it will be shown below, this can have a high (negative) impact in terms of the measured complexity of certain coded scenes.

Notice, that for the profiles not allowing arbitrarily shaped VOs, the MPEG-4 VCV becomes a single buffer single decoding rate model, i.e., there is no distinction between the MBs regarding their decoding complexity. For the profiles allowing arbitrary shapes, the additional B-VCV buffer introduces further constraints in the encoded data, notably by limiting the amount of encoded MBs with shape information that can be decoded with a given profile@level device.

VOP Decoding Time Duration

The fact that for the current MPEG-4 Visual profiles@levels the VCV decoding rate is twice the B-VCV decoding rate (i.e., $H = 2 \cdot H_B$) determines that the decoding time duration of a given VOP is limited by the VCV if the ratio between the number of MBs with shape information and the total number of MBs in the VOP is less than 50%, and by the B-VCV otherwise. This means that the decoding time duration, td_i , for a given VOP i is given by

$$td_i = \begin{cases} \frac{M_{Bi}}{H_B} \Leftarrow M_{Bi} \geq \frac{1}{2} M_i \\ \frac{M_i}{H} \Leftarrow M_{Bi} < \frac{1}{2} M_i \end{cases}$$

where M_{Bi} represents the amount of MBs with shape information and M_i the total number of MBs in VOP i .

An important singularity of the MPEG-4 VCV model that should be taken into account when producing MPEG-4 content is the fact that the same amount of MBs in a scene can lead to different decoding times, depending on how they are arranged in terms of VOs. For example, a single VOP with 198 MBs (99 non-boundary + 99 boundary) would take approximately 33 ms to decode for the Core Profile @ Level 1 (see decoding rates in Table 2.6), i.e.,

$$td = \max \left[\frac{198}{5940}, \frac{99}{2970} \right] \approx 33 \text{ ms},$$

while the same amount of MBs taken as two VOPs (VOP1 = 69 non-boundary + 30 boundary, and VOP2 = 30 non-boundary + 69 boundary) belonging to two different VOs, would take approximately 40 ms to decode, for the same profile and level, i.e.,

$$td = \max \left[\frac{99}{5940}, \frac{30}{2970} \right] + \max \left[\frac{99}{5940}, \frac{69}{2970} \right] \approx 40 \text{ ms}$$

This is a direct consequence of equation (4.3) which imposes that decoding of a given VOP can only start after the previous VOP in the VCV has been completely removed from the two buffers, even if one of the VCV buffers became empty meanwhile¹⁶.

MB Decoding Complexity

Another important aspect deserving attention is the impact of certain types of MBs, notably transparent and skipped MBs, in terms of the scene complexity evaluation, this means in terms of the VCV buffer occupancy.

The fact that transparent MBs are not distinguished from opaque and boundary MBs, in terms of the VCV model¹⁷, although they carry no texture information, may lead to an overestimation of the scene complexity and thus to the impossibility of compliantly encode scenes which were expected to be encoded with a certain profile@level combination (in comparison with similar scenes in terms of complexity which can be encoded with that profile@level). This situation will be illustrated below in Figure 4.17 to Figure 4.20, where it is clear that peaks or high percentages of transparent MBs result usually in violations of the VCV model.

This situation can be particularly critical for objects mainly composed by transparent MBs, which although containing only a few MBs with texture data, are considered of high complexity in terms of the VCV model (see Figure 4.14), or in the case of scenes composed by several VOs with overlapping bounding boxes, where certain MBs overlap in the scene and thus contribute more than once (even as transparent) to the VCV and VMV buffer occupancies (see Figure 4.15). Notice that the computational load of a transparent MB is mainly due to the extended padding process, which consists in filling the MB according to the following procedure (see Clause 7.6.1.3 of [29]):

- Transparent MBs near boundary or opaque MBs are filled by replicating vertically or horizontally the samples at the border of the neighboring MB¹⁸.
- The remaining transparent MBs, having only other transparent MBs as neighbors, are filled with the value $2^{bits_per_pixel-1}$.

The computational load of this process is relatively low when compared to other MB coding types, e.g., where the IDCT has to be performed; all these MBs count in the same way for the MPEG-4 VCV model.

The impact of transparent MBs in terms of the video buffering verifier mechanism is more critical in terms of the VCV than the VMV model given the more strict limits imposed by the Profiles and Levels definitions in Annex N of MPEG-4 Visual [29] for the VCV model. In terms of the VMV model, this impact is not so critical, although some decoder implementations could use more efficient ways for storing transparent MBs, notably those padded with constant values.

¹⁶ The standard is not clear regarding the decoding of two VOPs with the same decoding time stamp, however it is not possible to assume that the two VOs can be decoded in parallel because this would require a higher decoding capacity.

¹⁷ In the MPEG-4 VCV model, when the number of boundary MBs in a VOP is less than 50% all MBs in the VOP have in the same weight in terms of the VOP decoding complexity.

¹⁸ When the transparent MB has more than one boundary or opaque neighbor, a pre-defined rule is used for selecting this neighbor.



Figure 4.14 – Container sequence: object mainly composed by transparent MBs (82 %)

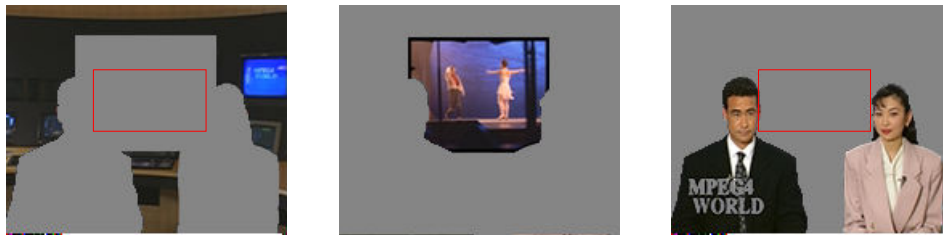


Figure 4.15 – News sequence: MBs inside the rectangles are counted three times (twice as transparent) for the VCV and VMV buffers

As shown in [21] the decoding complexity of each MB, measured in terms of the MB decoding time, varies significantly according to the MB coding type and not only according to the boundary and non-boundary distinction as assumed by the MPEG-4 VCV [29]. Consequently, the work presented in [21] proposes, after intensive decoding time measurements, MB complexity weights, which model more effectively the decoding complexity of a MPEG-4 video coded object.

Taking into account that the MPEG-4 VCV is implicitly designed for the most complex MB type (worst case scenario), the complexity weights proposed in [21] are defined relatively to the most complex MB type in the context of each profile, i.e., the maximum complexity weight is set to 1 for this MB type and all the other weights are relative to this one and thus have a complexity weight lower than 1. To simplify the alternative VCV models, some MB types with similar decoding complexity are grouped in just one complexity class as shown in Table 4.3 (following a conservative approach, the relative complexity weight attributed to each class is the weight of the most complex MB type included in that class).

As can be seen from Table 4.3, the relative complexity weights are significantly lower for transparent and skipped MBs than for other types of MBs. This set of relative complexity weights allows the implementation of a “trading system”, where it is possible, for example, to trade one of the most complex MBs by two MBs with half the relative complexity, while still maintaining the bitstream decodable by a compliant decoder, this means without having to require higher decoding resources.

Table 4.3 – MB decoding complexity classes and relative complexity weights [21]

MB Class	MB Coding Type	Relative Weight	
		Simple Profile	Core Profile
C ₁	Inter4V+InterCAE Inter+InterCAE Inter4V+IntraCAE	–	1.00
C ₂	Inter+IntraCAE Intra+IntraCAE	–	0.88
C ₃	Inter4V+NoUpdate Inter+NoUpdate Intra+NoUpdate	–	0.77
C ₄	Inter4V+Opaque Inter+Opaque Intra+Opaque	–	0.70
C ₅	Skipped+InterCAE	–	0.40
C ₆	Skipped+IntraCAE	–	0.32
C ₇	Skipped+NoUpdate	–	0.21
C ₈	Skipped+Opaque	–	0.12
C ₉	Transparent	–	0.12
C ₁₀	Inter4V (only rect. VOs)	1.00	0.66
C ₁₁	Inter (only rect. VOs) Intra (only rect. VOs)	0.89	0.59
C ₁₂	Skipped (only rect. VOs)	0.13	0.09

MPEG-4 VCV OCCUPANCY FOR TYPICAL TEST SEQUENCES

In the following, some results regarding the implementation of the MPEG-4 video buffering verifier mechanism for the MPEG-4 version 1 Visual Core Profile are presented with the purpose to highlight some characteristics of the MPEG-4 video buffering verifier mechanism. Notably some statistics for the VCV and VMV models obtained while encoding some MPEG-4 test sequences with the MPEG-4 Core Profile for Levels 1 and 2 (CP@L1 and CP@L2) are presented [19]. The plot of the VMV occupancy is also shown to illustrate how the two models perform together in critical conditions.

Figure 4.16 to Figure 4.20 present the statistics of the different types of MBs (transparent, opaque, and boundary) and the VCV and VMV buffer occupancies for the test sequences listed in Table 4.4. All sequences have been coded at 15 Hz, without B-VOPs. In this case, the feedback mechanism that prevents the violation of the VCV and VMV models has been disabled, which means that whenever the encoder detects a violation of these video buffering verifier models, the set of bitstreams is signaled as non-compliant. This was done on purpose to highlight some critical situations for each of the models. Table 4.5 presents the VMV and VCV buffer sizes and decoding rates for the profile@levels used.

Table 4.4 – Test sequences used for each Profile@Level

Core Profile @ Level 1 (CP@L1)	Core Profile @ Level 2 (CP@L2)
Stefan (2 VOs) at QCIF	Stefan (2 VOs) at CIF
Children (3 VOs) at QCIF	Children (3 VOs) at CIF
Coastguard (3 VOs) at QCIF	Coastguard (3 VOs) at CIF
News QCIF (4 VOs) at QCIF	News (4 VOs) at CIF
	Container (6 VOs) ¹⁹ at QCIF
	Container (6 VOs) at CIF

Table 4.5 – VMV and VCV buffer sizes and decoding rates for the profile@levels used [19]

Profile @ Level	Maximum number of objects	VMV buffer size (MB)	VCV buffer size (MB)	VCV decoding rate (MB/s)	Boundary VCV decoding rate (MB/s)
Core Profile @ Level 2	16	2376	792	23760	11880
Core Profile @ Level 1	4	594	198	5940	2970

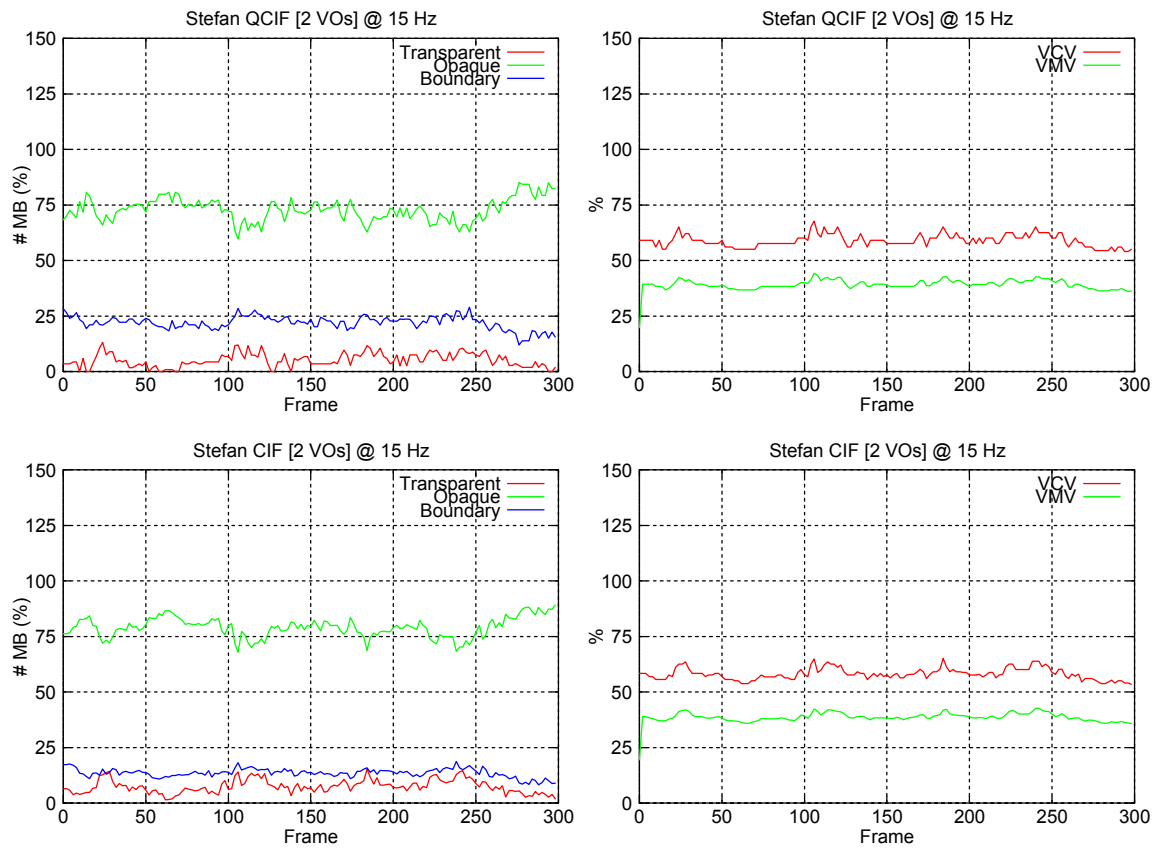


Figure 4.16 – VCV and VMV occupancies for Stefan: (top) CP@L1; (bottom) CP@L2

¹⁹ This scene cannot be coded in CP@L1 since it exceeds the maximum number of VOs for that level.

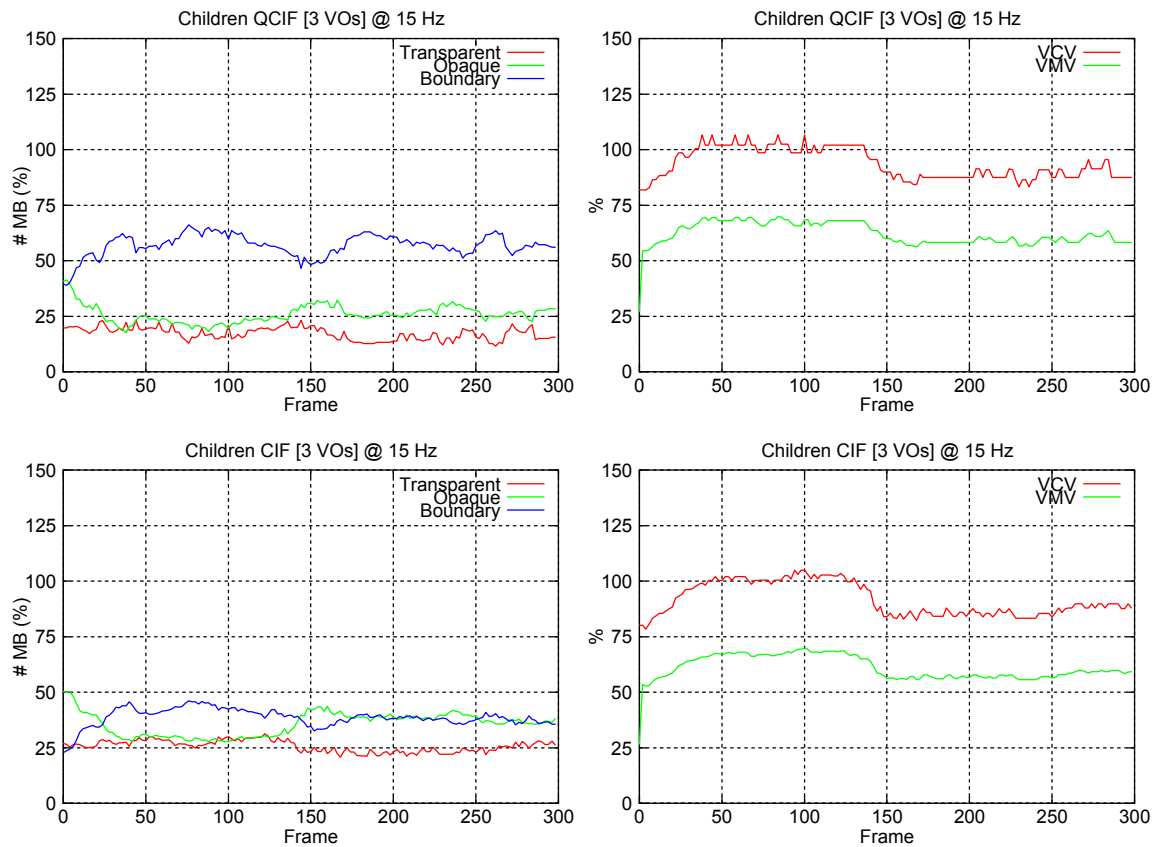


Figure 4.17 – VCV and VMV occupancies for Children: (top) CP@L1; (bottom) CP@L2

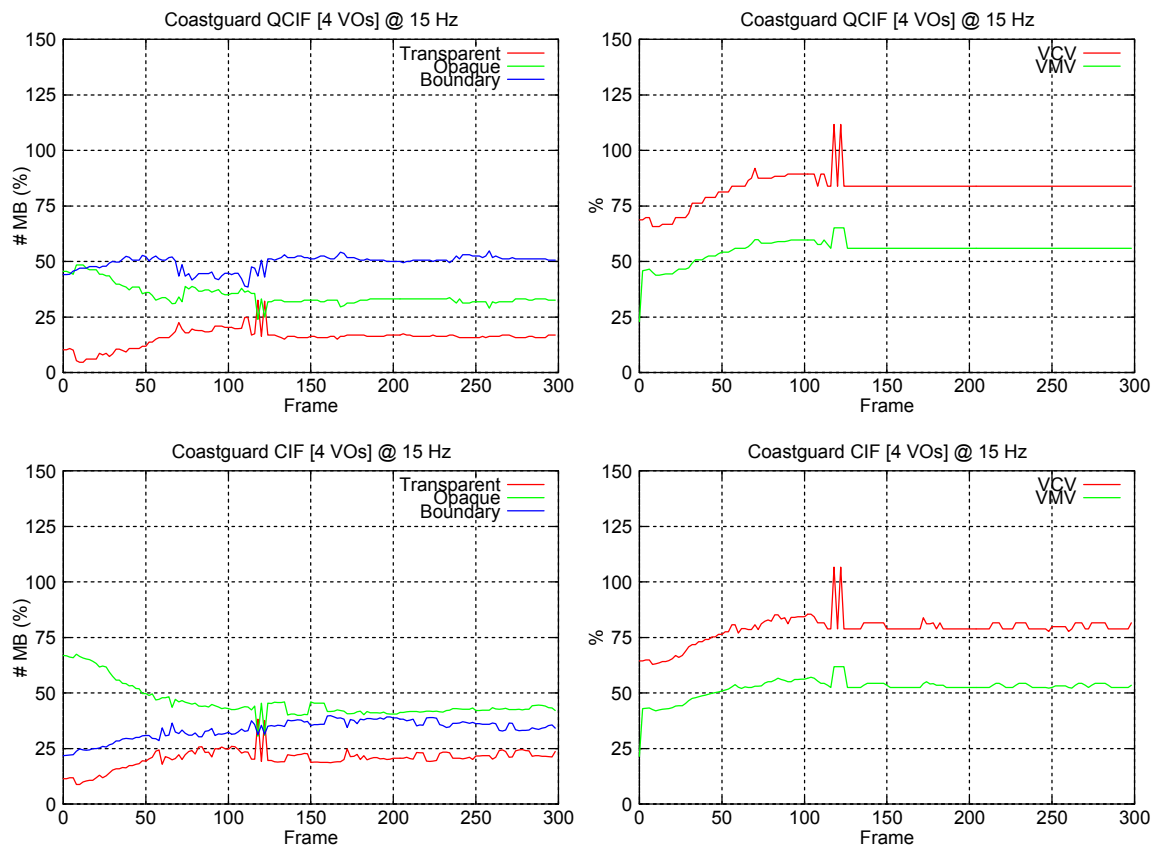


Figure 4.18 – VCV and VMV occupancies for Coastguard: (top) CP@L1; (bottom) CP@L2

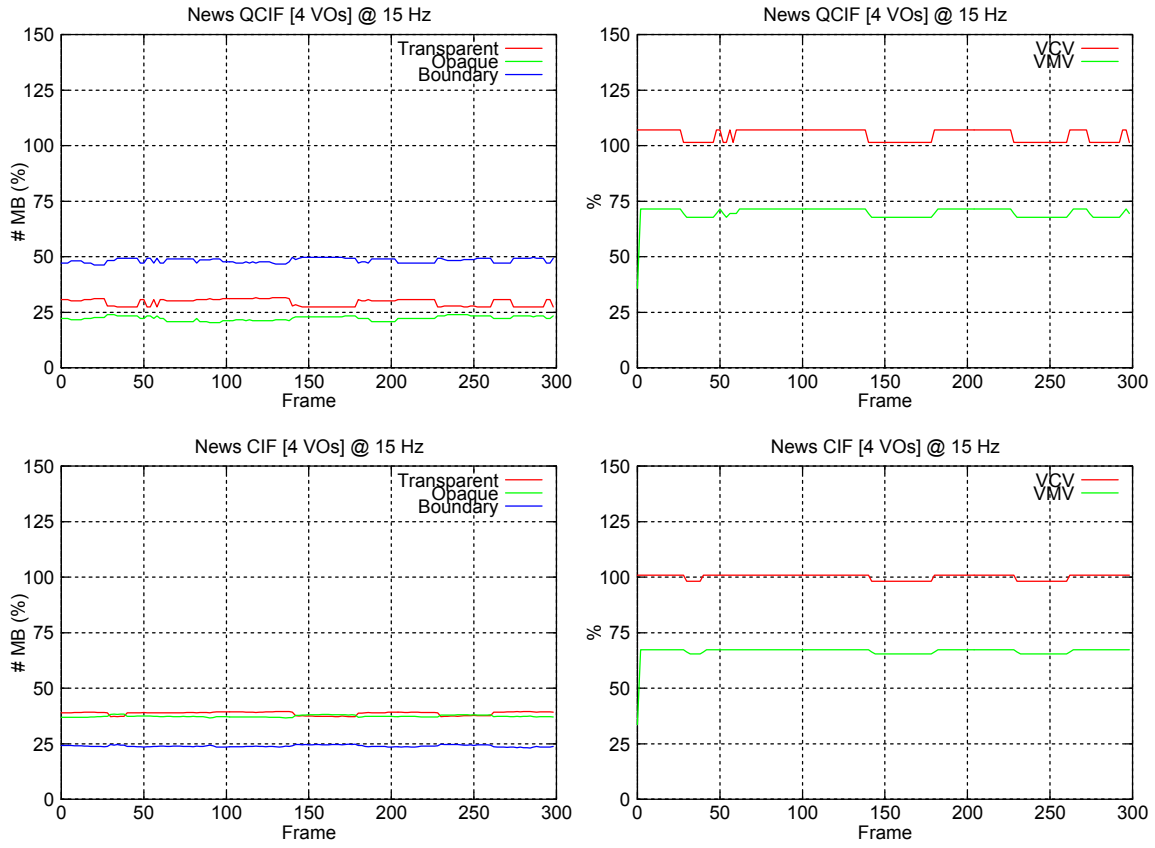


Figure 4.19 – VCV and VMV occupancies for News: (top) CP@L1; (bottom) CP@L2

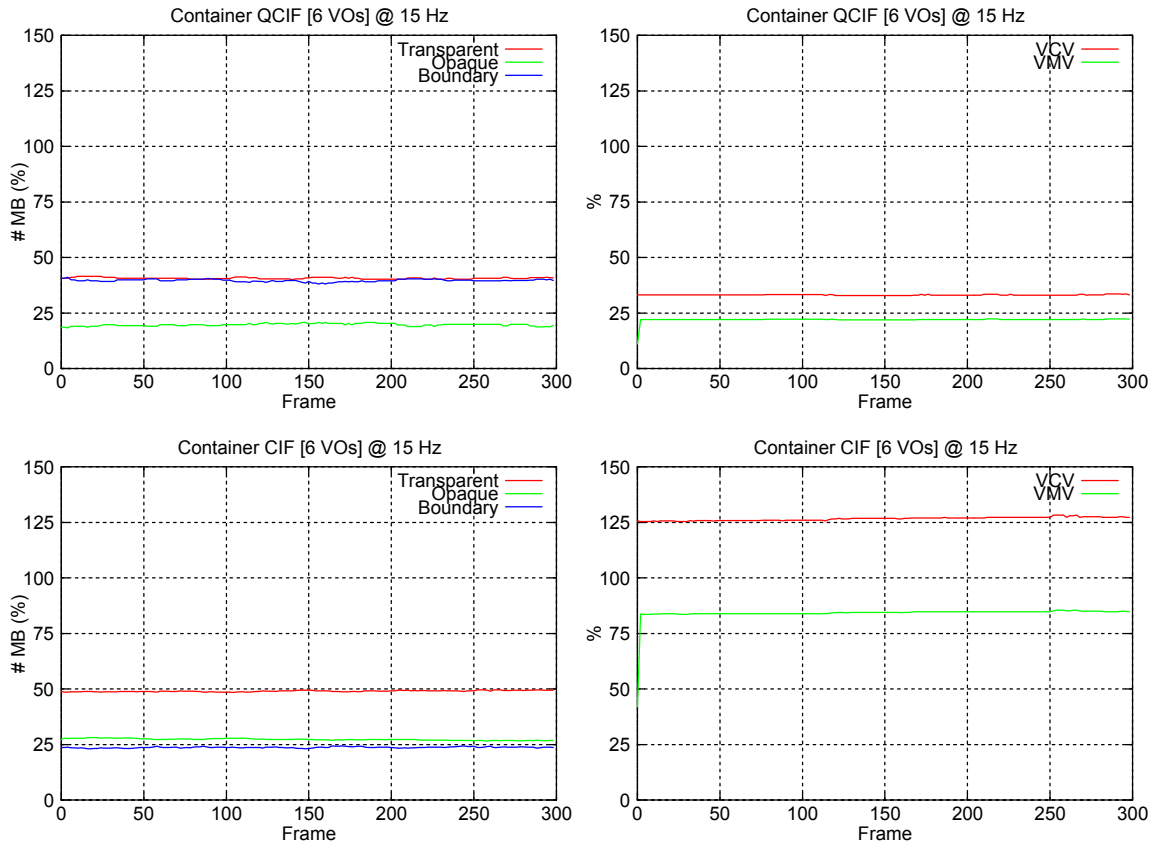


Figure 4.20 – VCV and VMV occupancies for Container: CP@L2

Although CP@L1 is typically targeted to encode QCIF scenes with up to 4 objects, the results presented above show that this is not possible for typical MPEG-4 test sequences like Children, Coastguard, and News since the VCV buffer model overflows. This fact brings some doubts about the adequacy of the MPEG-4 VCV specification for the considered profile@level. In the case of the Coastguard sequence, the VCV overflow is due to a vertical panning in object 3 (the riverside) between frames 118 and 122, which abruptly increases the size of its bounding box and consequently increases the number of transparent MBs during a few frames, leading to an overflow of the VCV buffer (see Figure 4.18).

The same behavior is experienced in the case of CP@L2, typically targeted to encode CIF scenes with up to 16 objects. Even with just 3 or 4 objects, as in the case of Children and News, the VCV buffer overflows. As can be seen in Figure 4.20 (bottom), the Container sequence with 6 objects clearly overflows the VCV buffer mainly due to the large percentage of transparent MBs in the scene when encoded in CIF format with CP@L2.

The results presented in this section show that several scenes typically targeted to be encoded with either CP@L1 or CP@L2 clearly exceed the VCV buffer size limits imposed by the Profile and Levels definitions. This fact prevents these scenes to be compliantly encoded with these profile@level combinations. This results in a limitation of the specification, mainly due to the excessive and inadequate weight that transparent MBs have in the MPEG-4 VCV model

IST VCV MODEL APPROACH

A VCV model where the three different types of MBs would be discriminated in terms of decoding complexity would be more adequate than the current MPEG-4 VCV model. In this context, the Single Buffer with Single Decoding Rate with MB weights or the Multiple Buffers with Multiple Decoding Rates solutions appear as good candidates for the VCV modeling. In the first case, the impact of transparent MBs could be reduced by assigning them a lower MB weight, while for the second case this could be done by defining an higher buffer size and higher decoding rate for transparent MBs. Notice, however, that, as shown in [21], the real MB decoding complexity is directly related to the MB texture and shape coding tools used. Thus, a more accurate VCV model must take this fact into consideration.

This approach has been followed in [22], where an alternative video complexity verifier model – IST VCV model – based on the set of relative MB complexity weights presented in Table 4.3 has been proposed for the Simple and Core Profiles. The major characteristics of the IST VCV model are:

- **Complexity model based on the MB coding tools** – The distinction in terms of decoding complexity between the various MBs is associated to the different MB texture and shape coding tools used, i.e., the MB complexity classes are related to a texture-shape coding tools combination for which a relative complexity weight is measured.
- **Single buffer with relative MB complexity weights** – In the proposed model, a single buffer stores all the encoded MBs, but each MB is weighted according to its complexity class. Thus, the IST VCV buffer occupancy corresponds to a weighted sum of the encoded MBs. The IST and MPEG-4 VCV buffer sizes are made the same, making possible to compare the two models in a simple way, since the decoding computational resources remain the same.
- **Single decoding rate** – The use of a single buffer with MB complexity weights implies a single decoding rate. The IST and MPEG-4 VCV decoding rates are made

the same, making possible to compare the two models in a simple way, since the decoding computational resources remain the same.

The main advantage of the IST VCV solution, relatively to the MPEG-4 VCV solution, is to model more closely the real decoding complexity of a given set of bitstreams building a video scene, since the different types of MBs are distinguished in terms of decoding complexity and thus decoding resources are not wasted due to the “killing” assumption that all MBs beside boundary MBs are equally and maximally difficult.

For the IST VCV model, the number of equivalent (to the most complex) MBs for a given VOP i , M_i , that is added to the VCV buffer at each decoding time instant, t_i , is given by the following expression

$$M_i = \sum_{j=1}^N \alpha_j M_{ij} \quad (4.9)$$

where α_j is the relative decoding complexity weight associated to the MB complexity class j , M_{ij} is the number of MBs in VOP i belonging to the complexity class j , and N is the number of complexity classes: 3 for the Simple Profile and 12 for the Core Profile²⁰ (see Table 4.3). In this context, the time that takes to decode VOP i , td_i , is then given by the following expression

$$td_i = \frac{M_i}{H}$$

where M_i is the equivalent number of MBs in VOP i , given by (4.9), and H is the VCV decoding rate for the profile@level in question. The interval of time where VOP i is being decoded extends from the time instant s_i to the time instant e_i , which are defined by the following expressions

$$s_i = t_i + \frac{vcv(t_i)}{H}, \quad e_i = t_i + \frac{vcv(t_i) + M_i}{H}$$

where t_i is again the VOP i decoding time, $vcv(t_i)$ is the VCV occupancy when the VOP i MBs, M_i , are added to the VCV and H is the VCV decoding rate for the profile@level in question. The new VCV model assumes a full sharing of the available decoding resources, which is in principle valid, at least for decoder software implementations.

The adoption by MPEG-4 of this alternative VCV model proposed by IST (positively considered by MPEG) would imply the change of the MPEG-4 video decoding complexity model, removing the B-VCV buffer while keeping the VCV buffer with the same parameters; moreover the VCV filling model would change from a simple addition of the number of MBs to an weighted addition of the number of MBs, using as weights the relative complexity weights presented in Table 4.3.

Since the IST VCV decoding rate and buffer size are unchanged relatively to the MPEG-4 VCV model for profiles not supporting scenes with arbitrarily shaped objects, a direct comparison between the two models can easily be done, because the decoder computational

²⁰ In this model B-VOPs were not considered.

resources are maintained. This comparison is illustrated in Figure 4.21 for two MPEG-4 test sequences encoded with the Simple Profile:

1. Akiyo in QCIF format encoded at 15 Hz with Simple Profile @ Level 1 at 64 Kbps
2. Stefan in CIF format encoded at 30 Hz with Simple Profile @ Level 3 at 384 Kbps

As can be seen in Figure 4.21, for both sequences, the MPEG-4 VCV occupancy is always 100%, which makes the encoded bitstreams MPEG-4 compliant for the considered profile@levels. With the IST VCV model, these two sequences can also be compliantly encoded with the same profile@level, but the VCV occupancies are lower, varying between 25% and 50% for the Akiyo sequence and being around 75% for the Stefan sequence, during the whole encoding process. The term “compliant” means here that maintaining the standardized decoding resources for a certain profile@level, the bitstream would be decodable, fulfilling the necessary time constraints since it is not really more complex than other MPEG-4 “officially” compliant bitstreams (for the relevant profile@level).

The fact that the VCV occupancy is always 100% for the MPEG-4 model is a direct result of both sequences being encoded at the maximum VOP rate allowed with this model for the spatial format and the profile@level combinations used, i.e., 15 VOP/s = (1485 MB/s) / (99 MB/VOP) for QCIF SP@L1; and 30 VOP/s = (11880 MB/s) / (396 MB/VOP) for CIF SP@L3. Table 4.6 shows the relation between the VCV decoder rate and VCV buffer size for the profiles under consideration.

Table 4.6 – Relation between the VCV decoder rate and VCV buffer size for the Simple Profile @ Level 1 and Simple Profile @ Level 3

Profile@Level	VCV buffer size (MB)	VCV decoder rate (MB/s)	VCV decoder rate / VCV buffer size (Hz)
Simple Profile @ Level 1	99	1485	15
Simple Profile @ Level 3	396	11880	30

With the IST VCV model it would be possible to encode the Akiyo sequence at a higher VOP rate, exploring the fact some MB types, e.g., Skipped, are in reality much less complex than the most complex MB coding type for the Simple Profile (Inter4V) according to Table 4.3. This would allow, for example, to selectively improve the subjective quality of some parts the scene, e.g., with strong eye and mouth movements, by encoding these parts more frequently with more complex MB types (e.g., Intra, Inter, or Inter4V) and encoding the background with less complex MB types (e.g., Skipped).

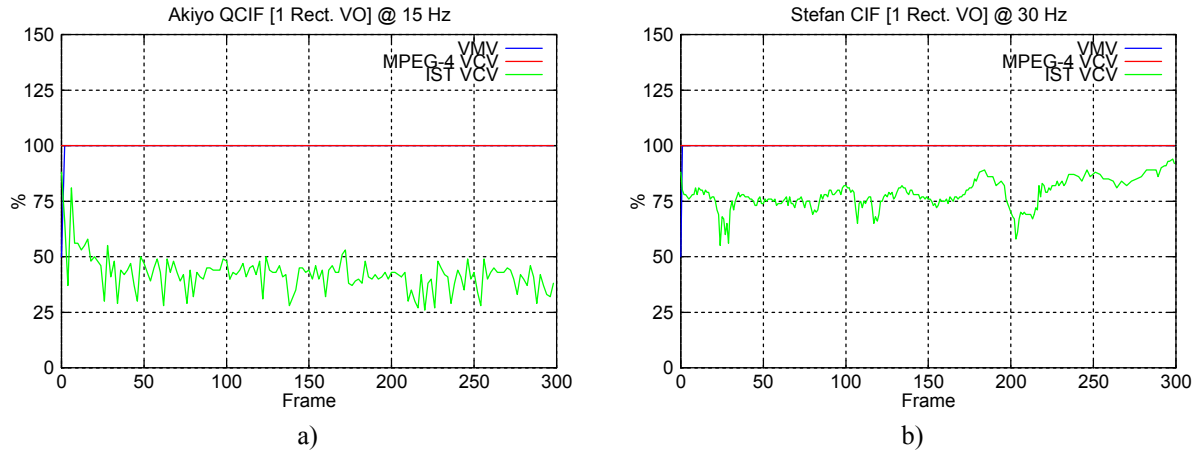


Figure 4.21 – MPEG-4 and IST VCV and VMV occupancies for the Simple Profile:
 a) Akiyo SP@L1 at 64 kbps; b) Stefan SP@L3 at 384 kbps

Notice, that in order to make a rigorous comparison between the MPEG-4 VCV and the IST VCV for profiles supporting scenes with arbitrarily shaped objects, the restriction imposed by the MPEG-4 B-VCV requiring that the number of boundary MBs for each decoding time is not greater than half of the B-VCV capacity must be considered. This means that, from a complexity point of view, the MPEG-4 VCV worst-case scenario corresponds to the case where the MBs are 50% of the most complex non-boundary MB type (“Inter4V + Opaque”) and 50% of the most complex boundary MB type (“Inter4V + InterCAE”).

To accommodate this case, and only for the purpose of the comparison between the two VCV models, the relative complexity weights have to be changed using as reference the average complexities of the “Inter4V + InterCAE” and the “Inter4V + Opaque” MB types, and not the “Inter4V + InterCAE” MB type complexity alone, as proposed in the IST VCV model, which should be used if the MPEG-4 B-VCV did not exist.

In this circumstance, the “trading system” is not simply referred to the “Inter4V + InterCAE” MB type, because the decoder does not have to support the case where the MBs are all of this type, but has to be referred to the average complexity between the most complex arbitrary shape MB type and the most complex non-arbitrary shape MB type. In this situation, it is natural that the complexity cost of an “Inter4V + InterCAE” MB is higher than 1, since this type is more complex than the reference complexity.

Considering all these facts, to compare the MPEG-4 and the IST VCV models, new relative complexity weights have to be computed. The new relative complexity weights for the various possible MB complexity classes presented above and for profiles supporting scenes with arbitrarily shaped objects (notably the Core Profile) are presented in Table 4.7 [142].

Table 4.7 – Comparison between the MPEG-4 and IST VCV models: MB decoding complexity classes and relative complexity weights for the Core Profile

MB Class	MB Coding Type	Relative Weight
C ₁	Inter4V+InterCAE Inter+InterCAE Inter4V+IntraCAE	1.17
C ₂	Inter+IntraCAE Intra+IntraCAE	1.03
C ₃	Inter4V+NoUpdate Inter+NoUpdate Intra+NoUpdate	0.89
C ₄	Inter4V+Opaque Inter+Opaque Intra+Opaque	0.83
C ₅	Skipped+InterCAE	0.47
C ₆	Skipped+IntraCAE	0.37
C ₇	Skipped+NoUpdate	0.25
C ₈	Skipped+Opaque	0.14
C ₉	Transparent	0.14
C ₁₀	Inter4V (only rect. VOs)	0.78
C ₁₁	Inter (only rect. VOs) Intra (only rect. VOs)	0.70
C ₁₂	Skipped (only rect. VOs)	0.10

Figure 4.22 shows the MPEG-4 and IST VCV occupancies for two MPEG-4 test sequences in three different encoding situations:

1. News in QCIF format encoded at 15 Hz with Core Profile @ Level 1 at 384 kbps.
2. Coastguard in QCIF format encode at 30 Hz with Core Profile @ Level 1 at 384 kbps.
3. News in CIF format encoded at 30 Hz with Core Profile @ Level 2 at 2000 kbps.

As can be seen in this figure, in all three situations the MPEG-4 VCV buffer overflows, and thus the bitstreams are not compliant. The IST VCV model, however, is always below 100% indicating that the scene complexity in all three cases is below the maximum complexity allowed for the profile@level selected.

To perform this comparison an encoder with only the VBV rate control active was used. The feedback mechanism that prevents the violation of the VCV and the VMV models has been disabled in order to allow the visualization of the corresponding buffer occupancy evolution even if it is above 100% occupancy. In this case, the encoded bitstreams are the same for both models since there is no feedback control, but the VCV fullness computation is done differently, depending on the considered model.

In Figure 4.22a the MPEG-4 VCV is always above 100%, although the scene is encoded at a VOP rate inferior to the relation between the VCV decoder rate and the VCV buffer size (see Table 4.8), since the number of MBs added to the VCV occupancy for each encoding time instant is higher than the VCV buffer size. The IST VCV model, however, is always only slightly above 50% indicating that many of these MBs are of low complexity (e.g., Transparent and Skipped), and thus that the given scene is not too demanding for the profile@level selected.

Figure 4.22b also shows that the scene considered in this case is not too complex to be encoded with the selected profile@level, since the IST VCV model occupancy is around 70% during most time of the encoding process. The MPEG-4 VCV occupancy peak that can be seen is caused by a great number of transparent MBs that appear in those VOPs. Since, in the IST VCV model, transparent MBs have a low relative computational weight, the peak is attenuated and the IST VCV buffer does not overflow.

Another example that shows the weakness of the MPEG-4 VCV model in the presence of transparent MBs and the effectiveness of IST VCV model is presented in Figure 4.22c. In this case the MPEG-4 VCV buffer capacity is largely exceeded during the encoding process. On the other hand, the IST VCV occupancy is always only around 50%, which shows that this scene can be encoded with the selected profile@level.

Table 4.8 – Relation between the VCV decoder rate and VCV buffer size for the Core Profile @ Level 1 and Core Profile @ Level 3

Profile@Level	VCV buffer size (MB)	VCV decoder rate (MB/s)	VCV decoder rate / VCV buffer size (Hz)
Core Profile @ Level 1	198	5940	30
Core Profile @ Level 2	792	23760	30

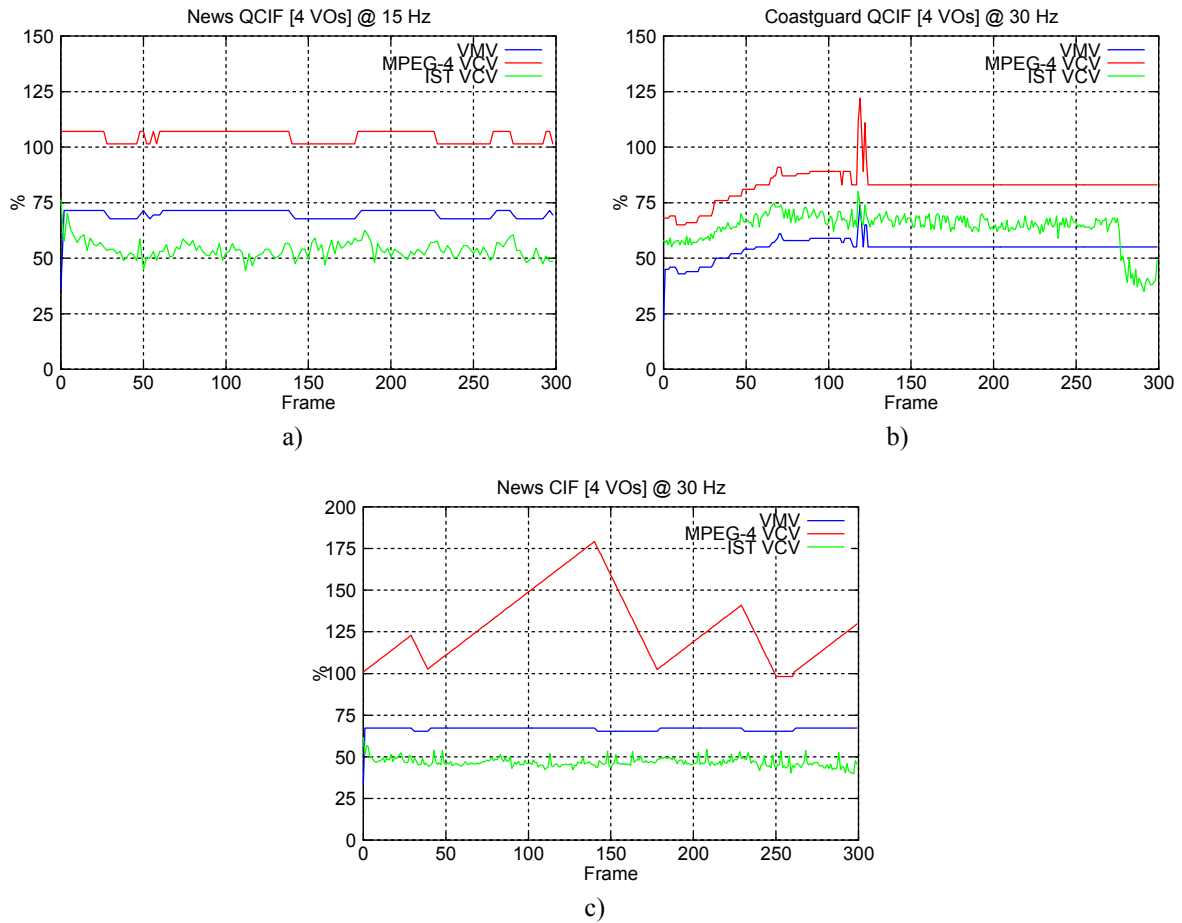


Figure 4.22 – MPEG-4 and IST VCV and VMV occupancies for: a) News CP@L1 at 384 kbps; b) Coastguard CP@L1 at 384 kbps; c) News CP@L2 at 2000 kbps

The influence of transparent MBs in the MPEG-4 VCV can be easily verified by looking to Figure 4.23, which shows the number of transparent, opaque, and boundary MBs for the News sequence in QCIF format. The number of boundary and opaque MBs stays approximately constant along the scene, while the number of transparent MBs oscillates between two (rather high and similar) values. As can be seen in Figure 4.22a and Figure 4.23, when the number of transparent MBs increases, there is a corresponding increase in the MPEG-4 VCV occupancy, and when the number of transparent MBs decreases, the MPEG-4 VCV occupancy decreases. On the other hand, the IST VCV buffer occupancy stays approximately constant, because of the low complexity weight that transparent MBs have in this model.

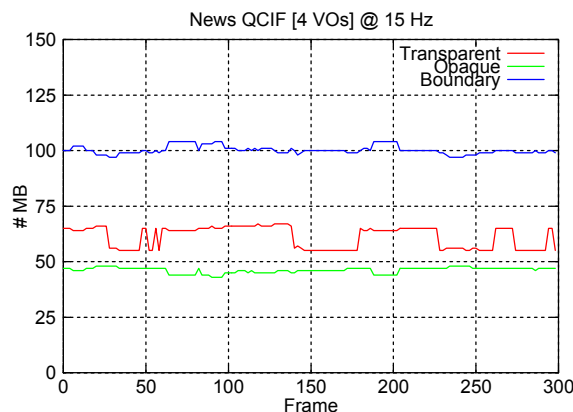


Figure 4.23 – Number of MBs per shape type for the News sequence

4.6.3 VCV Encoder Implementation

This section describes the main steps of the VCV model implementation in the context of an MPEG-4 video encoder. In order to produce valid bitstreams, the encoder must ensure that the two following conditions are always verified:

1. **The VCV buffers never overflow** – The encoder has to guarantee that there is always enough space in the VCV buffers at the time the MBs of the VOP currently being encoded are added to the corresponding buffers.
2. **Each VOP is completely decoded in time** – The encoder must guarantee that the VOPs are completely decoded before they are needed for composition, i.e., before their composition time plus the VCV latency. For this, the encoder has to take into account the decoding rate for each of the VCV buffers.

For each target encoding time (typically the multiples of $1/f_i$, where f_i denotes the temporal coding rates for the various objects in the scene), the encoder analyzes the set of VOPs to be encoded and tests if the above conditions can be verified for this set of VOPs. Whenever the rate control mechanism detects a possible violation of the VCV model, it can take one of the following actions to decrease the occupancy of the VCV buffers:

- Skip the encoding of one or more VOPs for that time instant.
- Reduce the number of MBs with shape information (in case this is the limiting factor), e.g., by changing the shape of the objects or merging objects (if this is an acceptable solution for the application in question; this action impacts on the “authoring” of the scene)
- Decrease the (bounding box) size of some VOPs, e.g., by splitting an object into two

or more objects it may be possible to reduce the total number of MBs in the resulting bounding boxes (if this is an acceptable solution for the application in question; this action impacts on the “authoring” of the scene).

Notice that the overflow of one (or both) of the VCV buffers may lead to the incomplete decoding of one or more VOPs and, consequently, to a coding desynchronization between encoder and decoder.

The video complexity verifier mechanism plays a very important role in MPEG-4 video coding to ensure that the flexibility opened by the object-based approach does not prevent interoperability between different terminals at a reasonable cost, notably in terms of computational load. This interoperability can only be achieved if the VCV model is not violated.

4.7 Analysis of the Video Rate Buffer Verifier

This section analyzes the MPEG-4 VBV mechanism considering different scenarios of operation, notably independent and combined VBV buffer control. Moreover the set of actions that an MPEG-4 video encoder can take whenever, at the encoding time, detects an imminent violation of the VBV model constraints is discussed.

4.7.1 VBV Model Approaches

The VBV model allows the encoder to monitor and control the bitstream memory needed to decode each ES in a scene. The VBV model enables the encoder to correctly schedule data transmission by specifying when information may be removed from the decoder bitstream buffers, so that these buffers do not overflow or underflow.

As referred previously, the required bitstream buffer resources can be indicated to the decoder by means of decoder configuration information, carried at the systems layer, or embedded in the visual syntax. Data scheduling is specified using timing information either by means of systems information or by using the visual syntax.

The encoder generates encoded data as the encoding process progresses and is also responsible for generating the corresponding timing information for each VOP. Consequently, the encoder must ensure that the encoded bits for each VOP are available for decoding at the time corresponding to the VOP decoding time. Additionally, the encoder has to ensure that the decoded VOP is available for composition at the time corresponding to the VOP composition time plus the VCV latency.

In the case of a scene composed by multiple VOs, synchronization can only be achieved if all the VOs adhere to the same time base. In the case of single VO scenes, since there is no synchronization needed among different VOs, the encoded data can be decoded and presented as it arrives provided that the temporal distance between the encoded VOPs in the visual stream is preserved.

This section presents the two major VBV buffer control approaches, in the case of multiple VOs scenes, comparing the adopted MPEG-4 VBV model with these approaches.

INDEPENDENT VBV BUFFER CONTROL APPROACH

In the case of independent VBV buffer control, each VO, in fact each VOL, has its own individual bitstream buffer resources at the decoder and the encoder has to ensure that the limits of each of these buffers are not violated. Regarding the sharing of the total

bandwidth/buffering resources among the various VOs, two possible scenarios can be identified:

I) Fixed Bandwidth Allocation

This is the more straightforward scenario where each VOL in the scene has its own bitstream buffer resources and shares in a pre-defined and rigid way the total channel rate. In this case, independent rate control is applied to each VOL. This situation is illustrated in Figure 4.24.

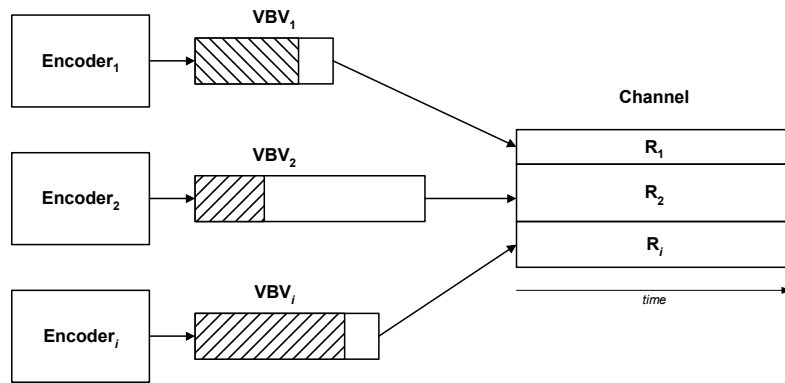


Figure 4.24 – Independent VBV buffer control with fixed bandwidth allocation

II) Dynamic Bandwidth Allocation

From an implementation point of view, it is convenient to jointly control the encoding of the several VOs in order to share resources among them and continuously reach the best global allocation of resources [13]. In this scenario, each VO has its own buffer but the draining rate of each buffer is dynamically adapted at each encoding time instant, according to its occupancy and the VO's characteristics, e.g., size, activity, complexity, etc. This situation is illustrated in Figure 4.25.

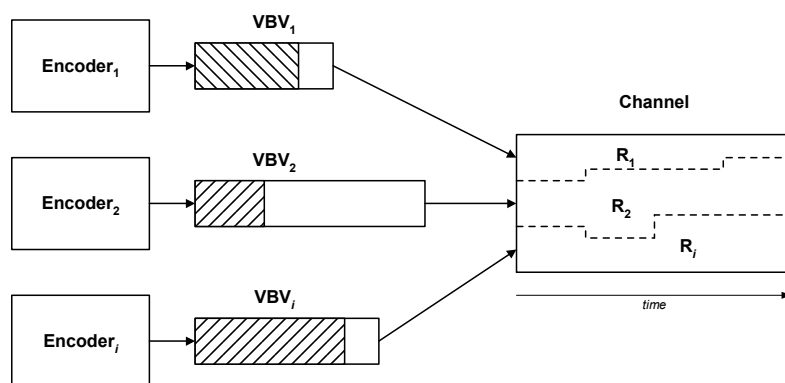


Figure 4.25 – Independent VBV buffer control with dynamic bandwidth allocation

COMBINED VBV BUFFER CONTROL APPROACH

This scenario is similar to the previous one with dynamic bandwidth allocation with the addition that here all VOs share the same buffer and bandwidth resources, i.e., the common buffer draining rate is equal to the channel rate. This situation is illustrated in Figure 4.26.

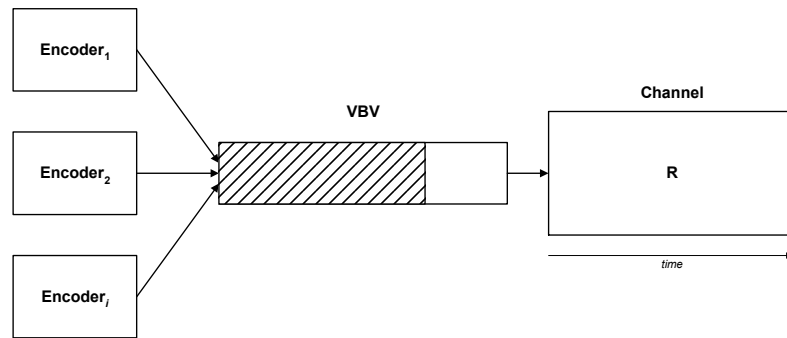


Figure 4.26 – Combined VBV buffer control with shared bandwidth resources

MPEG-4 VBV BUFFER CONTROL APPROACH

As for the VMV and VCV models, the MPEG-4 Visual [29] does not standardize any method for achieving proper VBV buffer control. This freedom allows the implementers to choose the best buffer control mechanism that guarantees proper VBV operation for each case.

As shown in [14] and [17], joint rate control strategies, where the available bit rate is dynamically allocated among the several VOs in the scene according to the instantaneous characteristics of each VO such as the size, motion, and texture complexity, can achieve superior performance, when compared to independent rate control.

Although the MPEG-4 VBV specification states that in the case of a scene composed by multiple VOs²¹, each with one or more VOLs, the VBV model should be applied independently to each VOL [29], the MPEG-4 VBV model supports both independent and combined buffer control. Notice, however, that MPEG-4 Visual does not allow the specification of a single buffer for the whole scene. This fact does not prevent the use of a combined VBV buffer control strategy, provided the sum of all individual buffers does not exceed the profile and level limits.

4.7.2 VBV Encoder Implementation

This section describes the main steps of the VBV model implementation in the context of an MPEG-4 video encoder. In order to produce valid bitstreams, the encoder must ensure that the VBV constraints are always verified. Whenever the rate control mechanism detects a possible violation of the VBV model (underflow or overflow), it should react with the most adequate action in the context of the application in question. This can be done in two distinct modes:

- **Preventive mode** – Whenever the VBV occupancy reaches certain high thresholds, the encoder takes adequate action(s) to neutralize the unwanted situation.
- **Reactive mode** – Whenever the coding decisions taken by the encoder lead to a violation of the VBV constraints, new coding decisions are adopted to avoid this situation until the problem is solved.

²¹ Section D.2 [29] states that “If a visual bitstream is composed of multiple VOs ...”. However such sentence is conceptually incorrect since clause 6.2.1 [29] explicitly states that: “14496-2 does not provide for the multiplexing of multiple elementary streams into a single bitstream. One visual bitstream contains exactly one elementary stream, which describes one layer of one visual object. A visual decoder must conceptually have a separate entry port for each layer of each object to be decoded”.

The reactive mode is typically more effective in achieving proper VBV buffer control than the preventive mode, since it usually involves choosing iteratively from multiple coding decisions the one that, fulfilling the VBV constraints, has the best trade-off in terms of rate-distortion operation. This way, in the reactive mode the encoder can control very precisely the output bit rate of the encoded data. However, the reactive mode is not very suited for real-time encoding since it may involve multiple encoding passes for the same data and thus may involve extra delays that are not acceptable for real-time.

Besides being less effective in achieving a certain target bit rate, the preventive mode cannot easily guarantee that the VBV will not be violated since it only has one encoding chance; the less severe the VBV thresholds to operate the preventive mode are and the less conservative the preventive mode actions are, the higher is the risk that a VBV violation happens. In the case of imminent VBV violation, and in order to still create a compliant set of streams, the encoder may be forced to take extreme actions resulting in a severe subjective impact, e.g., skip the encoding of the remaining MBs in a given VOP that has spent too many bits, thus threatening to overflow the encoder buffer.

These two modes could also be applied to the VMV and VCV models; however, for the VMV and VCV models proper operation can easily be guaranteed with the preventive mode since they both deal with precisely measurable resources, even in advance. For the VBV model, this is not true because it is extremely difficult to predict the actual number of bits that will be used for a certain set of encoding decisions, before really encoding the data, and thus large deviations from the target bit rate values may occur. For the VBV model, only the reactive mode can absolutely guarantee that the VBV is never violated without taking extreme actions. This fact justifies its use in some scenarios such as off-line encoding.

Typically, the encoder rate control can take one of the following actions to prevent or avoid VBV violation:

- Adjust the MB quantization parameters (texture distortion).
- Adjust the shape encoding losses (shape distortion).
- Skip the encoding of some of the incoming VOP(s).
- Introduce stuffing bits (increase the VBV buffer occupancy if encoder underflow is the problem).

Both the overflow and underflow of the VBV buffer must be avoided. Notice that the VBV buffer overflow may lead to a loss of data and thus to the incomplete decoding of one or more VOPs, while the VBV buffer underflow may prevent the decoder to decode the incoming data on time and thus both may lead to a coding desynchronization between encoder and decoder.

The video rate buffer verifier mechanism plays a very important role in MPEG-4 video coding to ensure that the flexibility opened by the object-based approach does not prevent interoperability between different terminals at a reasonable cost, notably in terms of bitstream memory. This interoperability can only be achieved if the VBV model is not violated.

4.8 Final Remarks

This chapter presented a detailed analysis of the MPEG-4 video buffering verifier mechanism highlighting its major features and drawbacks and comparing the adopted MPEG-4 model with other relevant alternative models.

For the VMV model, this analysis showed that when B-VOPs are used the adopted MPEG-4

VMV solution does not take into account all the picture memory needed at the decoder, notably the memory needed for their composition, but overestimates the picture memory needed only for the decoding process. Since composition is not normative in MPEG-4, a Decoding Memory approach should have been fully adopted for the MPEG-4 VMV model to clearly reflect that situation.

With respect to the VCV model, it was shown that several scenes typically targeted to be encoded with either Core Profile @ Level 1 or Core Profile @ Level 2 exceed the VCV buffer size limits imposed by the Profile and Levels definitions, preventing these scenes to be encoded with these profile@level combinations. This limitation of the MPEG-4 VCV model is mainly due to the MB decoding complexity model adopted, which classifies the different MBs in a scene, in terms of decoding complexity, based only on the boundary and non-boundary distinction. This approach overestimates the decoding complexity of some MB types, notably transparent MBs, and thus overestimates the decoding complexity of scenes containing a large number of such MBs. Moreover, the MPEG-4 VCV does not take into account the main factors that determine the actual differences among MBs regarding their decoding complexity, i.e., the MB coding types used (texture and shape), as shown in [21].

To overcome the limitations of the MPEG-4 VCV model, a VCV model where the MB complexity would be based on the MB coding type would be more adequate. The Single Buffer with Single Decoding Rate with MB weights or the Multiple Buffers with Multiple Decoding Rates solutions appear as the best candidates for the VCV modeling in MPEG-4 video coding. In this context, a VCV model based on the former approach – the IST VCV model [22] – has been compared with the current MPEG-4 VCV model and has shown to be more effective in evaluating the actual scene decoding complexity. The efficient use of decoding resources is very important, notably for application environments where resources are scarce and expensive such as mobile applications.

Regarding the VBV model, it was shown that although the MPEG-4 VBV specification states that in the case of a scene composed by multiple VOs, each with one or more VOLs, the VBV model should be applied independently to each VOL, the MPEG-4 VBV model can still support the dynamic bit rate allocation among the several VOs in the scene. This is especially important since as it was shown in [14, 17] this dynamic bit rate allocation can achieve superior performance in terms of the rate-distortion trade-off, when compared to fixed bit rate allocation.

The video buffering verifier mechanism constitutes a fundamental piece of the MPEG-4 video profiling approach and consequently a very important tool of any MPEG-4 video encoder. This mechanism plays a central role in MPEG-4 video coding to ensure that the flexibility opened by the object-based approach does not prevent interoperability between different terminals, achieved at a reasonable cost, notably in terms of picture memory, computational load, and bitstream memory.

Although the video buffering verifier mechanism is essentially described in terms of decoder operation, it is a task of the encoder to guarantee that it is never violated. For this, the encoder has to “shape” the encoded data in a way that it does not violate the constraints imposed by the three models building this mechanism. Such task is mainly dealt with by the rate control module that takes into account the status of the several video buffering verifier buffers for the best control of the encoder in terms of achieving the best quality performance. In this context, a model for the integration of the video buffering verifier mechanism in a video encoder including the rate control module was also proposed and analyzed.

Chapter 5

Rate-Distortion Modeling for Low-Delay Video Encoding

5.1 Introduction

As mentioned in Chapter 2, the MPEG-4 standard defines a toolbox of coding techniques supporting a wide range of applications, notably involving high-quality, high-complexity professional encoders as well as low-cost, low-complexity consumer products. In this context, any video encoding scheme includes a trade-off along the following four dimensions [143]: 1) Distortion, 2) Rate, 3) Complexity, and 4) Delay.

Typically, the aim of a given encoder is to minimize the coded bit rate given a minimum (acceptable) target decoded video quality – *distortion constraint*, or to maximize the perceived video quality given a maximum target bit rate – *rate constraint*. Most of the times, this rate-distortion trade-off has to be achieved taking into account encoder and decoder implementation restrictions – *complexity constraint*, and end-to-end delay restrictions – *delay constraint*.

These constraints have different importance depending on the type of application involved, e.g., for broadcasting applications the rate-distortion trade-off and, in some cases, the decoder complexity are the more important constraints, while for two-way real-time communications the codec complexity and the end-to-end delay become more important constraints.

In terms of rate control, the more relevant contributions, in terms of delay, are the processing and the buffering delay (see Section 3.3), since these are the ones directly related to the encoding process. An encoding system where the encoder is able to pre-analyze several

pictures before really encoding them enables the rate control mechanism to take more guided actions since the ‘future is known in advance’. The buffer size also constrains the type of rate control required: the higher the buffer dimension, the higher the allowable bit per picture variability and, consequently, a looser rate control may be sufficient; on the contrary, for small buffers, the number of bits per picture needs to be kept nearly constant and, consequently, a very tight rate control is needed.

In the context of this Thesis, a low-delay video encoding scenario will be characterized by:

- Low processing delay.
- Low buffering delay.

Given some pre-defined restrictions, such as complexity and delay constraints, rate control methods aim essentially at achieving an optimal rate-distortion trade-off. To achieve this goal the rate control mechanisms need to carry on appropriate control actions, e.g., to define the appropriate encoding time instants or to allocate the available bit rate. In this context, two fundamental issues emerge:

- **Rate-distortion modeling** – Targets the design of adequate models for describing the rate-distortion behavior associated to the encoding system. These models must capture the statistical characteristics of the source and describe the encoding process as a function of some encoder control parameters, reflecting the lossy encoding rate-distortion trade-off.
- **Control process modeling** – Targets the design of a suitable model for describing the encoding process and the corresponding control actions.

Optimal encoding in a rate-distortion sense requires that the encoding parameters be adequately adjusted at proper encoding time instants; moreover, whenever the encoding process dynamics deviate from the underlying model, the rate control system should adapt to these changes.

This chapter considers the problem of rate and distortion modeling for Intra and Inter coding in the context of object-based MPEG-4 video encoding, notably for low-delay video encoding scenarios. In the case of Intra coding, the VOP to encode does not depend on other past or future VOPs; therefore, its rate and distortion characteristics depend exclusively on the current quantizer parameter(s) and VOP statistics. In the case of Inter coding, the rate and distortion functions depend not only on the current VOP but also on its reference VOP(s); therefore the rate and distortion functions become bidimensional and consequently more difficult to estimate during encoding. For Intra coding, this chapter follows a conventional approach, modeling the rate and distortion characteristics as rate-quantization, distortion-quantization, or rate-distortion functions. For each of these three modeling scenarios an efficient (low fitting error) model is proposed. In the case of Inter coding, a new rate-distortion modeling approach is proposed where the rate-quantization and distortion-quantization functions are approximated by a stationary component, depending only on the quantization parameter of the current VOP, plus a adaptation term that intends to compensate the fact that, when Inter-coded VOPs are encoded with a quantization parameter higher or lower than the average quantization parameter, the rate-quantization and distortion-quantization models tend to deviate from the stationary model.

The chapter is organized as follows: after this introduction, Section 5.2 introduces some basic concepts of rate-distortion theory, with emphasis on rate-distortion modeling for DCT-based video, and provides a critical analysis of applying these concepts in practical video coding

systems; Section 5.3 proposes a set of rate and distortion models, some of them, used for different rate control objectives in the remaining of this Thesis; finally, Section 5.4 summarizes the main conclusions and contributions of this chapter.

5.2 Rate-Distortion Modeling

Although the early seventies have seen some interest in applying rate-distortion theory to pictures [144], “rate-distortion theory and the practice of lossy source coding have become more closely connected today than they were in the past”, as mentioned by Berger in 1998 [145].

As stated in Shannon’s seminal paper [146], “The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point”. According to [147], this problem can be divided into two component problems:

- What information should be transmitted?
- How should it be transmitted?

While in [146] most of Shannon’s attention was devoted to give appropriate answers to the second part of this problem, notably how to deal with the effect of noise and how to take advantage of the statistical properties of the source, only a decade after he first introduced the fundamentals of information theory some attention was devoted to the first part of the problem in [148]. There, Shannon addressed the problem of coding a discrete source of information assuming a certain tolerable level of distortion of the recovered information compared to the original information.

The problem of transmitting an exact or approximate copy of the original information sets the boundary between lossless and lossy compression. In lossless compression, the encoded video has exactly the same amount of information as the original video from an information theory point of view, which means that the reconstructed video is equal to the original, i.e., there is no loss of information in the compression process. However, the maximum compression ratios that can be obtained with a given lossless compression scheme are theoretically bounded by the entropy of the source [149].

Higher compression ratios can only be achieved through lossy compression, which means that the reconstructed video is no longer equal to the original, i.e., there is some loss of information in the compression process. Conceptually, with lossy compression any compression ratio can be achieved by successively dropping information from the original.

The trade-off between compression ratio (or equivalently the used rate) and distortion is inherent to every lossy video compression scheme, i.e., a low distortion requires a high rate (low compression ratio) while a high distortion requires a low rate (high compression ratio), as illustrated in Figure 5.1.

Rate-distortion (RD) optimization, seen as the selection of a set of encoding parameters that maximizes some function that reflects a rate-distortion trade-off, is far from being a trivial task. In fact, complexity is a determining factor in using RD optimization techniques since they are typically computationally intensive. In this case there are two major sources of complexity [150]:

- RD data collection, which may involve several encoding/decoding operations.
- Selection of the best RD operation point.

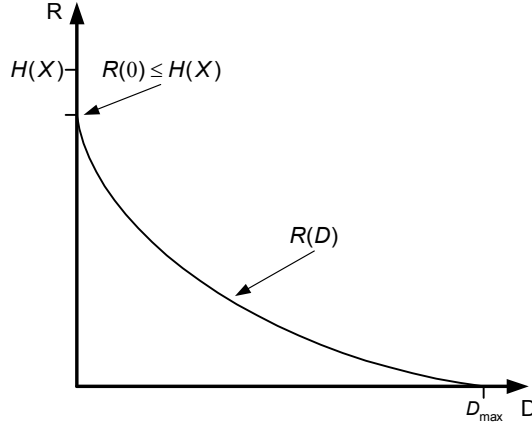


Figure 5.1 – Typical rate-distortion function for a discrete source

5.2.1 Fundamentals of Rate-Distortion Theory

Traditionally, rate-distortion theory addresses the minimization of the channel capacity requirement while holding the average distortion at or below an acceptable level [144]. Conversely, if a certain channel capacity, C , is available, the rate-distortion function can be used to determine the minimum average distortion, D , that verifies the condition $R(D) < C$, for error-free transmission.

Figure 5.2 shows a block diagram of a typical communication system. Since the aim of the work presented in this Thesis is concerned only with the problem of lossy source coding, an error-free transmission is assumed, that is $U = V$, which means that the distortion between the source output, X , and the destination input, Y , is exclusively due to the encoding process.

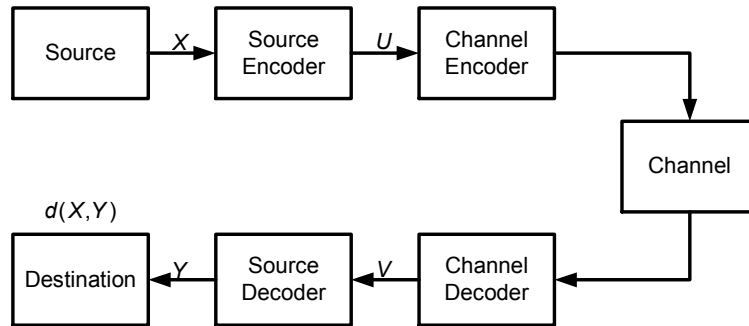


Figure 5.2 – Rate-distortion encoder and decoder model [151]

Assuming a time discrete source of information, generating a sequence of amplitude discrete symbols, encoded in successive blocks of length N , each block can be described by one of an enumerable set of messages $\{X_i\}$ with a probability function $P(X_i)$.

Such system can be characterized by the conditional probability $Q(Y_j | X_i)$ of message Y_j being decoded given that message X_i has been encoded, i.e.,

$$Q(Y_j | X_i) = \frac{P(X_i, Y_j)}{P(X_i)}.$$

In this context, the marginal probability function of the decoded message is

$$P(Y_j) = \sum_i P(X_i)Q(Y_j | X_i).$$

An insightful theoretical measure related to the information transmitted is the average mutual information between X and Y for each block of length N , which expresses the information acquired about X when Y is received, and is given by

$$I_N(X, Y) = \sum_{i,j} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(X_i)P(Y_j)} = \sum_{i,j} P(X_i)Q(Y_j | X_i) \log \frac{Q(Y_j | X_i)}{P(Y_j)} \quad (5.1)$$

The mutual information is measured in bits if a base-2 logarithm is used or nats if the natural logarithm is used.

The concept of mutual information is derived from the concept of self-information of a discrete random variable, i.e., the information gained by observing the outcome of the random variable X , which is defined as

$$I(X_i) = -\log P(X_i)$$

The rationale for such definition is that the information acquired by knowing that a low probability outcome occurred is high, while the information acquired by knowing that a high probability outcome occurred is low, e.g., the self-information acquired by knowing that your neighbor won the lottery is much higher than the information acquired by knowing that he has lost. Notice that the average self-information for a given source defines the entropy of the source, i.e.,

$$H(X) = -\sum_i P(X_i) \log P(X_i)$$

For lossless encoding, i.e., when $Y = X$, then

$$Q(Y_j | X_i) = \begin{cases} 1, & j = i \\ 0, & j \neq i \end{cases} \text{ and } P(Y_i) = P(X_i)$$

In this case, the information transmitted is equal to the average self-information of X , also called the N th-order entropy of the source, i.e.,

$$I_N(X, Y) = H_N(X) = -\sum_i P(X_i) \log P(X_i) = H_N(Y) \quad (5.2)$$

which means that lossless encoding requires at least $H_N(X)$ bits. When the encoding is lossy, i.e., $Y \neq X$, then (5.1) becomes

$$\begin{aligned} I_N(X, Y) &= \sum_{i,j} P(X_i, Y_j) \left[\log \frac{1}{P(X_i)} + \log \frac{P(X_i, Y_j)}{P(Y_j)} \right] \\ &= \sum_i P(X_i) \log \frac{1}{P(X_i)} + \sum_{i,j} P(X_i, Y_j) \log \frac{P(X_i, Y_j)}{P(Y_j)} \\ &= H_N(X) - H_N(X | Y) = H_N(Y) - H_N(Y | X) \end{aligned} \quad (5.3)$$

where $H_N(X | Y)$ is the conditional entropy of the source encoder input (source output) X given the observed source decoder output Y .

Since the entropy is a measure of uncertainty, the mutual information between X and Y given by (5.3) is the uncertainty in the source output X minus the uncertainty in the source output X given the observed decoder output Y . In the case of lossless encoding, the uncertainty in the source output X given the observed decoder output Y is zero, i.e., $H_N(X|Y) = 0$, since $Y = X$, thus the mutual information between X and Y is equal to the entropy of the source $H_N(X)$. In the extreme case, where X and Y are independent, then $H_N(X|Y) = H_N(X)$ and the average mutual information between X and Y is zero, i.e., no information is transmitted. Moreover, given that $H_N(X|Y) \leq H_N(X)$, the average mutual information between X and Y is bounded, i.e., $0 \leq I_N(X, Y) \leq H_N(X)$.

For the work presented in this Thesis, the case of greater interest is the lossy encoding case, where Y exhibits a strong statistical dependency on X , thus $H_N(X|Y)$ is neither zero nor equal to $H_N(X)$. For a lossy encoding scheme characterized by a conditional probability $Q(Y_j | X_i)$ with a distortion measure between X and Y defined by $d(X, Y)$ that verifies the following conditions

$$\begin{cases} d(X_i, Y_j) \geq 0 \\ d(X_i, Y_j) = 0 \quad \text{for } X_i = Y_j \end{cases}$$

the average distortion per source symbol is given by

$$D(Q) = \frac{1}{N} E[d(X, Y)] = \frac{1}{N} \sum_{i,j} P(X_i) Q(Y_j | X_i) d(X_i, Y_j) \quad (5.4)$$

The N -block rate-distortion function, $R_N(D^*)$, is, by definition, the minimum of the average mutual information per symbol, for an average distortion per symbol, given by (5.4), less or equal than D^* , where D^* is a maximum (acceptable) average distortion.

Since the source is given, the optimization of the encoding scheme can only be done over the conditional probability $Q(\cdot | \cdot)$. Thus, the N -block rate-distortion function is defined as

$$R_N(D^*) = \inf_{Q \in Q_D} \left[\frac{1}{N} I_N(X, Y) \right] \quad (5.5)$$

where $\inf[\cdot]$ is the infimum¹ of the set defined by the average mutual information per symbol for all encoding schemes belonging to Q_D , and Q_D is the set of all possible encoding schemes such that $D(Q) \leq D^*$, i.e.,

$$Q_D = \{Q(Y_j | X_i) : D(Q) \leq D^*\}$$

As the block size, N , increases the N -block rate-distortion function converges to the rate-distortion function of the source,

$$R(D^*) = \lim_{N \rightarrow \infty} R_N(D^*) \quad (5.6)$$

¹ “The infimum is the greatest lower bound of a set S , defined as a quantity m such that no member of the set is less than m , but if ε is any positive quantity, however small, there is always one member that is less than $m + \varepsilon$ ” [152].

That is, the rate-distortion function of the source is the limit, as the block size tends to infinity, of the average mutual information per source symbol, subject to the constraint that the average distortion be less or equal than D^* , where the minimization is carried over all possible encoding schemes characterized by the conditional probability $Q(Y_j | X_i)$. This function possesses interesting properties, notably continuity, differentiability, convexity, and monotonicity [147].

Real encoding systems are limited to rather small block sizes to reduce the computational complexity and encoding delay; e.g., in MPEG-4 video, transform coding and quantization are carried over 8×8 blocks of samples. Due to these complexity and delay trade-offs, the theoretical rate-distortion limit can hardly be achieved.

The central entity in rate-distortion theory is then the rate-distortion function, $R(D)$ ², that represents the minimum required bit rate to encode a given source with an average distortion D . As can be seen in Figure 5.1, $R(0)$ equals the entropy of the source, $H(X)$, when the distortion measure requires perfect reconstruction for $D = 0$; otherwise, $R(0) \leq H(X)$, e.g., for subjective distortion measures where visually lossless (or transparency) does not require perfect reconstruction (i.e., does not strictly require that $X = Y$). Similarly, the maximum distortion is bounded by a positive real value, D_{\max} , corresponding to the case where no information is sent, i.e., $R = 0$.

Berger [147] developed a closed-form solution³ for the rate-distortion function for Gaussian uncorrelated sources and a squared-error distortion measure D

$$R(D) = \begin{cases} \frac{1}{2} \log_2 \frac{\sigma^2}{D}, & 0 \leq D \leq \sigma^2 \\ 0, & D > \sigma^2 \end{cases} \quad (5.7)$$

where σ^2 is the variance of the Gaussian source and $D = (X - Y)^2$.

The basic problem in rate-distortion theory can then be stated as follows: *given a source distribution and a distortion measure, what is the minimum expected distortion achievable at a particular rate? Or, equivalently, what is the minimum rate to achieve a particular distortion?* [84].

Up to this point, the problem addressed by rate-distortion theory is that of determining the smallest rate at which information can be sent from source to destination in order to achieve a predefined fidelity [147]. The power of rate-distortion theory comes from the ability to set bounds on the performance of any lossy data compression scheme. However, by itself does not provide a mechanism for constructing a scheme that achieves the bounds. According to [151], there are three major difficulties in trying to apply rate-distortion theory to the coding of video information:

1. The $R(D)$ function is usually very difficult to compute.
2. The source is difficult to characterize statistically: the Gaussian assumption does not hold generically; statistics vary considerably from picture to picture and from region to

² The superscript ^{*} has been dropped for simplicity.

³ “An equation is said to be a closed-form solution if it solves a given problem in terms of functions and mathematical operations from a given generally accepted set. (...)” [153].

region within an picture.

3. It is difficult to find a distortion measure that is simultaneously meaningful and mathematically tractable.

Applying rate-distortion theory to practical encoders requires setting up some trade-offs along the following dimensions:

- **Model mismatch** – How close the model characterizes the source and the model assumptions are valid.
- **Complexity** – The amount of memory, delay, and computational power required to approach the theoretical bounds.

A good rate-distortion model should capture the essential statistical dependencies of the source and simultaneously meet the system requirements, notably, in terms of encoding complexity, delay, and memory.

5.2.2 Operational Rate-Distortion

Lossy encoding schemes, such as those based on transform coding, scalar quantization, are characterized by a finite set of quantizer and other coding parameters, leading consequently to a finite set of rate-distortion points – Admissible Rate-Distortion Points (see Figure 5.3a). However, not all admissible rate-distortion points for a given coding scheme and a given source are optimal in a rate-distortion sense. The set of optimal rate-distortion points define the Operational Rate-Distortion Function (ORDF) [154]. In this context, a (D, R) point belongs to the ORDF if there is no other (D, R) point with a lower distortion and the same or lower rate or, equivalently, if there is no other (D, R) with lower rate and the same or lower distortion (see Figure 5.3a).

Sometimes it is convenient to see the ORDF as a continuous function (dotted line in Figure 5.3a), e.g., to compare the performance of competing coding schemes (as illustrated in Figure 5.3b, coding scheme A performs better than coding scheme B for rates above R_1 , while coding scheme B performs better than coding scheme A for rates below R_1).

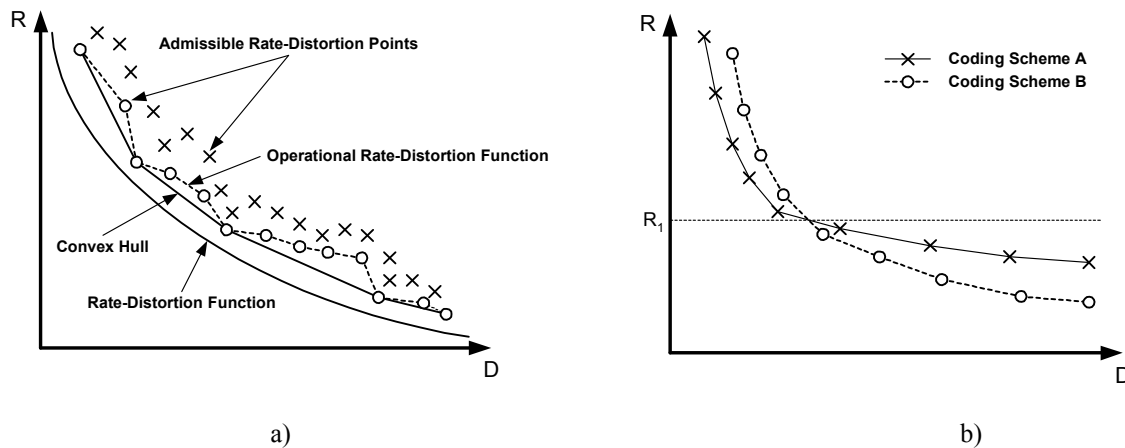


Figure 5.3 – Rate-distortion plots: a) computation of the ORDF from the set of admissible rate-distortion points; b) comparison of coding schemes using the ORDF

In a rate-distortion context, encoding optimization consists in selecting the best coding parameters for a given source, such that for a given target bit rate the distortion is minimized

or, equivalently, for a given maximum distortion the encoder bit rate is minimized.

Some optimization algorithms, in searching for the best rate-distortion operation point, require that the rate-distortion characteristic be convex. Notice that, a given function is convex if, for any pair of values x_1 and x_2 , and any real number $\lambda \in [0,1]$, the following condition is verified (see Figure 5.4)

$$f((1-\lambda) \cdot x_1 + \lambda \cdot x_2) \leq (1-\lambda) \cdot f(x_1) + \lambda \cdot f(x_2).$$

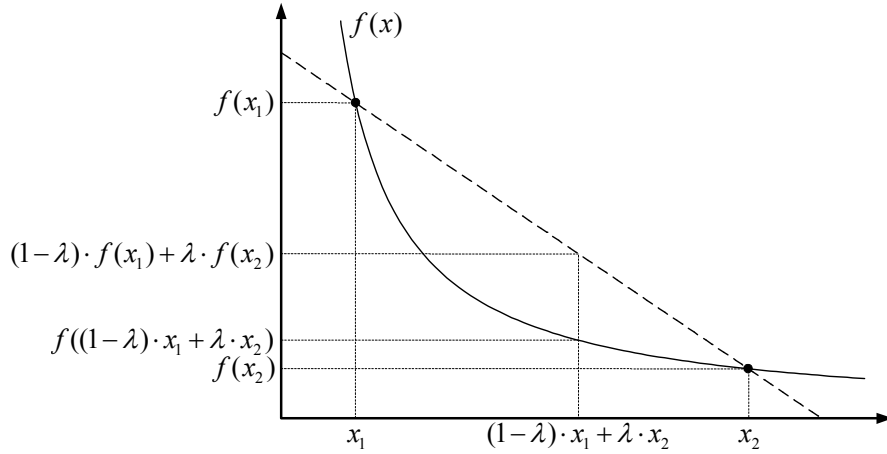


Figure 5.4 – Example of a convex function

In this context, the convex hull of the set of admissible rate-distortion points is used, which is formed by the segments connecting the highest number of (D, R) points belonging to the ORDF, such that the resulting curve is convex (see Figure 5.3a).

The operational rate-distortion function is useful to find the optimal performance of a given compression scheme, while the rate-distortion function defines the theoretical limit independently of the coding scheme used, thus serving as a benchmark. Optimal performance in a rate-distortion sense is closely related to optimal bit allocation, i.e., how to distribute the available bits among different sources of information, such that the overall distortion is minimized.

5.2.3 The Constrained Bit Rate Control Problem

Typically the rate control problem can be formulated as a constrained problem as follows: *given a set of M coding units (pictures, MBs, etc.), a bit budget R_{\max} , and a certain distortion measure, D , find the set of coding parameters (typically the set of quantizers) that minimizes the overall distortion without exceeding the bit budget, i.e.,*

$$\text{minimize } \sum_{i=1}^M D_i(Q_i) \text{ subject to } \sum_{i=1}^M R_i(Q_i) \leq R_{\max} \quad (5.8)$$

where $R_i(\cdot)$ and $D_i(\cdot)$ are, respectively, the rate and distortion functions, and Q_i the quantizer parameter of each coding unit i , such that $Q_i \in \{Q_1, Q_2, \dots, Q_N\}$.

This problem can be solved using Lagrange optimization [86], which is a mathematical tool to solve constrained optimization problems. The main strength of this method is the possibility to convert the hard to solve constrained problem (5.8) into a more easy to solve unconstrained

problem (5.9)

$$\text{minimize } \sum_{i=1}^M D_i(Q_i) + \lambda \sum_{i=1}^M R_i(Q_i) \quad (5.9)$$

This formulation is supported by the following theorem, derived from the work presented in [86].

Theorem: Let S be the finite set all admissible quantizer vectors and $\mathbf{Q} \in S$ a member of that set. Let $R(\mathbf{Q})$ and $D(\mathbf{Q})$ be real-valued functions defined over S . Then, for any $\lambda \geq 0$, the optimal solution $\mathbf{Q}^*(\lambda)$ for the unconstrained problem,

$$\min_{\mathbf{Q} \in S} (D(\mathbf{Q}) + \lambda R(\mathbf{Q})), \quad (5.10)$$

is also the optimal solution of the constrained problem,

$$\min_{\mathbf{Q} \in S} D(\mathbf{Q}) \text{ subject to } R(\mathbf{Q}) \leq R(\mathbf{Q}^*(\lambda)). \quad (5.11)$$

The theorem does not guarantee any solution for the constrained problem (5.11). Its main result is that for any non-negative λ , there is a constrained problem whose solution is identical to the unconstrained problem (5.10). In this case, if $R_{\max} = R(\mathbf{Q}^*(\lambda))$ then $\mathbf{Q}^*(\lambda)$ is the desired solution for the constrained problem (5.11). In [154] and [86], some iterative methods to find a λ such that $R_{\max} = R(\mathbf{Q}^*(\lambda))$ are proposed.

The Lagrange optimization method, also called Lagrange multiplier method, constitutes a powerful tool for solving the rate control problem in a rate-distortion encoding framework that can be described in a form similar to (5.8). In this context, deriving appropriate rate-distortion models is of particular importance since these models can significantly reduce the computational complexity of the rate control process as they can be used to estimate the operational rate-distortion points without requiring the actual encoding of the data.

5.2.4 Rate-Distortion Modeling for DCT-based Video Coding

Devising a good image or video compression algorithm involves selecting an appropriate source model and, given that model and any relevant bounds, striving to optimize the coding performance [150]. Therefore, a source model for good rate-distortion performance should be:

- Simple enough in order that a compression scheme tailored to this model can achieve good performance with reasonable cost.
- Complex enough to capture the main characteristics of the source.

Since the goal of this Thesis is to develop efficient rate control algorithms for object-based video coding, this section analyses the rate-distortion modeling for this type of encoders.

The main objective of transform coding is to decompose the source pictures into components exhibiting more stationary statistics and that can be coded separately (decorrelation), and additionally to concentrate the energy in a small number of coefficients (energy compaction) [151]. In this context, the Karhunen-Loève Transform (KLT) is the optimal transform; though the KLT basis functions are source dependent, which makes it less viable for practical encoders. The DCT, however, has close performance to the optimal KLT in the sense of energy compaction and signal decorrelation [143] and the DCT basis functions are source independent, thus making it more functional in terms of coding and transmission.

Signal decorrelation allows modeling each DCT coefficient representing a given frequency as a memoryless source (e.g., Gaussian or Laplacian) [150]. Additionally, the transform domain allows applying frequency selective quantization (distortion), which is more related to the human perception of visual information. Moreover, when the DCT coefficients are ordered in zigzag, their mean square value is approximately monotonically decreasing [151].

As mentioned above, high compression ratios can only be achieved through lossy compression, which is mostly achieved through quantization of the original signal or, in the case of transform-based coding, through quantization of the resulting transform coefficients.

A typical distortion measured used in lossy compression is the square error, i.e.,

$$D = \sum_{i=1}^N E \left[\left| X_i - \hat{X}_i \right|^2 \right] = E \left[\left\| \mathbf{X} - \hat{\mathbf{X}} \right\|^2 \right] \quad (5.12)$$

where X is the original signal and \hat{X} its quantized version.

Since the DCT is an orthogonal transform, i.e., $\mathbf{T}^T = \mathbf{T}^{-1}$, the distortion measure is preserved by the transformation, i.e.,

$$D = E \left[\left\| \mathbf{Y} - \hat{\mathbf{Y}} \right\|^2 \right] = E \left[\left\| \mathbf{X} - \hat{\mathbf{X}} \right\|^2 \right] \quad (5.13)$$

with $\mathbf{Y} = \mathbf{T} \cdot \mathbf{X}$.

This result also implies that the total energy of an picture block of dimension N in the frequency domain equals the corresponding block energy in the spatial domain [155], i.e.,

$$\sum_{i=1}^N x_i^2 = \sum_{i=1}^N y_i^2 \quad (5.14)$$

Assuming that each DCT coefficient, y_i , is independent and uniformly quantized with a quantizer step Q_i , for the quadratic distortion measure (5.12), the distortion introduced by the quantization process is expressed as

$$D = \sum_{i=1}^N E \left[\left| Y_i - \hat{Y}_i \right|^2 \right]$$

where \hat{y}_i is the reconstructed value of coefficient y_i after quantization and reverse quantization, which can also be represented by $\hat{y}_i = Q(y_i)$.

Under the assumption of high-resolution quantization, i.e., when the quantizer step is sufficiently small such that $D_i \ll \sigma_{Y_i}^2$, where D_i is the coefficient distortion and $\sigma_{Y_i}^2$ its variance, then the average distortion per DCT coefficient is given by

$$D_i = E \left[\left(Y_i - \hat{Y}_i \right)^2 \right] = \frac{Q_i^2}{12} \quad (5.15)$$

where Q_i is the quantizer step of the i -th coefficient [155].

Using (5.12) and (5.15) leads to an average distortion for a block of dimension N given by

$$D = \frac{1}{N} \sum_{i=1}^N \frac{Q_i^2}{12} \quad (5.16)$$

Notice, that this result was obtained under the assumption of high-resolution quantization [155], which means that it should be seen as an upper bound of the average distortion per block.

In the case of Inter coding, it is not the input block that is coded but the prediction error, $e_n = X_n - \tilde{X}_n$, where \tilde{X}_n is a prediction for X_n (see Figure 5.5). In this case, the quantization is applied to the prediction error, e_n , and the quantized prediction error, $\hat{e}_n = Q(e_n)$, is sent to the decoder. Notice that both the encoder and the decoder must have the same prediction, \tilde{X}_n , i.e.,

$$\begin{aligned} e_n &= X_n - \tilde{X}_n \\ \hat{e}_n &= \hat{X}_n - \tilde{X}_n \end{aligned} \quad (5.17)$$

From (5.17) the following equality can be easily derived

$$E[(e_n - \hat{e}_n)^2] = E[(X_n - \hat{X}_n)^2] \quad (5.18)$$

which means that the mean square error in reproducing X_n with \hat{X}_n is equal to the mean square error obtained by quantizing the prediction error e_n . This sets the fundamental theorem of predictive quantization [155], which states that the quantization error obtained by quantizing the signal or the prediction error is the same.

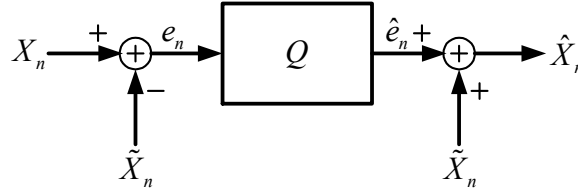


Figure 5.5 – Prediction error quantization

As mentioned in [156], for small distortions of a quadratic distortion measure, the rate-distortion function, $R(D)$, of an entropy-coded uniform quantizer for a zero-mean independent identically distributed (i.i.d.) random variable X can be approximated by

$$R(D) = \frac{1}{\alpha \log_2 e} \log_2 \left(\varepsilon^2 \frac{\sigma_X^2}{D} \right) \quad (5.19)$$

while D can be approximated by

$$D = \frac{Q^2}{\beta} \quad (5.20)$$

where Q is the quantizer step, $\beta = 12$, $\alpha = 2/\log_2 e$ for uniform, Laplacian, and Gaussian distributions, ε is source dependent (according to [157], ε^2 is 1 for Gaussian distributions, $2e^2/2\pi e$ for Laplacian, and $12/2\pi e$ for uniform), and σ_X^2 is the source variance.

It is important to notice that for memoryless, non-Gaussian sources, there is no explicit rate-distortion function but upper and lower bounds are available, where the upper bound is the rate-distortion function for the Gaussian source given by (5.19), i.e., for $\varepsilon^2 = 1$, and the lower bound is given by

$$R_L(D) = H(X) - \frac{1}{2} \log_2(2\pi e D) \quad (5.21)$$

where $H(X)$ is the source entropy.

Combining (5.19) and (5.20) leads to an explicit rate-quantization function

$$R(Q) = \frac{1}{\alpha \log_2 e} \log_2 \left(\varepsilon^2 \beta \frac{\sigma_X^2}{Q^2} \right) \quad (5.22)$$

Notice that this theoretical formulation is only accurate when the quantization step is smaller than the signal standard deviation, since for large quantizer steps it is likely that the signal will be quantized to zero and the maximum average distortion, in this case, will be bounded by the signal variance. Moreover, in a real encoding system, the entropy coder is not ideal, thus the actual number of bits produced by such coder will be higher. In this case, the actual encoding rate may be adjusted by a scaling factor, i.e., $R_{\text{actual}} = k \cdot R$ with $k > 1$.

In a typical DCT-based coder, the transform is used to decompose the input pictures into ideally independent components, the uniform quantizer is used to reduce the number of possible values, and the VLC encoder is used to further compress the coded data.

The quantization process usually assumes a Rayleigh distribution for the DC coefficient and a Laplacian or Gaussian distribution for the remaining coefficients [158]. Since each coefficient is a linear combination of the block samples, the central limit theorem suggests that, as the block size increases, the coefficients tend to have a Gaussian distribution. For the DC coefficient of Intra coded blocks, this result is not valid, since the DC coefficient will always be positive.

As expressed by (5.7), rate-distortion theory indicates that a Gaussian random variable with variance σ_X^2 cannot be represented with less than $\frac{1}{2} \log_2(\sigma_X^2/D)$ bits for a reconstruction mean-square error less than D . This fact suggests that the coefficients should be selected based on their variance. According to [151], when the DCT coefficients are ordered according to increasing spatial frequencies their variance is approximately monotonically decreasing.

The output of the DCT encoder is then typically re-ordered (zigzag scan order) into a vector $X = [X_1 \cdots X_N]$ of DCT coefficients, such that the lower indices correspond to lower spatial frequencies and the higher indices correspond to higher spatial frequencies.

If the DCT vector is stationary and each coefficient can be considered independent, then the block entropy is the sum of the entropies of each coefficient, i.e.,

$$H(X) = \sum_{i=1}^N H(X_i) \quad (5.23)$$

Using the same rationale, it is possible to derive from (5.19) and (5.23) that the average number of bits per quantized DCT coefficient, i.e.,

$$\bar{R}(\bar{D}) = \frac{1}{N} \sum_{i=1}^N R_i(D_i) = \frac{1}{N \alpha \log_2 e} \log_2 \left[\prod_{i=1}^N \left(\varepsilon_i^2 \frac{\sigma_i^2}{D_i} \right) \right] \quad (5.24)$$

where the average distortion per quantized DCT coefficient is given by

$$\bar{D} = \frac{1}{N} \sum_{i=0}^{N-1} D_i = \frac{1}{N} \sum_{i=0}^{N-1} \frac{Q_i^2}{\beta_i} \quad (5.25)$$

Notice that in the MPEG-4 Visual standard [29], there are two quantization methods:

- **Quantization method I** – In this method, each AC coefficient has a different quantizer composed by a quantizer scale, Q_S , constant for the entire block, and a weighting factor, W_i , that is coefficient dependent. In this case, the quantizer step for each coefficient is $Q_i = Q_S \cdot W_i$.
- **Quantization method II** – In this method, all AC coefficients are quantized with the same quantizer, i.e., $Q_i = 2Q_S$, or equivalently $Q_i = Q_S \cdot W_i$ with $W_i = 2$ for $i = 1, \dots, N$.

Incorporating this information in (5.24) and (5.25), leads to the following rate-distortion model functions

$$\bar{R}(Q_S) = \frac{1}{N\alpha \log_2 e} \log_2 \left[\left(\frac{1}{Q_S^2} \right)^N \prod_{i=0}^{N-1} \left(\frac{\beta_i \varepsilon_i^2 \sigma_i^2}{W_i^2} \right) \right] = \frac{1}{\alpha \log_2 e} \log_2 \left[\frac{1}{Q_S^2} F \right] \quad (5.26)$$

with

$$F = \left[\prod_{i=0}^{N-1} \left(\frac{\beta_i \varepsilon_i^2 \sigma_i^2}{W_i^2} \right) \right]^{1/N}$$

and

$$\bar{D} = \frac{Q_S^2}{N} \sum_{i=0}^{N-1} \frac{W_i^2}{\beta_i} \quad (5.27)$$

As mentioned in [156], each DCT coefficient can be seen as having an effective variance of $\varepsilon_i^2 \sigma_i^2$ which can be lower than the weighted distortion $Q_S^2 W_i^2 / \beta_i$. In this circumstance, the maximum average distortion per coefficient is given by

$$\bar{D} = \frac{Q_S^2}{N} \sum_{i \in S_1} \frac{W_i^2}{\beta_i} + \frac{1}{N} \sum_{i \in S_2} \varepsilon_i^2 \sigma_i^2 \quad (5.28)$$

where $S_1 = \left\{ i \in \{1, \dots, N\} \wedge \left(\varepsilon_i^2 \sigma_i^2 \geq \frac{W_i^2 Q_S^2}{\beta_i} \right) \right\}$ and $S_2 = \{1, \dots, N\} \setminus S_1$.

Notice that the coefficients in S_2 are quantized to zero since their variance is lower than the weighted distortion, thus their maximum distortion is bounded by their effective variance, i.e., $\varepsilon_i^2 \sigma_i^2$. In this case, the rate-distortion function becomes

$$\bar{R}(Q_S) = \frac{1}{N\alpha \log_2 e} \log_2 \left[\left(\frac{1}{Q_S^2} \right)^{N_{S_2}} \prod_{i \in S_2} \left(\frac{\beta_i \varepsilon_i^2 \sigma_i^2}{W_i^2} \right) \right] = \frac{1}{\alpha \log_2 e} \log_2 \left[\left(\frac{1}{Q_S^2} \right)^{\frac{N_{S_2}}{N}} F \right] \quad (5.29)$$

where

$$F = \left[\prod_{i \in S_2} \left(\frac{\beta_i \varepsilon_i^2 \sigma_i^2}{W_i^2} \right) \right]^{1/N}$$

The direct use of (5.29) is cumbersome mainly due to the need to estimate α , β_i , and ε_i^2 from picture data, e.g., for each picture to be encoded. Moreover, for quantizers with dead-zone, α and β_i depend on the dead-zone threshold [156]. However, if the threshold varies linearly with Q_s , then (5.29) can be written as

$$\bar{R}(Q_s) = \frac{1}{\alpha(Q_s) \log_2 e} \log_2 \left(\frac{1}{Q_s^2} F(Q_s) \right) \quad (5.30)$$

Under the assumption that among consecutive coding time instants the picture characteristics remain approximately constant and that the α , β_i , and ε_i^2 parameters are approximately constant for small variations of Q_s , then $F(Q_s)$ and $\alpha(Q_s)$ can be estimated from previous coding results, making the model (5.30) very handy. According to [156], α is less picture-dependent than F , thus only F needs to be updated regularly. Notice that (5.30) can be written as

$$\bar{R}(Q_s) = \frac{2}{\alpha(Q_s) \log_2 e} \log_2 \left(\frac{1}{Q_s} \right) + \frac{1}{\alpha(Q_s) \log_2 e} \log_2 (F(Q_s)) \quad (5.31)$$

In real encoding situations, the actual rate-quantization function does not always resemble a logarithmic function as expressed by (5.31), notably, for a wide range of Q_s values. Figure 5.6 illustrates this situation showing the experimental rate-quantization data obtained by encoding one QCIF frame of the *Foreman* sequence in Intra and Inter modes, and the curve obtained by fitting (5.31) to the experimental data.

However, in the context of this Thesis, a careful analysis of (5.31) for $Q_s \in \{1, \dots, 31\}$ showed that $\log_2(1/Q_s)$ can be accurately approximated by the following parametric curve as illustrated in Figure 5.7:

$$f(Q_s) = \kappa \left(1/Q_s^\gamma - 1 \right) \quad (5.32)$$

where κ and γ are the curve parameters (in the case of Figure 5.7 $\kappa = 146$ and $\gamma = 0.01$).

As a result, (5.31) can be re-written as

$$\bar{R}(Q_s) = \alpha \frac{1}{Q_s^\gamma} + \beta \quad (5.33)$$

where α and β are the new model parameters.

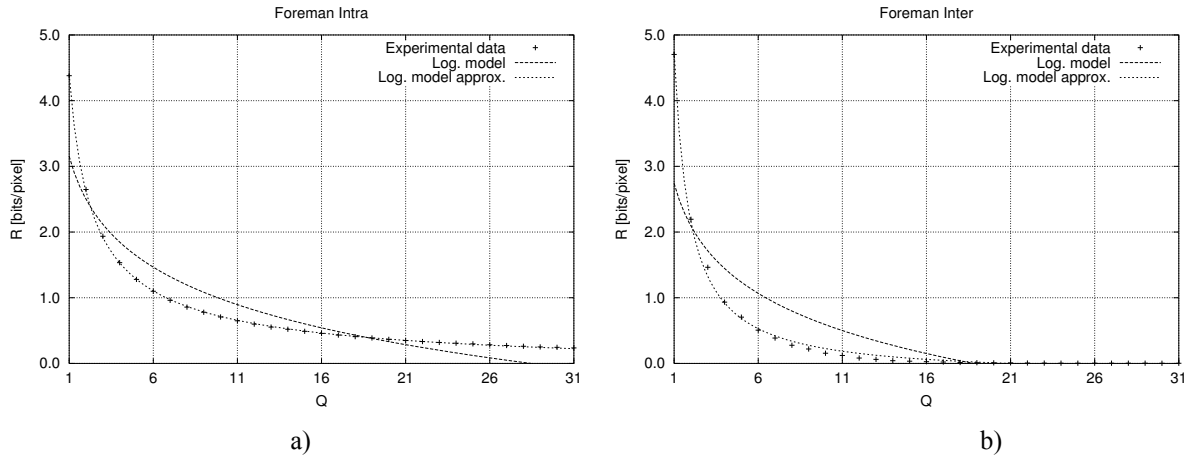


Figure 5.6 – Experimental and model rate-quantization characteristics for a frame of the Foreman sequence: a) Intra-coded; b) and Inter-coded

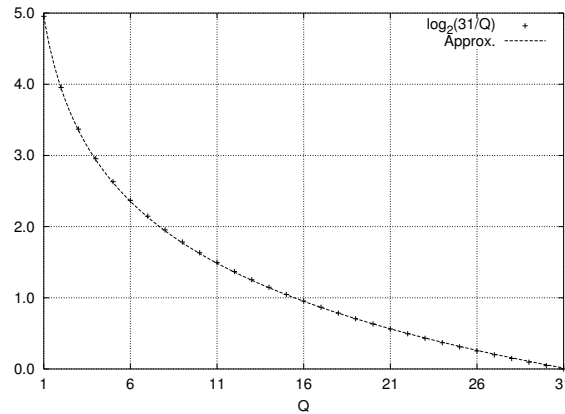


Figure 5.7 – Approximation of the logarithmic function

Notice, however, that it is difficult to cover a wide range of bit rates with a single model; thus it seems to make sense to devise a different model for different ranges of bit rates/operation conditions. Additionally, the following problems require careful consideration:

- In order to obtain almost constant quality, the quantization parameter should not change abruptly from picture to picture and from MB to MB.
- In order to not violate the VBV buffer constraints, the encoder has to keep the number of bits per picture between safe bounds.

Moreover, it is not possible to fully cover the nonstationarity and data dependency with just a simple model as the one defined by (5.33); thus some modeling errors cannot be avoided. In fact, in actual video coding, theoretical and experimental results do not always agree:

1. Ergodicity and stationarity assumptions not always hold, i.e., neither a single sample picture is representative of all pictures (i.e., of the random process), neither the picture characteristics remain invariant over time.
2. The human visual system is highly nonlinear and thus the subjective visual quality cannot be easily approximated by mean square error measures.

To compensate for this deviation between theoretical models and actual coding results it is necessary to develop some adaptation and compensation mechanisms. Control theory provides a very useful framework to solve these problems. This approach will be addressed in the

following chapter while this chapter considers the definition of simple rate-distortion models that can capture the rate and distortions characteristics of the video encoder.

5.2.5 Review of Rate-Distortion Modeling

In applying rate-distortion theory to video coding, the following approaches can be used separately or in conjunction:

- **Analytical modeling** – The encoding system is described by a set of mathematical expressions and, based on this description and the statistical model of the video source, a closed-form rate-distortion model is derived [159, 156, 116]. These models are usually derived in the form of $R(Q)$ and $D(Q)$ functions, where Q represents a quantization parameter. This way a straight formula relating the coding parameters and the output rate and distortion is obtained. In this approach, the source is typically modeled as Laplacian, Gaussian, or Generalized Gaussian [160].

As mentioned in the previous section, the main drawback of analytical rate-distortion modeling is the fact that in actual video coding, theoretical and experimental results do not always agree; thus large modeling errors can be obtained if the source statistics deviate from the theoretical assumption.

- **Empirical modeling** – In this case, the rate-distortion characteristics are estimated from observed coding results using curve-fitting techniques. This can be done prior to the real encoding of each picture through the analysis of multiple encoding trials – *multiple encoding estimation approach* [161, 160], or based on the encoding results of previous coding units – *delayed estimation approach* [101]. The multiple encoding approach is mathematically more accurate but also computationally very intensive, and is typically not well suited for real-time encoding. The main drawback of the delayed estimation approach is that it may lead to large control errors, notably at scene changes in low bit rate encoding. In order to reduce the computational load of the multiple encoding approach, it is possible to use only a sub-set of the coding parameters with the missing points obtained by interpolation.

This section reviews some rate-distortion modeling approaches for DCT-based video coding identifying their main strengths and drawbacks. This analysis is organized starting with less complex models and ending with more complex models.

MPEG-2 VIDEO TM5 HYPERBOLIC RATE-DISTORTION MODEL

The TM5 rate control algorithm [134], developed in the context of the MPEG-2 Video standard [10], assumes a simple rate-distortion model for each picture coding type (I, P or B) in the form of a rate-quantization hyperbolic function that can be seen as a simplified version of (5.33), where $\alpha = X_t$, $\beta = 0$ and $\gamma = 1.0$, i.e.,

$$R(Q) = X_t \frac{1}{Q} \quad (5.34)$$

Each model parameter, X_t , for each picture coding type t , $t \in \{I, P, B\}$, is designated as a **global complexity measure** for the corresponding picture type. Therefore in the TM5 algorithm the picture complexity is linked to the number of bits produced by the encoder for a given average quantization parameter. Notice that the same complexity measure is used for the whole picture.

As can be immediately inferred from (5.34), the higher the picture complexity the higher the number of bits produced when encoding the picture with a given average quantization parameter.

The model parameters, X_t , are updated at the end of each encoding time instant based on the generated bits for the just encoded picture and the average MB quantization parameter as

$$X_t = S_t Q_t$$

where S_t is the number of bits spent in the encoding of the given picture and Q_t the average MB quantization parameter.

In the TM5 algorithm, the rate-distortion model is used not to compute directly the coding parameters but to distribute the target number of bits for each Group of Pictures (GOP), R_t , among the different pictures in the GOP, according to the picture coding type, at the bit allocation stage. Before encoding each picture, the bit allocations for each picture type are updated based on the updated complexity measures of the corresponding picture coding type and the remaining number of bits to encode the GOP as follows

$$T_I = \frac{R}{1 + N_P \frac{X_P}{X_I} \cdot \frac{1}{K_P} + N_B \frac{X_B}{X_I} \cdot \frac{1}{K_B}}, T_P = \frac{R}{N_P + N_B \frac{X_B}{X_P} \cdot \frac{K_P}{K_B}}, \text{ and } T_B = \frac{R}{N_P \frac{X_P}{X_B} \cdot \frac{K_B}{K_P} + N_B} \quad (5.35)$$

where T_I , T_P , and T_B are the target number of bits to encode the corresponding picture type; R is the remaining number of bits to encode the current GOP; N_P and N_B are, respectively, the remaining number of P and B pictures to encode in the GOP; and K_P and K_B are adjustment constants (in [134] $K_P = 1.0$ and $K_B = 1.4$).

The computation of the quantization parameter to encode each MB uses a feedback approach based on virtual buffers (one virtual buffer for each picture coding type) and quantization adjustment based on a MB activity measure.

The main advantage of this algorithm resides on its simplicity and low complexity. Its main drawbacks are poor buffer control and target bit rate matching, notably for low bit rate encoding. Moreover, since the model parameters are estimated from past encoding statistics the TM5 algorithm does not handle scene changes efficiently.

MPEG-4 VISUAL QUADRATIC RATE-DISTORTION MODEL

Chiang and Zhang [101] and Lee *et al* [104] assume that the video source has a Laplacian distribution, i.e., with a probability density function (pdf) of the following form

$$p(x) = \frac{\alpha}{2} e^{-\alpha|x|}, \quad -\infty < x < \infty \quad (5.36)$$

and use a distortion measure given by the magnitude of the reconstruction error, i.e.,

$$D(x, \tilde{x}) = |x - \tilde{x}|$$

where x is the original sample and \tilde{x} the corresponding reconstructed sample.

Under these assumptions, there is a closed-form solution for the rate-distortion function derived in [147]

$$R(D) = \ln\left(\frac{1}{\alpha D}\right) \quad (5.37)$$

where $0 < D < \frac{1}{\alpha}$, i.e., $D_{\min} = 0$ and $D_{\max} = \frac{1}{\alpha}$.

Expanding (5.37) into a Taylor's series leads to

$$\begin{aligned} R(D) &= \left(\frac{1}{\alpha D} - 1\right) - \frac{1}{2}\left(\frac{1}{\alpha D} - 1\right)^2 + R_3(D) \\ &= -\frac{3}{2} + \frac{2}{\alpha}D^{-1} - \frac{1}{2\alpha^2}D^{-2} + R_3(D) \end{aligned}$$

where $R_3(D)$ is the third order remainder of the series.

Based on the above expression, Chiang and Zhang [101] proposed a quadratic rate-distortion model in the form of a rate-quantization function formulated as

$$R(Q) = A_1 \frac{1}{Q} + A_2 \frac{1}{Q^2} \quad (5.38)$$

where Q is the quantization parameter and A_1 and A_2 are the model parameters.

According to the authors, the reduction in the modeling error does not significantly diminishes if the order of the model is increased beyond two, thus making it useless to use a higher order model.

This model, suggested by MPEG-4 Visual in its informative Annex on rate control [29], is used to estimate the quantization parameter for each VOP before encoding it, i.e., given a certain target bit allocation, R_T , to encode the VOP at hand, the model is used to compute a single quantization parameter, Q , for all MBs in the VOP that, according to the model, leads to the pre-defined number of bits to encode that VOP. This is done solving the following equation,

$$R_T = A_1 \frac{1}{Q} + A_2 \frac{1}{Q^2} \quad (5.39)$$

The model parameters are re-estimated after the encoding of each VOP. For this purpose, the encoder stores the number of bits, R_i , and the quantization parameter, Q_i , used to encode all MBs in VOP i and based on the data collected for a set of N previously encoded VOPs estimates the model parameters A_1 and A_2 using least squares estimation.

For this case, there is a simple way to compute the model parameters, through the minimization of the model error, i.e.,

$$\chi^2 = \sum_{i=1}^N \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right)^2 \quad (5.40)$$

This minimization leads to the following equations

$$\begin{cases} \frac{\partial \chi^2}{\partial A_1} = 0 \\ \frac{\partial \chi^2}{\partial A_2} = 0 \end{cases} \Leftrightarrow \begin{cases} -2 \sum_{i=1}^N \frac{1}{Q_i} \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right) = 0 \\ -2 \sum_{i=1}^N \frac{1}{Q_i^2} \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right) = 0 \end{cases} \quad (5.41)$$

with the following solution

$$\begin{cases} A_1 = \frac{\left(\sum_{i=1}^N \frac{R_i}{Q_i} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^4} \right) - \left(\sum_{i=1}^N \frac{R_i}{Q_i^2} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^3} \right)}{\left(\sum_{i=1}^N \frac{1}{Q_i^4} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) - \left(\sum_{i=1}^N \frac{1}{Q_i^3} \right)^2} \\ A_2 = \frac{\left(\sum_{i=1}^N \frac{R_i}{Q_i^2} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) - \left(\sum_{i=1}^N \frac{R_i}{Q_i} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^3} \right)}{\left(\sum_{i=1}^N \frac{1}{Q_i^4} \right) \left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) - \left(\sum_{i=1}^N \frac{1}{Q_i^3} \right)^2} \end{cases} \quad (5.42)$$

In [101] and [104], however, the authors use a modified version of (5.39), i.e.,

$$R_i Q = A_1 + A_2 \frac{1}{Q} \quad (5.43)$$

which leads to a model error

$$\chi^2 = \sum_{i=1}^N \left(Q_i R_i - A_1 - A_2 \frac{1}{Q_i} \right)^2 = \sum_{i=1}^N Q_i^2 \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right)^2 \quad (5.44)$$

The minimization of (5.44) leads to the following equations

$$\begin{cases} \frac{\partial \chi^2}{\partial A_1} = 0 \\ \frac{\partial \chi^2}{\partial A_2} = 0 \end{cases} \Leftrightarrow \begin{cases} -2 \sum_{i=1}^N \left(Q_i R_i - A_1 - A_2 \frac{1}{Q_i} \right) = 0 \\ -2 \sum_{i=1}^N \frac{1}{Q_i} \left(Q_i R_i - A_1 - A_2 \frac{1}{Q_i} \right) = 0 \end{cases} \Leftrightarrow \begin{cases} -2 \sum_{i=1}^N Q_i \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right) = 0 \\ -2 \sum_{i=1}^N \left(R_i - A_1 \frac{1}{Q_i} - A_2 \frac{1}{Q_i^2} \right) = 0 \end{cases} \quad (5.45)$$

with the following solution

$$\begin{cases} A_1 = \frac{\left(\sum_{i=1}^N R_i Q_i \right) \left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) - \left(\sum_{i=1}^N R_i \right) \left(\sum_{i=1}^N \frac{1}{Q_i} \right)}{\left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) N - \left(\sum_{i=1}^N \frac{1}{Q_i} \right)^2} \\ A_2 = \frac{\left(\sum_{i=1}^N R_i \right) N - \left(\sum_{i=1}^N R_i Q_i \right) \left(\sum_{i=1}^N \frac{1}{Q_i} \right)}{\left(\sum_{i=1}^N \frac{1}{Q_i^2} \right) N - \left(\sum_{i=1}^N \frac{1}{Q_i} \right)^2} \end{cases} \quad (5.46)$$

In the context of this Thesis, a careful analysis of the model parameters estimation proposed in

[101] and [104], revealed that the use of (5.43) instead of (5.39) leads to a higher modeling error. This is because the model errors in (5.44) are weighted by Q_i^2 relatively to (5.40) resulting that the model errors for higher quantization parameter values have higher weight in the estimation of A_1 and A_2 than the lower values. Thus, the estimation of the model parameters through (5.46) instead of (5.42) bias the model in favor of the data points corresponding to the higher quantization parameter values, leading to a model that has a higher modeling error for the lower quantization steps and a lower modeling error for the higher quantization steps. This situation is illustrated in Figure 5.8 for a frame of the *Foreman* sequence encoded in Intra and Inter modes using the MPEG-4 reference software video encoder [32]. In this case, the model parameters obtained through (5.42) for the Intra and Inter cases lead to a root mean square (RMS) value of the deviation of the data from the model of $\sigma = 0.045$ and $\sigma = 0.147$, respectively for the Intra and Inter case, while the model parameters obtained through (5.46) lead to a RMS of $\sigma = 0.198$ and $\sigma = 0.945$, where σ is the square root of the reduced χ^2 , i.e., the square root of the model error divided by the number of data points minus the number of parameters in the model, that is

$$\sigma = \sqrt{\chi^2 / (N - 2)} \quad (5.47)$$

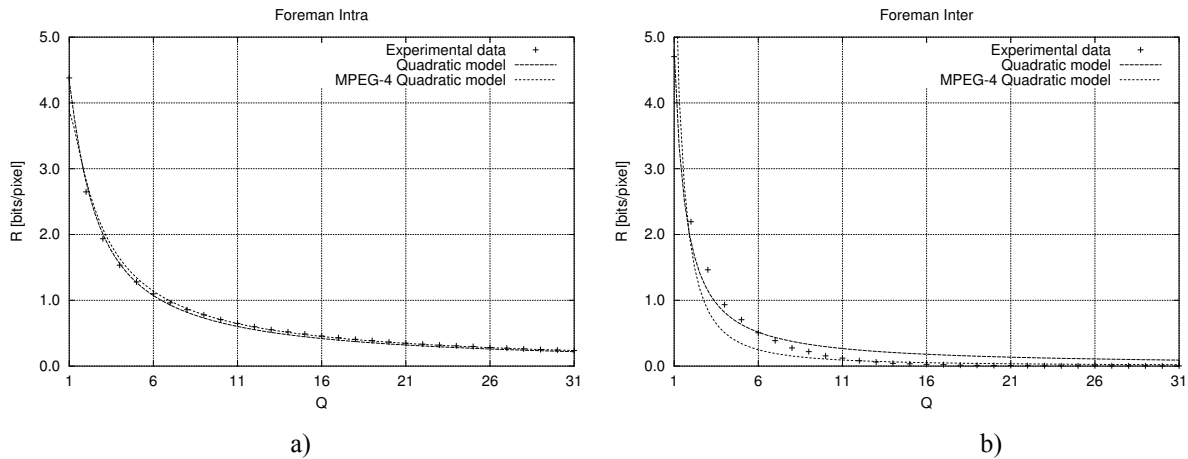


Figure 5.8 – Quadratic rate-distortion models for a frame of the *Foreman* sequence:
a) Intra-coded; b) Inter-coded

As can be seen in Figure 5.8, the quadratic rate-distortion model approximates well the experimental data. However, in real-time encoding only the encoding statistics of previous encoded pictures are typically available, thus the model must be estimated from a subset of rate-distortions points. In this case, the quadratic model may lead to inaccurate results, as illustrated in Figure 5.9. In Figure 5.9, the quadratic model and the hyperbolic model (5.34) are estimated using only a subset of the experimental data, i.e., $8 \leq Q \leq 24$. As can be seen from Figure 5.9, both models approximate well the experimental curve for the same range of quantization parameter values used during the estimation; however the quadratic model for quantization parameter values outside this range exhibits a great deviation from the experimental data.

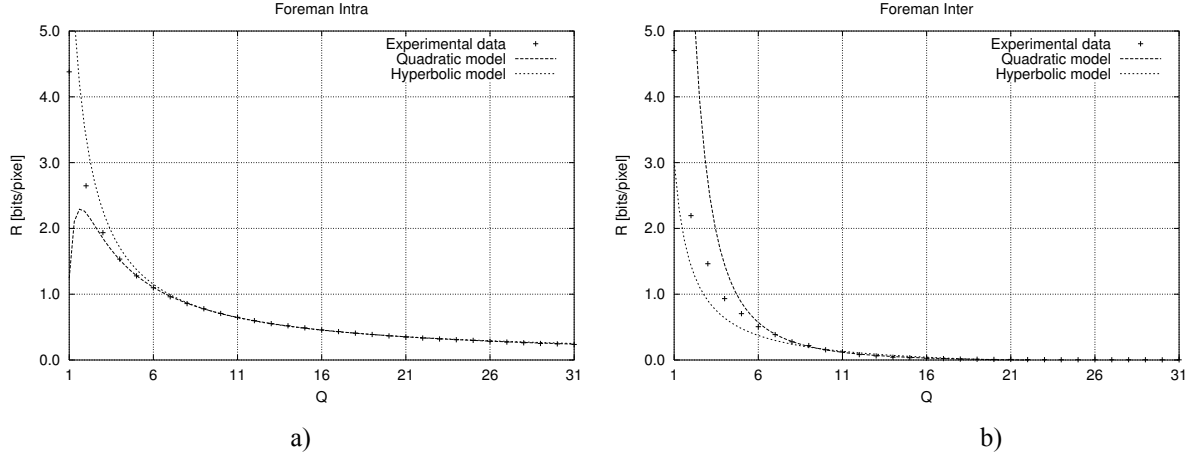


Figure 5.9 – Estimation of the quadratic and hyperbolic rate-distortion models for a frame of the Foreman sequence using a sub-set of the experimental data points ($8 \leq Q \leq 24$):

a) Intra-coded; b) Inter-coded

MPEG-4 VISUAL MB RATE-DISTORTION MODEL

In the ITU-T H.263 [8] test model TMN8 [162] and the MPEG-4 video verification model VM8 [108], a simplified quadratic rate-distortion model is suggested for MB-level rate control, where the number of bits generated by a given MB as a function of the quantizer parameter is modeled as

$$R_i(Q) = A_i \frac{1}{Q^2} \sigma_i^2 \quad (5.48)$$

where A_i is the model parameter and σ_i is a MB complexity measure computed from the picture data. In [162], σ_i is given by the square root of the prediction error variance, while in [108] σ_i is given by the mean absolute difference (MAD) between the MB and its prediction.

The target number of bits for MB i , R_{T_i} , is a fraction of the target number of bits for the picture/VOP, R_T , based on the MB complexity measure, σ_i , the MB perceptual weight, α_i , and the number of bits already spent in the previous MBs of the current VOP, i.e.,

$$R_{T_i} = \frac{\alpha_i \sigma_i}{\sum_{k=i}^N \alpha_k \sigma_k} \left(R_T - \sum_{k=1}^{i-1} S_k \right) \quad (5.49)$$

where S_k is the number of bits generated by MB k .

Based on (5.48) and (5.49), the quantization parameter to encode MB i is given by

$$Q_i = \sqrt{A_i \frac{1}{R_T - \sum_{k=1}^{i-1} S_k} \frac{\sigma_i}{\alpha_i} \sum_{k=i}^N \alpha_k \sigma_k} \quad (5.50)$$

This model provides a feedback equation for the rate control algorithm, because the target number of bits for each MB and the model parameter are updated at the end of each MB encoding. If the encoder is producing too many bits for each MB, A_i will tend to increase and consequently the quantization parameter for future MBs will tend to increase also, thus reducing the number of bits produced. Similarly, if the encoder is producing too few bits for

each MB, A_i will tend to decrease and consequently the quantization parameter for future MBs will tend to decrease also, thus increasing the number of bits produced. The estimation of A_i is however a very sensitive step of the algorithm. If A_i changes swiftly, based only on a few MB coding results, it can lead to a great instability in the computation of the quantization parameter; on the other hand, if A_i changes slowly it cannot handle efficiently the model errors. This type of problems is considered in Chapter 6 through adequate compensation mechanisms for the MB quantization step computation.

When applied to the full picture, the simplified MB quadratic model (5.48) leads to a high model error as can be seen in Figure 5.10, notably when compared with the hyperbolic (5.34) or the quadratic model (5.39). In fact, the actual rate-quantization functions for the experimental data presented in Figure 5.10 vary linearly with $1/Q^\gamma$, with $\gamma \approx 0.8$ for the Intra case and $\gamma \approx 1.3$ for the Inter case. This explains the larger deviations in Figure 5.10a relatively to Figure 5.10b, since the simplified quadratic model always assumes a linear variation with $1/Q^2$.

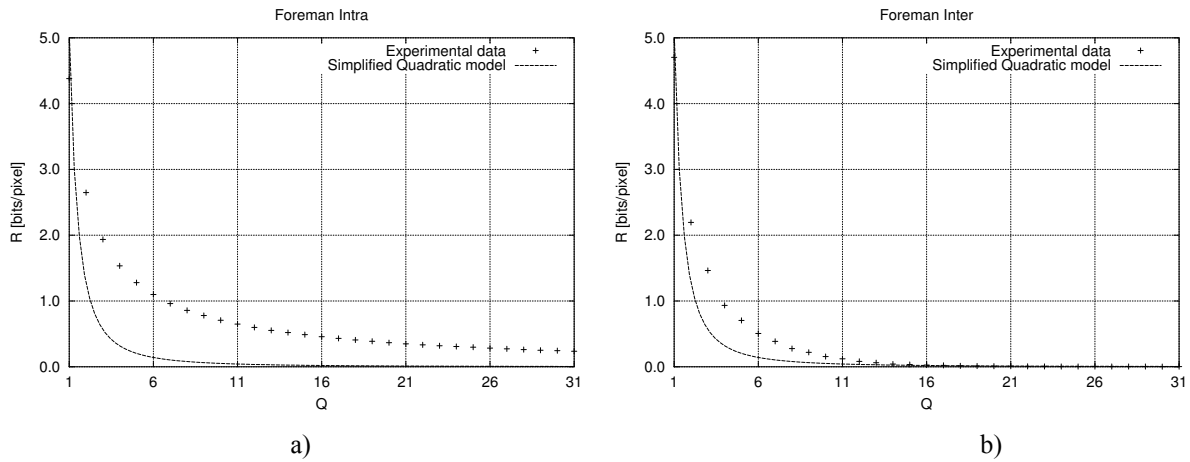


Figure 5.10 – Estimation of the simplified MB quadratic rate-distortion model for a frame of the Foreman sequence: a) Intra-coded; b) Inter-coded

CUBIC SPLINE RATE-DISTORTION MODEL

In the context of MPEG-2 Video encoding, some authors claim that closed-form rate-distortion models such as negative exponential models are not sufficiently accurate to model practical video encoders [163, 164, 160], and propose a cubic spline approximation model. This model is used in a context of multiple-pass encoding to estimate the rate-distortion characteristics of the source. Since not all possible quantizer values are tried, the authors set this way a trade-off between measuring and modeling the rate-distortion characteristics of the video encoder, i.e., the points not measured are interpolated through a curve fitting method. In this approach, the main goal is to obtain a low relative model mismatch error, ε , defined as

$$\varepsilon = \left| \frac{\hat{x} - x}{x} \right|$$

where x is the original measured value (rate or distortion), and \hat{x} is the corresponding estimated value according to the model, obtained by interpolation.

The $R(Q)$ and $D(Q)$ models for Intra coded pictures are computed by encoding each picture

and measuring the rate and distortion data for a set of M quantizers – control points. Between each two consecutive control points, a cubic function f_i is computed as

$$f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$$

In order to find the parameter values a_i , b_i , c_i , and d_i , the following constraints are imposed:

1. The interpolation function should take the same value as the measured data at each control point, i.e.,

$$f_i(x_i) = y_i \quad \text{and} \quad f_i(x_{i+1}) = y_{i+1}$$

2. The first-order derivative should be continuous at the control points, i.e.,

$$f'_{i-1}(x_i) = f'_i(x_i)$$

where

$$f'_i(x_i) = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} \quad \text{and} \quad f'_i(x_{i+1}) = \frac{y_{i+2} - y_i}{x_{i+2} - x_i}$$

In order to capture the exponential-decay property of typical rate-distortion data, a set of unequally spaced quantizers is used (in [163, 164, 160] six different quantizer values are used, i.e., $\{3, 5, 8, 13, 21, 31\}$).

For Inter coded pictures, due to picture prediction, the rate-distortion characteristics depend also on the coding quality of the reference pictures. In this case, the rate-distortion characteristics become multi-dimensional, depending not only on the coding parameters of the current picture, but also on the coding parameters of previous reference pictures. Even for a reduced number of quantizer values, to account for all possible combinations of reference picture and current picture quantizers requires a large number of encoding steps for each picture.

In order to reduce the computational cost of estimating the rate-distortion characteristics for Inter-coded pictures, a different distortion model is used that takes into account Inter-picture dependency, modeling the distortion of a P-picture as a linear function of the distortion of the reference I-picture, i.e.,

$$D_P(Q_I, Q_P) = \begin{cases} \alpha - \beta [D_I(Q_P) - D_I(Q_I)] & \Leftarrow Q_I \leq Q_P \\ \alpha & \Leftarrow Q_I > Q_P \end{cases} \quad (5.51)$$

where Q_I and $D_I(\cdot)$ are, respectively, the quantizer and the distortion of the reference I-picture, Q_P and $D_P(\cdot)$ the quantizer and the distortion of the Inter-coded picture, and α and β the model parameters computed using a set of pairs of quantizer values (Q_I, Q_P) (in [163, 164, 160] $Q_I \in \{5, 13\}$ and $Q_P \in \{3, 5, 8, 13, 21, 31\}$, i.e., twelve control points are used $\{(5,3), (5,5), \dots, (13,31)\}$).

The rationale for (5.51) is that the picture prediction error tends to be smaller when the reference picture distortion is smaller and consequently the predicted picture distortion and rate will also be smaller. When the reference picture distortion is higher, the predicted frame will also exhibit a higher distortion and rate and more MBs will be coded in Intra mode, decreasing the dependency of the coded frame on its reference.

According to [163], the rate characteristics exhibit low interframe dependency, thus the rate

model proposed in [163] uses the following linear piecewise function

$$R_p(Q_l, Q_p) = \begin{cases} R_p(Q_l, Q_p) & \Leftarrow Q_l \leq Q_1 \\ \frac{R_p(Q_l, Q_p)(D_l(Q_2) - D_l(Q_l)) + R_p(Q_2, Q_p)(D_l(Q_l) - D_l(Q_1))}{D_l(Q_1) - D_l(Q_2)} & \Leftarrow Q_1 < Q_l < Q_2 \\ R_p(Q_2, Q_p) & \Leftarrow Q_l \geq Q_2 \end{cases} \quad (5.52)$$

where Q_1 and Q_2 are two control points for the reference picture.

It is important to notice that the rate-distortion models defined by (5.51) and (5.52) are typically used for offline video coding aiming at reducing the computational cost of measuring all possible operational rate-distortion points for the given coding unit, which can be the full sequence or just a sub-set of pictures⁴.

In [160], the above rate and distortion models (5.51) and (5.52) are used to minimize the average picture distortion over each GOP of a sequence or alternatively to minimize the distortion variation. For a set of admissible quantizers $Q = \{1, 2, \dots, 31\}$ and a maximum buffer occupancy, B_{\max} , minimizing the average picture distortion over a GOP of length N is equivalent to find the vector $\mathbf{Q}^* = (Q_1^*, Q_2^*, \dots, Q_N^*)^T$, with $Q_i^* \in Q$ for $i = 1, 2, \dots, N$, such that

$$\mathbf{Q}^* = \arg \min_{\mathbf{Q} \in Q^N} \frac{1}{N} \sum_{i=1}^N D_i(\mathbf{Q})$$

subject to $B(i, \mathbf{Q}) \leq B_{\max}$, $i = 1, 2, \dots, N$, with $B(i, \mathbf{Q}) = \max\left(B(i-1, \mathbf{Q}) + R_i(\mathbf{Q}) - \frac{R}{F}, 0\right)$,

where $R_i(\cdot)$ is the rate function, R is the channel bit rate, and F is the picture rate.

Similarly, minimizing the distortion variation over a GOP of length N is equivalent to find for each possible control point, $Q_1 \in Q$, the vector \mathbf{Q}^* , such that

$$\mathbf{Q}^* = \arg \min_{\mathbf{Q} \in Q^{N-1}} \frac{1}{N} \sum_{i=2}^N |D_i(Q_i) - D_{i-1}(Q_{i-1})|$$

subject to $B(i, \mathbf{Q}) \leq B_{\max}$, $i = 1, 2, \dots, N$, where Q_1 can be selected to minimize the average distortion achieved by \mathbf{Q}^* .

The major advantage of these empirical rate-distortion models, obtained by polynomial interpolation, over analytical models, described by closed-form solutions, is the low model mismatch error. However, they require a considerable number of encoding passes per picture, making them less suitable for real-time encoding. In this model, the pre-analysis step is performed by re-encoding several times the same picture.

LOGARITHMIC RATE-DISTORTION MODEL

In [161], the authors observed that the $R(Q)$ characteristics for MPEG-1 and MPEG-2 Video encoding resemble the rate-distortion characteristic of Gaussian random variables suggesting in a first approach that they may be modeled by

⁴ In [163, 164, 160], the coding unit is the Group of Pictures (GOP).

$$R(Q) = \alpha \left(\log \frac{1}{Q} \right) + \beta \quad (5.53)$$

where α and β are the model parameters.

However, the authors recognize that the actual $R(Q)$ curves diverge from real Gaussian rate-distortion curves, thus proposing a refined model of the form

$$R(Q) = \alpha \frac{1}{Q^\gamma} + \beta \quad (0 < \gamma \leq 2) \quad (5.54)$$

where α , β , and γ are the new model parameters.

Notice that (5.54) requires more complex parameter estimation techniques (e.g., nonlinear least squares) than (5.53) for which least squares estimation can be easily applied. Moreover, nonlinear least squares techniques are iterative and may not converge (the convergence of nonlinear least squares methods can often be greatly improved if the initial parameter values are close to the best-fit parameters).

In [161], the authors report that although the model (5.54) fits well the actual rate-distortion curves, there is apparently no single model for all pictures in the sequence. Additionally the authors report that I-frames are characterized by $0.5 \leq \gamma \leq 1$ while P-frames are characterized by $0.5 \leq \gamma \leq 1.5$. The authors also point an alternative approach of fixing $\gamma = 1$ and locally adjust the parameters α and β for different ranges of Q .

In [161], the $R(Q)$ function is used in a context of multiple encoding steps. In a first step, the current frame is encoded with an initial quantization parameter $Q_1 = Q_0$; if the bit deviation from the target bit allocation, R_r , is greater than the pre-determined threshold ε than a new encoding step takes place with $Q_2 = Q_1 - \delta$ if $R_1 < R_r + \varepsilon$ or $Q_2 = Q_1 + \delta$ if $R_1 > R_r + \varepsilon$. After this encoding step an $R(Q)$ model is estimated using (Q_1, R_1) and (Q_2, R_2) which is then used to find the new quantization parameter $Q_T = R^{-1}(R_r)$. This process is repeated until $|R_i - R_r| < \varepsilon$.

The main advantage of this approach is that it can reduce the number of encoding steps in a multiple-encoding scenario with a close control on the deviation between the actual number of coded bits and the target, notably when compared with methods that use a pre-defined number of encoding steps. Its main drawback for real-time encoding is exactly the requirement of multiple encoding steps.

MULTI-EXPONENTIAL RATE-DISTORTION MODEL

In [165], the authors propose a rate-distortion model solely for Intra-coded frames, in the context of MPEG-2 Video encoding, where the typical decay of operational rate-distortion functions (see Figure 5.1) is modeled through exponential functions of the following form

$$R(Q) = A \exp(BQ) \quad (5.55)$$

Recognizing that it is not possible to model accurately the actual rate-distortion curves with just a single function like (5.55) for the range of possible quantizer values, the authors propose to model the actual rate-distortion functions as a sum of exponential functions, i.e.,

$$R(Q) = \sum_{i=1}^n A_n \exp(B_n Q) \quad (5.56)$$

It is important to notice that while the parameters in (5.55) can be easily estimated through linear least-squares fitting techniques, (5.56) requires more complex nonlinear fitting methods such as the Levenberg-Marquardt method [166], or the Prony method [167].

In [165], the authors report that the actual rate-distortion curves can be approximated by three exponential functions (i.e., using six parameters) where the first function models the rate-quantization behavior for large quantization parameter values ($Q \geq 10$), the second function models the rate-quantization behavior for small quantization parameter values ($Q < 10$), and the third function acts as a compensation for the estimation errors caused by the use of the first and second functions alone.

To estimate the model parameters, each picture is encoded for three different quantization parameter values ($Q = \{1, 10, 25\}$). Based on each pair of (Q, R) points, the authors estimate the parameters of each exponential function.

ρ -DOMAIN RATE-DISTORTION MODEL

In [116], the authors propose a source model where the rate function is given as a function of the percentage of zeros among the quantized DCT coefficients, notably

$$R(\rho) = \theta(1 - \rho) \quad (5.57)$$

where θ is the model parameter and ρ is the percentage of zeros.

Defining $h_l(x)$ and $h_p(x)$ as the histograms of the DCT coefficients, respectively, for Intra and Inter Coded MBs in a frame, an one-to-one mapping between ρ and the quantization parameter Q can be established, i.e.,

$$\rho(Q) = \frac{1}{M} \sum_{|x| < 2Q} h_l(x) + \frac{1}{M} \sum_{|x| < 2.5Q} h_p(x) \quad (5.58)$$

where M is the total number of DCT coefficients in the video frame (including zero and non-zero coefficients). The mapping of the rate curve between the Q domain and the ρ -domain is performed by table look-up and bi-linear interpolation. First, the rate function is estimated in the ρ -domain and then mapped into the Q domain.

In [168], the authors introduce also a distortion model in the ρ -domain assuming that the DCT coefficients have a Laplacian distribution. Recognizing that the theoretical distortion model is highly nonlinear and complex, the following exponential approximation is used

$$D(\rho) = \sigma^2 e^{-\alpha(1-\rho)} \quad (5.59)$$

where α is a constant (typically $10 \leq \alpha \leq 20$) and σ^2 is the DCT coefficient variance. For a Laplacian distribution with an amplitude probability density function (PDF) of the form

$$p(x) = \frac{\lambda}{2} e^{-\lambda|x|}$$

the relationship between σ^2 and λ is given by $\sigma^2 = 2/\lambda^2$.

In [116], the model described by (5.57) and (5.58) is proposed for rate control purposes.

Before encoding, a rate-quantization function, $R(Q)$, is estimated for each picture and based on this function a target quantization parameter, Q_T , is computed such that $Q_T = R^{-1}(R_T)$, where R_T is the target number of bits to encode the picture.

Since Q_T is a real value and the quantization parameter must be an integer between 1 and 31, coding the complete picture with the rounded value of Q_T may lead to a significant deviation from the target number of bits, R_T ; thus the authors propose a simple approach where the quantization parameter to encode a given MB is selected randomly between two possible values, $Q_T^+ = \lceil Q_T \rceil^5$ and $Q_T^- = \lfloor Q_T \rfloor^6$, in a way that the average quantization parameter approaches Q_T .

Let $\gamma = Q_T - Q_T^-$ and ϕ be an uniformly distributed random variable in $[0,1]$. To encode each MB, a sample of ϕ is generated and the MB is encoded with a quantization parameter, Q , obtained by the following rule

$$Q = \begin{cases} Q_T^+ & \text{if } \phi < \gamma \\ Q_T^- & \text{if } \phi \geq \gamma \end{cases}$$

This way, the fraction of MBs encoded with Q_T^+ will be approximately γ , and the fraction of MBs encoded with Q_T^- will be approximately $1-\gamma$; since $Q_T^+ = Q_T + 1 - \gamma$ and $Q_T^- = Q_T - \gamma$, this leads to an average quantization parameter close to Q_T , i.e.,

$$\bar{Q} = \gamma Q_T^+ + (1-\gamma)Q_T^- = \gamma(Q_T + 1 - \gamma) + (1-\gamma)(Q_T - \gamma) = Q_T$$

As a result, the actual number of generated bits will be close to R_T , provided that the rate-quantization model is accurate.

According to the authors, the main advantage of this rate-distortion modeling approach is the low mismatch error between the model and the actual coding results. The main drawback is the complexity in the estimation of the model since a table lookup needs to be built for each coding unit in order to obtain the $\rho(Q)$ function. The $R(\rho)$ model parameter can be obtained either by multiple encoding steps or through the encoding results of previous encoding time instants.

5.3 Proposal of Rate and Distortion Models for Low-Delay Video Encoding

In the context of object-based video coding, notably in MPEG-4 video encoding [29], rate and distortion models characterize the relation between the average number of bits/pixel to code a given VOP, the average VOP distortion, and the relevant coding parameters. These models, usually defined in terms of rate-quantization, distortion-quantization, and rate-distortion functions, can play a very important role in real-time video encoding, since they can be used

⁵ $\lceil x \rceil$ represents the smallest integer greater or equal than x .

⁶ $\lfloor x \rfloor$ represents the highest integer lower or equal than x .

to obtain near optimal operation performance in terms of the rate-distortion trade-off without the drawbacks of having to encode multiple times the same VOP to find the best combination of coding parameters.

For constant bit rate video encoding, the rate-quantization models are useful to compute the quantization parameters to encode each VOP given the bit allocation for the corresponding time instant. Similarly, for approximately constant quality encoding, the distortion-quantization models allow to compute the VOP quantization parameters that lead to the target average VOP distortion. Finally, in a multiple video objects encoding scenario, where the rate control mechanism must keep the quality among the several VOs approximately constant, rate-distortion models can be used to guide the bit allocation module in order to produce a bit allocation for the various VOs in the scene that leads to a similar quality.

As mentioned in Section 5.2.4, since theoretical and experimental coding results do not always agree, it is important to use rate and distortion models that simultaneously grab the main characteristics of the source data and the coding algorithm, that are simple to obtain, and finally, that are robust to modeling errors, i.e., easily adaptable in non-stationary conditions. In this context, this section proposes a set of rate and distortion models with the above characteristics particularly adequate for low-delay video encoding.

As pointed out in [169], a fundamental lesson of rate-distortion theory is that better performance can be achieved by using a collection of simple models instead of a single all-encompassing model. This principle means that there are typically performance benefits by separating sources of uncertainty and designing a global model as a collection of simple models rather than a single, more complex model. Therefore, when using a multiple-model approach, the following questions naturally arise:

- How many models should be included?
- Which models should be used?
- What are the price and payoff of the multiple-model approach?

Nevertheless, if the main issue is the modeling error than typically the global model requires several parameters since it is difficult to represent the high-level statistics of the data with a model using a reduced number of parameters. The higher the number of parameters in the model, the higher the number of points needed to estimate the RD model, e.g., the quadratic RD model (5.38) needs at least two distinct (Q, R) points to estimate its parameters, while the hyperbolic model (5.34) needs only one point. Some methods (e.g., [160, 165]) use a pre-defined number of (Q, R) control points for estimating the RD model. These methods involve multiple encoding steps (quantization and VLC encoding) since the control points are collected before the real encoding of the data.

Applying to video compression the principle of using a global model that is formed by multiple simpler models requires identifying the main characteristics affecting the RD models. The characteristics/parameters that immediately come out are:

- The coding mode, e.g., Intra, Inter, or Bidirectional.
- The target average compression ratio/average distortion.
- The statistical source data model (type of model), e.g., Laplacian or Gaussian, and the model parameters, e.g., mean and variance.

Each coding mode leads typically to coded data with different rate and distortion

characteristics, as illustrated in Figure 5.11, where the experimental rate-quantization, distortion-quantization, and rate-distortion functions for one frame of the *Foreman* and *Stefan* sequences encoded in Intra and Inter modes are represented (in the case of Inter coding three different values for the average reference picture quantization parameter, Q_{ref} , were used).

From Figure 5.11, it is clear that for the Intra mode the rate and distortion functions exhibit a similar variation over the whole range of the quantization parameter, i.e., strictly decreasing in the case of the rate-quantization and rate-distortion functions and strictly increasing in the case of the distortion-quantization function. For Inter coding, however, the rate and the distortion functions tend to saturate, notably for $Q \gg Q_{ref}$, indicating a clear dependence of the current picture rate and distortion functions not only on the quantization parameter of the current picture but also on the quantization parameter of its reference picture. This is shown as different rate and distortion curves for different Q_{ref} values.

It is important to notice that in a rate-distortion framework, the number of bits used to encode a given VOP, R_{VOP} , can be divided into two components: one that depends on the quantization parameter(s) used to encode the VOP – *quantizer dependent rate* – $R(Q)$, and another auxiliary component that can be considered quantizer independent – *quantizer independent rate* – R_{aux} , i.e.,

$$R_{VOP} = R(Q) + R_{aux} \quad (5.60)$$

The quantizer dependent rate represents the bits used to encode the quantized DCT coefficients of the luminance and chrominance, i.e.,

$$R(Q) = R_Y(Q) + R_{UV}(Q) \quad (5.61)$$

In the case on Intra coding, the quantizer independent rate includes the header bits (e.g., in the case of MPEG-4 Video [29], the VOP header and the coded block patterns for the luminance and chrominance) as well as the bits used to encode the shape information in the case of arbitrarily shaped VOs⁷, i.e.,

$$R_{aux} = R_{header} + R_{shape} \quad (5.62)$$

In the case of Inter coding, the quantizer independent rate includes additionally the bits used to encode the motion information that is also quantizer independent, i.e.,

$$R_{aux} = R_{header} + R_{shape} + R_{motion} \quad (5.63)$$

Figure 5.12 illustrates the different rate components for the *Foreman* sequence and one arbitrarily shaped VO of the *Stefan* sequence – the *Player* – encoded in Intra and Inter modes.

In terms of rate control, the *quantizer independent rate* components can be easily estimated from the previous encoding time instants, or, as in the case of the shape and motion bits, they can be obtained before texture encoding during a pre-analysis step, where motion estimation and shape coding are performed [26]. Therefore, in terms of rate-distortion modeling, the *quantizer dependent rate* requires a more careful analysis. This analysis will be done separately for the Intra and Inter coding modes, since, as mentioned above, these two coding modes exhibit different characteristics in terms of its rate and distortion functions.

⁷ In this case, it is assumed that the shape information is losslessly encoded.

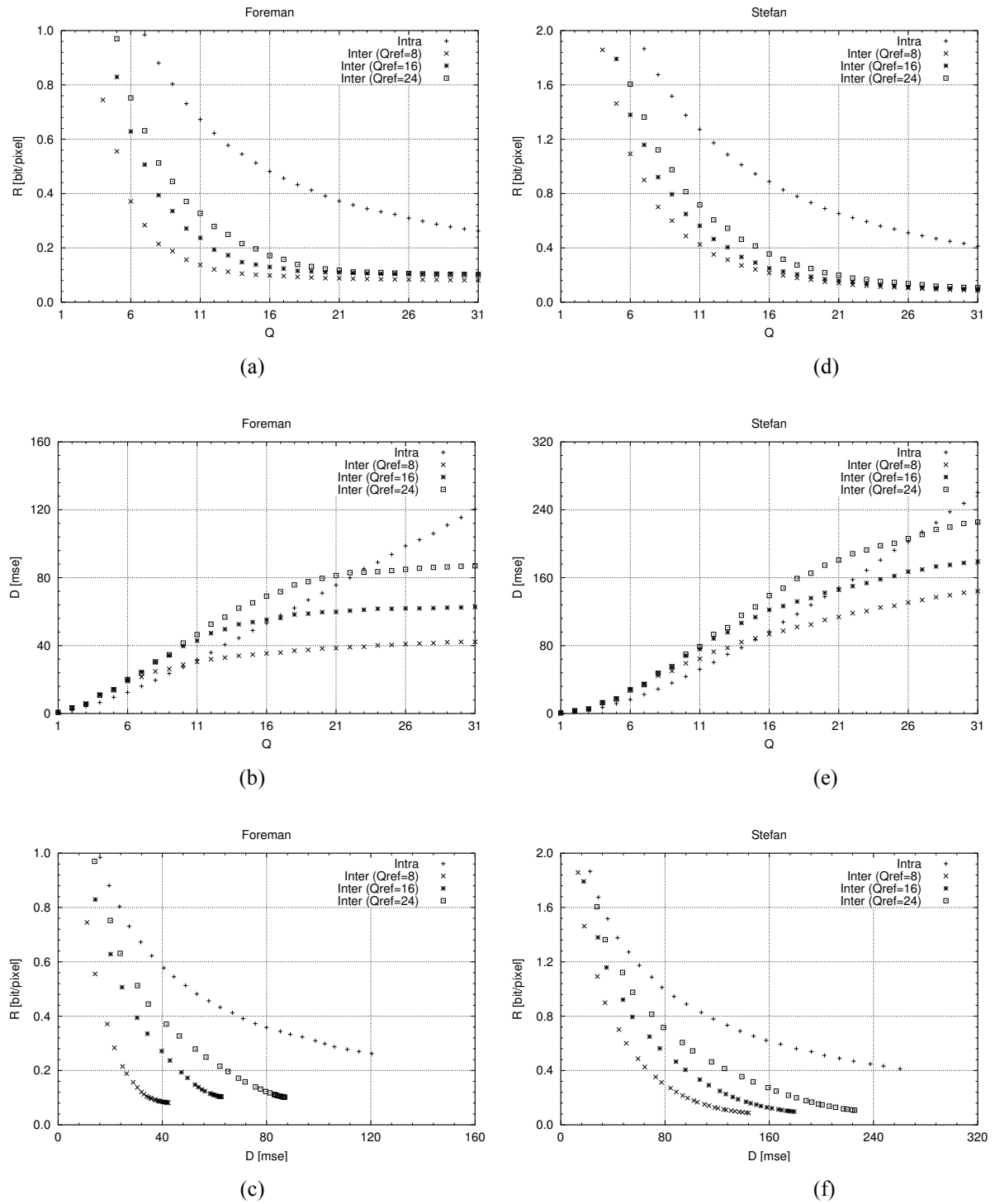


Figure 5.11 – Intra and Inter rate and distortion functions for a frame of the Foreman and Stefan sequences: a) Foreman rate-quantization; b) Foreman distortion-quantization; c) Foreman rate-distortion; d) Stefan rate-quantization; e) Stefan distortion-quantization; f) Stefan rate-distortion

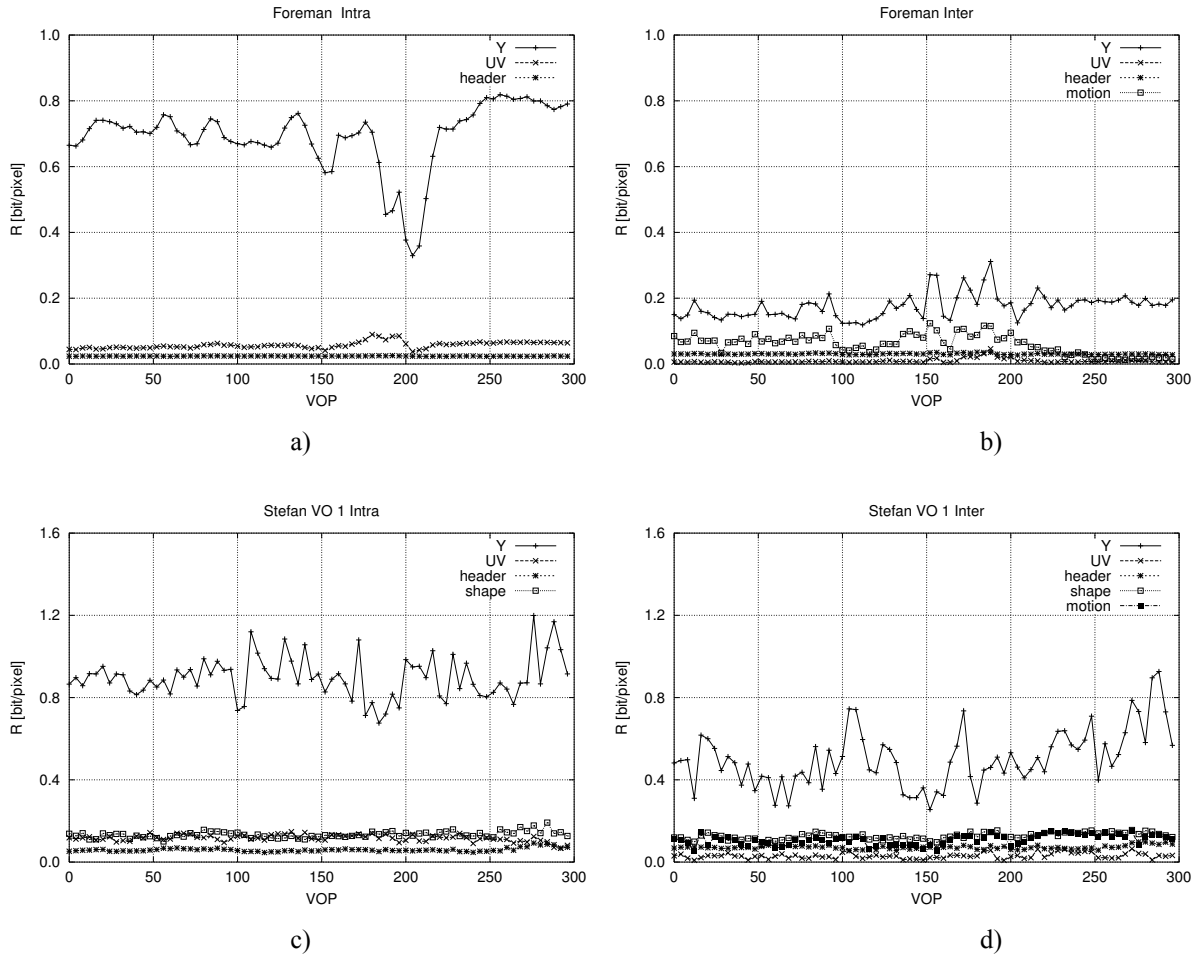


Figure 5.12 – Rate components for the Foreman and one arbitrarily shaped VO of the Stefan sequences encoded in Intra and Inter mode: a) Foreman Intra; b) Foreman Inter; c) Stefan VO 1 Intra; d) Stefan VO 1 Inter

Since the purpose of this work is to study and compare several rate and distortion models in order to find for each modeling scenario the model that better approximates the experimental data – **low model mismatch requirement** (see 5.2.1) – it is important to choose analytical models with few model parameters that resemble the typical behavior of the experimental curves as illustrated in Figure 5.11. Moreover, the model parameters should be easy to estimate during real-time encoding – **low complexity requirement** (see 5.2.1).

In order to evaluate how well each model represents the experimental rate and distortion characteristics, several test sequences (see Figure 5.13) in QCIF and CIF format at 30 HZ will be encoded with the MPEG-4 reference software video encoder [32] without rate control for different values of the quantization parameter.

The model parameters will be computed for each model and for each encoded picture using a nonlinear least squares estimation algorithm, more precisely the Levenberg-Marquardt algorithm [166].



Figure 5.13 – First frame of each test sequence: a) Foreman; b) Stefan; c) News; d) Kayak; e) Mother and Daughter (M&D); f) Football

In order to compare the different models it is also important to identify some meaningful comparison criteria, which are presented below.

Average Model Fitting Error Criterion

Since the main objective of this work is to obtain a model that closely approaches the experimental data over a wide range of source characteristics, it is important to have a measure that grabs this feature.

The measure adopted to evaluate how well a given model matches the experimental data for each picture is the standard deviation of the fit (*stdfit*), which is the root mean square of the error between the experimental and the estimated data, i.e., for a model with m parameters described by a function, $f(x)$, estimated from N data points, the *stdfit* is defined by

$$stdfit = \sqrt{\frac{\sum_{i=1}^N (y_i - f(x_i))^2}{N - m}} \quad (5.64)$$

where $N - m$ is the number of degrees of freedom of the model fitting operation (in the cases considered here $N = 31$, corresponding to all possible quantizer values).

Thus, the first model comparison criterion used in this model comparison is the minimization of the average *stdfit* over all pictures of each sequence.

Model Fitting Error Deviation Criterion

The *Average Model Fitting Error Criterion* however does not give enough information on how well a given model adapts itself to changes in the source data along the sequence, i.e., how stable is the estimated model along time. This characteristic can be grabbed by the standard deviation of the *stdfit*. Thus, this measure is used to give some insight on the

deviations of the model fitting results along time.

Model Estimation Complexity

While polynomial models can be estimated through least squares estimation, other models require nonlinear estimation techniques. Typically nonlinear estimation techniques are iterative, i.e., the algorithm starts with some initial model parameter estimates and iterates until a stop criterion is met. The stop criterion is usually defined as a threshold, ε , on the relative change of the fitting error between two iterations. Additionally, in order to prevent the iterative process to run infinitely, also the maximum number of iterations is limited. Therefore the model estimation complexity should also be taken into account. This will be evaluated using the average number of iterations per picture along the sequence required to estimate each model.

The Levenberg-Marquardt algorithm threshold, ε , was set to 10^{-3} , i.e., the algorithm stops whenever $|sse_{i+1} - sse_i| < 10^{-3}$, where sse is the sum of the square errors between the model and the actual data. Notice that a lower threshold will lead to a better model approximation at the expense of a higher number of iterations. Moreover, the model parameters must not be too different in magnitude. The larger the ratio of the largest and the smallest absolute parameter values, the slower the fit will converge. This problem can be circumvented when some a priori knowledge about the parameter values exists. In this case, the model parameters can be normalized, e.g., substitute each model parameter a_i by $\kappa_i \cdot a'_i$, where each κ_i is a constant such that the new model parameters a'_i are close in magnitude.

Model Parameter Variation Criterion

In order for a model to be useful, notably in real-time encoding situations, the range of variation of its parameters should be relatively bounded in order that the model parameters can be easily predictable, hence the model can be straightforwardly estimated. Consequently, the constancy of the model parameters is also important in this context of model comparison.

This criterion, however, is not easily usable since it is difficult to compare variation ranges of parameters with different meanings, notably when comparing different models. Nevertheless, for some models it can be observed that some model parameters exhibit much less variations along the sequence than others, i.e., are less content dependent. For these cases, the low variation parameters can be held constant. In this context, the models will be evaluated qualitatively, taking into account the number of varying parameters and the impact in terms of the Model Fitting Error Criterion on fixing the parameters with less variation.

5.3.1 Rate and Distortion Models for Intra Coding

In the case of Intra coding, the VOP to be encoded does not depend on other (past or future) VOPs; therefore, its rate and distortion characteristics depend exclusively on the current quantizer parameter(s) and the VOP statistics.

RATE-QUANTIZATION MODEL

The rate-quantization model is useful when the primary rate control objective is to maximize the picture quality given a certain target average bit-rate. In this Thesis, the following models are proposed for study and comparison:

Rate-Quantization Model I

$$R(Q) = \exp(-a \cdot Q^c + b) \quad (5.65)$$

Rate-Quantization Model II

$$R(Q) = a \cdot \frac{1}{Q^c} + b \quad (5.66)$$

Rate-Quantization Model III

$$R(Q) = \frac{a}{Q^c + b} \quad (5.67)$$

Rate-Quantization Model IV

$$R(Q) = a \cdot \frac{1}{Q^2} + b \cdot \frac{1}{Q} + c \quad (5.68)$$

where a , b , and c , are the model parameters for each model.

In this case, the test sequences have been encoded with the MPEG-4 reference software video encoder [32] using only the Intra coding mode with different values of the quantization parameter $Q \in \{1, \dots, 31\}$. Table 5.1 to Table 5.4 illustrate the rate-quantization model parameters results for the *Foreman* and *Stefan* sequences in QCIF and CIF formats, indicating for each model parameter its minimum, maximum, and mean value and standard deviation, measured over all encoded pictures for each sequence (the results for the other sequences indicated in Figure 5.13 are included in Annex A).

The less the model parameters depend on the picture content, the more robust and useful is the model for rate control purposes. For example, the first three models contain one term where the quantization parameter, Q , is raised to a parameter c . For Model I (5.65) c has an average value of approximately 0.2 for QCIF and 0.1 for CIF; for Model II (5.66) c has an average value of approximately 0.6 for QCIF and 0.8 for CIF; and for Model III (5.67) c has an average value of approximately 1.0 for QCIF and CIF. Since these parameters also exhibit small standard deviations, they can be considered less picture dependent than the other model parameters and can be kept constant if a simpler model is aimed.

Table 5.5 summarizes the *Average Model Fitting Error Criterion* results for the set of test sequences presented in Figure 5.13. As can be seen from this table, Model I exhibits the lowest average *stdfit* error for the QCIF set while Model II exhibits the lowest average *stdfit* error for the CIF set. Considering both formats, the average *stdfit* error results are very similar for these two models, while Model III and Model IV exhibit a higher average *stdfit* error in both cases.

Regarding the *Model Fitting Error Deviation Criterion* (see Table 5.6), all the models exhibit generally similar results, indicating analogous constancy in terms of the fitting error variation along the sequences.

Regarding the *Model Estimation Complexity Criterion* (see Table 5.7), Models II, III, and IV converge faster than Model I, requiring typically less than three iterations to converge, while Model I requires on average around six iterations to converge. It is important to notice that the number of iterations depends on the threshold selected, e.g., for a threshold equal to 10^{-1} all models converge typically in less than two iterations on average (see Table 5.8) with similar

average *stdfit* values, except for Model I where the average *stdfit* increases approximately 70% in this case.

Regarding the *Model Parameter Variation Criterion*, as it is illustrated in Table 5.1 to Table 5.4, all models exhibit, typically, two changing parameters (parameter *a*, and *b*) and one steady parameter (parameter *c*).

Table 5.1 – Rate-quantization model parameters for the Foreman sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.46	8.00	4.73	1.47
	<i>b</i>	4.18	9.31	6.28	1.38
	<i>c</i>	0.09	0.25	0.16	0.05
II	<i>a</i>	3.09	6.28	5.05	0.77
	<i>b</i>	-0.53	-0.09	-0.29	0.13
	<i>c</i>	0.64	0.81	0.69	0.03
III	<i>a</i>	4.41	12.45	8.07	2.41
	<i>b</i>	0.33	1.20	0.68	0.29
	<i>c</i>	0.90	1.14	1.00	0.08
IV	<i>a</i>	-3.68	-0.96	-2.54	0.64
	<i>b</i>	3.96	9.41	7.24	1.30
	<i>c</i>	-0.07	0.07	0.02	0.05

Table 5.2 – Rate-quantization model parameters for the Foreman sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.10	42.91	13.99	14.22
	<i>b</i>	3.84	43.96	15.42	14.07
	<i>c</i>	0.02	0.28	0.12	0.09
II	<i>a</i>	2.54	6.35	4.48	1.07
	<i>b</i>	-0.60	0.06	-0.19	0.21
	<i>c</i>	0.63	0.98	0.79	0.09
III	<i>a</i>	1.83	13.61	6.98	3.62
	<i>b</i>	-0.30	1.41	0.55	0.48
	<i>c</i>	0.76	1.20	1.04	0.10
IV	<i>a</i>	-3.81	-0.03	-1.68	1.15
	<i>b</i>	2.57	9.57	5.95	2.02
	<i>c</i>	-0.11	0.08	-0.02	0.05

Table 5.3 – Rate-quantization model parameters for the Stefan sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	1.69	2.37	2.02	0.13
	<i>b</i>	3.75	4.23	3.94	0.09
	<i>c</i>	0.22	0.28	0.25	0.01
II	<i>a</i>	7.16	10.03	7.90	0.52
	<i>b</i>	-1.59	-0.83	-1.07	0.14
	<i>c</i>	0.46	0.53	0.50	0.10
III	<i>a</i>	13.36	20.65	15.50	1.41
	<i>b</i>	1.08	1.52	1.28	0.09
	<i>c</i>	1.00	1.01	0.99	0.01
IV	<i>a</i>	-7.66	-5.09	-5.86	0.49
	<i>b</i>	11.22	15.87	12.50	0.85
	<i>c</i>	0.11	0.17	0.13	0.01

Table 5.4 – Rate-quantization model parameters for the Stefan sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.86	6.60	4.38	0.79
	<i>b</i>	4.78	8.13	6.06	0.72
	<i>c</i>	0.10	0.19	0.14	0.02
II	<i>a</i>	4.83	7.82	5.81	0.57
	<i>b</i>	-0.77	-0.22	-0.41	0.11
	<i>c</i>	0.54	0.65	0.60	0.02
III	<i>a</i>	6.25	13.10	8.50	1.32
	<i>b</i>	0.35	0.88	0.57	0.11
	<i>c</i>	0.84	0.92	0.88	0.02
IV	<i>a</i>	-5.33	-2.50	-3.47	0.55
	<i>b</i>	6.95	12.11	8.67	0.99
	<i>c</i>	0.13	0.20	0.16	0.01

Table 5.5 – Rate-quantization average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.007	0.018	0.019	0.032
	Stefan	0.030	0.019	0.063	0.119
	News	0.011	0.008	0.022	0.047
	Kayak	0.016	0.024	0.039	0.055
	M&D	0.008	0.018	0.009	0.011
	Football	0.013	0.041	0.025	0.038
AVG QCIF		0.014	0.021	0.030	0.050
CIF	Foreman	0.011	0.022	0.013	0.013
	Stefan	0.025	0.011	0.039	0.078
	News	0.020	0.013	0.012	0.028
	Kayak	0.018	0.012	0.035	0.052
	M&D	0.011	0.007	0.006	0.008
	Football	0.039	0.020	0.018	0.025
AVG CIF		0.021	0.014	0.021	0.034
AVG QCIF + CIF		0.017	0.018	0.025	0.042

Table 5.6 – Rate-quantization model fitting error deviation results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.002	0.013	0.007	0.007
	Stefan	0.004	0.003	0.007	0.012
	News	0.001	0.002	0.002	0.002
	Kayak	0.008	0.009	0.009	0.019
	M&D	0.001	0.002	0.001	0.001
	Football	0.005	0.013	0.009	0.018
AVG QCIF		0.004	0.007	0.006	0.010
CIF	Foreman	0.004	0.012	0.014	0.014
	Stefan	0.002	0.002	0.005	0.010
	News	0.002	0.001	0.001	0.001
	Kayak	0.004	0.005	0.005	0.011
	M&D	0.001	0.001	0.001	0.001
	Football	0.040	0.011	0.008	0.012
AVG CIF		0.009	0.005	0.006	0.008
AVG QCIF + CIF		0.006	0.006	0.006	0.009

Table 5.7 – Rate-quantization model estimation complexity results ($\varepsilon = 10^{-3}$)

	SEQ	MODEL			
		I	II	III	IV*
QCIF	Foreman	10.4	2.8	2.8	2.8
	Stefan	3.5	3.5	2.5	2.1
	News	5.6	2.9	2.5	2.0
	Kayak	4.2	3.0	2.7	2.9
	M&D	6.3	2.5	2.9	3.0
	Football	7.1	3.0	3.1	3.3
AVG QCIF		6.2	3.0	2.8	2.7
CIF	Foreman	15.4	2.7	3.1	3.4
	Stefan	4.0	3.0	2.5	2.2
	News	3.1	2.4	2.6	2.1
	Kayak	4.6	3.1	2.7	2.8
	M&D	3.0	2.7	2.9	2.7
	Football	7.8	3.0	3.1	3.3
AVG CIF		6.3	2.8	2.8	2.8
AVG QCIF + CIF		6.3	2.9	2.8	2.7

* **Note:** Model IV can also be estimated in one single iteration using linear least squares estimation.

Table 5.8 – Rate-quantization model estimation complexity results ($\varepsilon = 10^{-1}$)

	SEQ	MODEL			
		I	II	III	IV*
QCIF	Foreman	2.0	1.8	1.6	1.4
	Stefan	1.6	1.8	1.2	1.1
	News	1.3	1.6	1.2	1.0
	Kayak	2.1	1.9	1.6	1.5
	M&D	1.4	1.1	1.5	1.4
	Football	2.6	1.9	2.1	2.0
AVG QCIF		1.8	1.7	1.5	1.4
CIF	Foreman	1.4	1.7	2.1	2.3
	Stefan	1.4	1.9	1.3	1.1
	News	1.0	1.3	1.3	1.0
	Kayak	1.8	2.1	1.6	1.4
	M&D	1.0	1.4	1.4	1.3
	Football	2.0	2.1	2.1	2.0
AVG CIF		1.4	1.8	1.6	1.5
AVG QCIF + CIF		1.6	1.7	1.6	1.5

* **Note:** Model IV can also be estimated in one single iteration using linear least squares estimation.

Below a new set of (simpler) models with only two adjustable parameters is presented where some parameters of the models (5.65) to (5.68) have been kept constant following the results in the tables above. Notice that by reducing the number of model parameters to two, all models can now be estimated through linear least squares estimates, since they can be rewritten as a straight line equation of the form: $y = a \cdot x + b$, i.e.,

Rate-Quantization Model I with $c = c_0$

$$\log_e R(Q) = -a \cdot Q^{c_0} + b, \text{ with } y = \log_e R(Q) \text{ and } x = Q^{c_0}$$

Rate-Quantization Model II with $c = c_0$

$$R(Q) = a \cdot \frac{1}{Q^{c_0}} + b, \text{ with } y = R(Q) \text{ and } x = \frac{1}{Q^{c_0}}$$

Rate-Quantization Model III with $c = c_0$

$$\frac{1}{R(Q)} = \frac{1}{a} \cdot Q^{c_0} + \frac{b}{a}, \text{ with } y = \frac{1}{R(Q)} \text{ and } x = Q^{c_0}$$

Rate-Quantization Model II, and III with $b = 0$

$$\log_e R(Q) = \log_e a - c \cdot \log_e Q, \text{ with } y = \log_e R(Q) \text{ and } x = \log_e Q$$

Rate-Quantization Model IV, with $c = 0$

$$R(Q) \cdot Q = a \cdot \frac{1}{Q} + b, \text{ with } y = R(Q) \cdot Q \text{ and } x = \frac{1}{Q}$$

Notice that in this case the model parameters can be computed through the following expressions:

$$a = \frac{N \left(\sum_{i=1}^N x_i y_i \right) - \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N y_i \right)}{N \left(\sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2}, \quad b = \frac{\left(\sum_{i=1}^N x_i^2 \right) \left(\sum_{i=1}^N y_i \right) - \left(\sum_{i=1}^N x_i \right) \left(\sum_{i=1}^N x_i y_i \right)}{N \left(\sum_{i=1}^N x_i^2 \right) - \left(\sum_{i=1}^N x_i \right)^2}$$

Table 5.9 to Table 5.12 illustrate the new rate-quantization model parameter results for the *Foreman* and *Stefan* sequences in QCIF and CIF formats, where some parameters of the models presented above have been kept constant (the results for the other sequences mentioned in Figure 5.13 are included in Annex A). Notice that the rate-quantization model proposed in the informative Annex L of the MPEG-4 Visual standard [29] corresponds to Model IV (5.68) with $c = 0$. As can be seen from the tables, the new model parameters tend generally to exhibit lower standard deviations when compared with the same parameters in the corresponding three parameters models.

Table 5.13 summarizes the *Average Model Fitting Error Criterion* results for the two parameters models, for the set of test sequences presented in Figure 5.13. As can be seen from this table, all models exhibit now a higher average *stdfit* error, relatively to the three parameters models. The average *stdfit* increases between approximately 40% for Model IV (5.68) and 200% for Model II (5.66). Nevertheless, Model I (5.65) still outperforms the other models, notably model IV (5.68), i.e., the suggested MPEG-4 Visual rate-quantization model [29]. This means that Model I (5.65) would be, in general, the best performing rate-quantization model in terms of the *Average Model Fitting Error Criterion*.

Regarding the *Model Fitting Error Deviation Criterion* (see Table 5.14), Model I and Model III exhibit typically lower model error variations expressed by a lower standard deviation of the *stdfit* error, thus performing better than the other models with only two parameters in

terms of this criterion.

Since all models can now be estimated through linear least squares estimates, all models are now equally complex in terms of the *Model Estimation Complexity Criterion* since in this case there is no need to use iterative methods.

Regarding the *Model Parameter Variation Criterion*, all models exhibit now two changing parameters, as it is illustrated in Table 5.9 to Table 5.12.

Table 5.9 – Rate-quantization model parameters for the Foreman sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	3.00	3.57	3.14	0.11
	b	4.29	5.00	4.68	0.19
	$c = 0.2$	-	-	-	-
II	a	3.17	6.34	5.13	0.76
	b	-0.67	-0.32	-0.49	0.10
	$c = 0.6$	-	-	-	-
III	a	3.89	9.42	7.85	1.12
	b	0.28	0.81	0.66	0.09
	$c = 1.0$	-	-	-	-
II, III	a	3.04	5.95	4.88	0.69
	$b = 0$	-	-	-	-
	c	0.78	0.91	0.81	0.02
IV	a	-3.32	-0.82	-2.70	0.47
	b	3.84	8.99	7.41	1.07
	$c = 0$	-	-	-	-

Table 5.10 – Rate-quantization model parameters for the Foreman sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	7.45	8.83	7.82	0.25
	b	8.76	9.91	9.25	0.17
	$c = 0.1$	-	-	-	-
II	a	2.53	6.19	4.45	1.01
	b	-0.18	-0.05	-0.12	0.03
	$c = 0.8$	-	-	-	-
III	a	2.87	9.11	5.95	1.73
	b	0.05	0.58	0.37	0.12
	$c = 1.0$	-	-	-	-
II, III	a	2.56	5.97	4.36	0.94
	$b = 0$	-	-	-	-
	c	0.84	0.99	0.88	0.03
IV	a	-3.09	-0.15	-1.56	0.80
	b	2.87	8.79	5.82	1.65
	$c = 0$	-	-	-	-

Table 5.11 – Rate-quantization model parameters for the Stefan sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	2.65	2.76	2.71	0.02
	b	4.54	4.82	4.64	0.05
	$c = 0.2$	-	-	-	-
II	a	6.95	9.63	7.62	0.45
	b	-0.61	-0.45	-0.51	0.03
	$c = 0.6$	-	-	-	-
III	a	13.84	20.66	15.72	1.16
	b	1.20	1.44	1.32	0.05
	$c = 1.0$	-	-	-	-
II, III	a	6.57	9.08	7.19	0.42
	$b = 0$	-	-	-	-
	c	0.70	0.73	0.71	0.01
IV	a	-9.19	-5.93	-6.86	0.56
	b	12.13	17.63	13.59	0.94
	$c = 0$	-	-	-	-

Table 5.12 – Rate-quantization model parameters for the Stefan sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	6.19	6.54	6.37	0.07
	b	7.92	8.17	8.05	0.04
	$c = 0.1$	-	-	-	-
II	a	4.73	7.38	5.60	0.51
	b	0.06	0.18	0.11	0.02
	$c = 0.8$	-	-	-	-
III	a	9.05	15.96	11.40	1.33
	b	1.00	1.32	1.15	0.07
	$c = 1.0$	-	-	-	-
II, III	a	4.70	7.30	5.55	0.50
	$b = 0$	-	-	-	-
	c	0.71	0.75	0.73	0.01
IV	a	-6.82	-3.51	-4.65	0.64
	b	8.05	13.72	9.95	1.09
	$c = 0$	-	-	-	-

Table 5.13 – Rate-quantization average model fitting error results with a reduced number of model parameters

	SEQ	MODEL				
		I	II	III	II,III	IV
QCIF	Foreman	0.029	0.047	0.034	0.073	0.046
	Stefan	0.046	0.082	0.062	0.175	0.143
	News	0.034	0.025	0.040	0.057	0.076
	Kayak	0.025	0.030	0.051	0.122	0.061
	M&D	0.019	0.063	0.015	0.042	0.013
	Football	0.026	0.057	0.047	0.119	0.047
AVG QCIF		0.030	0.051	0.042	0.098	0.064
CIF	Foreman	0.034	0.043	0.028	0.057	0.026
	Stefan	0.035	0.114	0.064	0.080	0.124
	News	0.036	0.014	0.052	0.013	0.064
	Kayak	0.043	0.082	0.037	0.086	0.056
	M&D	0.022	0.033	0.012	0.007	0.013
	Football	0.031	0.049	0.028	0.062	0.032
AVG CIF		0.034	0.056	0.037	0.051	0.053
AVG QCIF + CIF		0.032	0.053	0.039	0.074	0.058

Table 5.14 – Rate-quantization model fitting standard deviation error results with a reduced number of model parameters

	SEQ	MODEL				
		I	II	III	II, III	IV
QCIF	Foreman	0.008	0.007	0.015	0.033	0.008
	Stefan	0.009	0.015	0.007	0.018	0.013
	News	0.004	0.003	0.003	0.005	0.003
	Kayak	0.008	0.01	0.009	0.014	0.017
	M&D	0.002	0.002	0.002	0.003	0.002
	Football	0.012	0.017	0.020	0.046	0.017
AVG QCIF		0.007	0.009	0.009	0.020	0.010
CIF	Foreman	0.033	0.036	0.031	0.050	0.026
	Stefan	0.009	0.024	0.004	0.017	0.013
	News	0.003	0.002	0.002	0.001	0.001
	Kayak	0.012	0.020	0.006	0.016	0.015
	M&D	0.002	0.002	0.001	0.001	0.001
	Football	0.022	0.031	0.016	0.038	0.013
AVG CIF		0.014	0.019	0.010	0.021	0.012
AVG QCIF + CIF		0.010	0.014	0.010	0.020	0.011

From the results presented above, it is possible to draw a first conclusion that the best average model fitting error approximately duplicates when the number of model parameters is decreased from three to two (see Table 5.5 and Table 5.13). Notice, however, that when the number of model parameters is reduced to two the model estimation complexity is also reduced (for the set of models in comparison) which creates a fitting error-complexity trade-off.

Regarding the identification of the best performing model, if the primary selection criterion is

the model fitting error, Model I and Model II become the best models since they achieve typically the lowest average fitting errors, with Model I achieving typically a lower average fitting error for QCIF resolution and Model II for CIF (see Table 5.5). Nevertheless it is hard to rank these two models based exclusively on the average fitting error. If the model complexity estimation is also considered, then Model II wins, since it converges additionally in a smaller number of iterations, thus being less complex to estimate (see Table 5.7).

However, if the model complexity estimation becomes the primary selection criterion the best model should be selected from the set of models with only two parameters since these models do not require iterative estimation methods. In this case, Model I (with $c = 0.2$ for QCIF and $c = 0.1$ for CIF) becomes the best choice, since this is the model that leads to the lowest average model fitting error from the set of models under consideration, closely followed by Model II (with $c = 0.6$ for QCIF and $c = 0.8$ for CIF) (see Table 5.13).

Finally, it is important to highlight that the results presented in this section indicate that Model IV, either with three or two parameters (the rate-quantization model suggested in the informative Annex L of the MPEG-4 Visual standard [29]), exhibits generally a higher modeling error than Model I with only two parameters.

DISTORTION-QUANTIZATION MODEL

The distortion-quantization model is useful when the primary rate control goal is to minimize the encoded bit-rate given a certain target picture quality. In this section, similarly to what has been done for the rate-quantization modeling investigation, four different distortion-quantization models are proposed for study and comparison based on the observation of experimental distortion-quantization curves:

Distortion-Quantization Model I

$$D(Q) = \exp(a \cdot Q^c + b) \quad (5.69)$$

Distortion-Quantization Model II

$$D(Q) = a \cdot (1 - \exp(-b \cdot Q^c)) \quad (5.70)$$

Distortion-Quantization Model III

$$D(Q) = a \cdot Q^c + b \quad (5.71)$$

Distortion-Quantization Model IV

$$D(Q) = a \cdot Q^2 + b \cdot Q + c \quad (5.72)$$

where a , b , and c , are the model parameters for each model.

Figure 5.14 presents the experimental distortion-quantization function for a frame of the *Foreman* and *Stefan* sequences encoded in Intra mode and the corresponding Model III (5.71) approximations. Notice that the theoretical distortion-quantization function for high-resolution quantization, (5.20), approximates the experimental data only for small values of the quantization parameter; for $Q > 5$ it significantly diverges from the actual characteristic, indicating that it can be used for modeling purposes only for a limited range of quantizer values. This function is a particular case of Model III (5.71) with $a = 2^2 / 12$, $b = 0$, and $c = 2$ (e.g., the Model III parameters are $a = 2.0$, $b = -3.7$, and $c = 1.2$ in Figure 5.14a, and

$a = 1.8$, $b = -5.8$, and $c = 1.5$ in Figure 5.14b).

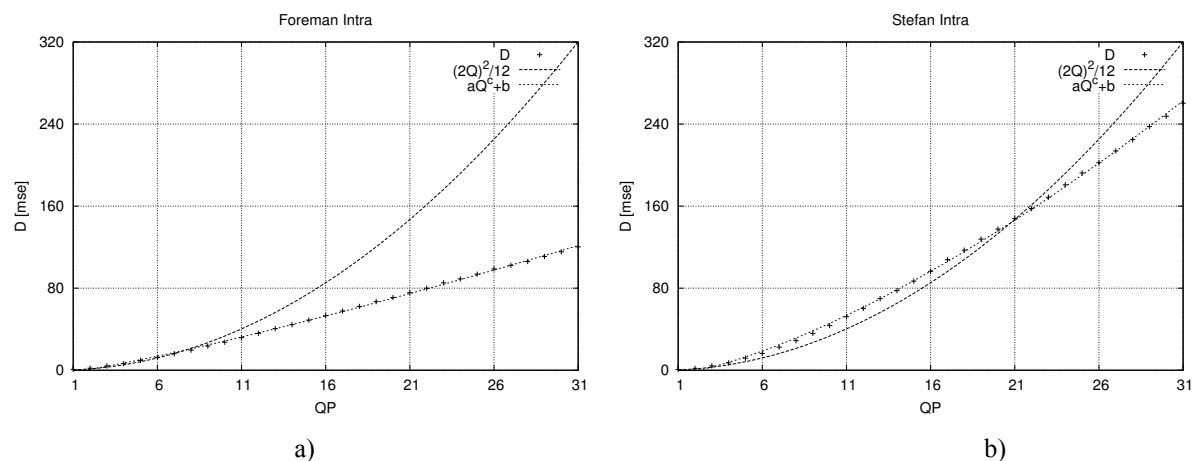


Figure 5.14 – Experimental distortion-quantization function and model approximation for Intra coding: a) Foreman sequence; b) Stefan sequence

Table 5.15 to Table 5.18 illustrate the distortion-quantization model parameters results for the *Foreman* and *Stefan* sequences in QCIF and CIF formats, indicating for each model parameter its minimum, maximum, mean and standard deviation, measured over all encoded pictures for each sequence (the results for the other sequences mentioned in Figure 5.13 are included in Annex A).

Similarly to what occurs for the rate-quantization models, parameter c of distortion-quantization Models I (5.69), II (5.70), and III (5.71), also exhibits small standard deviations and can be kept constant if a simpler model is aimed. For Model I, c has an average value of 0.1; for Model II, c has an average value of 1.5; and for model III, c has an average value of 1.2. In the case of Model IV (5.72), parameter a also exhibits a small standard deviation and has an average value of zero, indicating that the distortion tends to grow linearly with Q instead of growing quadratically.

Table 5.19 summarizes the *Average Model Fitting Error Criterion* results for the set of test sequences presented in Figure 5.13. As can be seen from this table, Model II exhibits the lowest average *stdfit* error in nearly all cases with an average *stdfit* error approximately 40% lower than Model III, the next best performing model in terms of this metric.

Regarding the *Model Fitting Error Deviation Criterion*, Model II is also the best performing model exhibiting generally a lower variation of the fitting error indicated by a lower standard variation of the *stdfit* error along the sequences (see Table 5.20).

Regarding the *Model Estimation Complexity Criterion*, Model I, III, and IV converge typically faster than Model II, requiring usually less than half the number of iterations of Model II to converge for a threshold $\varepsilon = 10^{-3}$ (see Table 5.21).

Regarding the *Model Parameter Variation Criterion*, as it is illustrated in Table 5.15 to Table 5.18 and was already mentioned above, all models exhibit, typically, two changing parameters and one steady parameter. Therefore, if a simpler model is aimed, these models can be simplified keeping the steady parameter constant and estimating the remaining parameters through linear least squares. However, reducing the number of model parameters to two has a higher impact in Models II, III, and IV, than in Model I, as can be seen comparing the results in Table 5.19 and Table 5.22. While the average model fitting error increases approximately 5% for Model I, when the number of parameters is reduced to two, for the other models the

average model fitting error increases between 100% and 180%, respectively for Model II and Model IV. For the results presented in Table 5.22, the following parameters have been set constant: $c = 0.1$ for Model I, $c = 1.5$ for Model II, $c = 1.2$ for Model III, and $a = 0$ for Model IV.

Table 5.15 – Distortion-quantization model parameters for the Foreman sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	0.76	12.07	11.09	0.93
	b	-10.75	1.25	-10.18	0.89
	c	0.08	0.45	0.09	0.02
II	a	89.77	497.45	280.50	72.76
	$b \times 10^{-3}$	1.88	8.41	3.44	1.35
	c	1.32	1.71	1.57	0.05
III	a	0.91	5.18	2.46	1.27
	b	-10.69	-1.17	-4.66	2.91
	c	0.92	1.38	1.21	0.12
IV	a	-0.01	0.07	0.03	0.02
	b	1.90	5.62	3.66	1.06
	c	-12.57	-3.11	-7.41	2.65

Table 5.16 – Distortion-quantization model parameters for the Foreman sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	3.15	22.84	15.44	5.33
	b	-21.52	-1.95	-14.57	5.18
	c	0.04	0.21	0.07	0.02
II	a	135.67	452.38	280.08	77.87
	$b \times 10^{-3}$	1.12	7.74	4.28	1.09
	c	1.00	1.68	1.39	0.16
III	a	0.49	5.72	2.77	1.57
	b	-12.18	0.69	-4.40	4.07
	c	0.84	1.36	1.10	0.08
IV	a	-0.02	0.03	0.01	0.01
	b	1.02	6.13	3.32	1.51
	c	-14.10	-0.49	-5.76	4.34

Table 5.17 – Distortion-quantization model parameters for the Stefan sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.99	11.89	11.06	0.96
	<i>b</i>	-10.84	1.27	-10.14	1.06
	<i>c</i>	0.10	0.43	0.11	0.02
II	<i>a</i>	432.77	776.85	614.21	88.89
	$b \times 10^{-3}$	0.77	1.45	1.08	0.15
	<i>c</i>	1.77	2.00	1.85	0.05
III	<i>a</i>	1.23	2.36	1.79	0.22
	<i>b</i>	-9.77	-3.19	-6.40	1.21
	<i>c</i>	1.41	1.58	1.48	0.03
IV	<i>a</i>	0.13	0.22	0.16	0.02
	<i>b</i>	3.42	5.99	4.56	0.46
	<i>c</i>	-17.80	-8.43	-12.78	1.71

Table 5.18 – Distortion-quantization model parameters for the Stefan sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.71	11.56	10.81	1.01
	<i>b</i>	-10.90	1.30	-10.30	1.12
	<i>c</i>	0.10	0.50	0.11	0.03
II	<i>a</i>	412.21	719.13	561.87	55.08
	$b \times 10^{-3}$	0.69	1.26	0.97	0.09
	<i>c</i>	1.67	1.85	1.77	0.04
III	<i>a</i>	0.75	1.42	1.01	0.12
	<i>b</i>	-4.23	-1.14	-2.47	0.56
	<i>c</i>	1.47	1.60	1.53	0.02
IV	<i>a</i>	0.09	0.18	0.12	0.02
	<i>b</i>	2.15	3.85	2.75	0.31
	<i>c</i>	-9.96	-4.52	-6.57	1.00

Table 5.19 – Distortion-quantization average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	2.593	0.626	1.315	1.675
	Stefan	4.874	1.487	2.476	3.738
	News	1.752	0.766	0.694	1.232
	Kayak	4.052	0.775	2.139	2.608
	M&D	1.489	0.752	0.642	0.699
	Football	2.903	0.948	1.393	1.707
AVG QCIF		2.944	0.892	1.443	1.943
CIF	Foreman	1.955	0.490	0.974	1.070
	Stefan	2.421	0.521	1.072	1.905
	News	0.834	0.261	0.322	0.666
	Kayak	3.195	0.572	1.598	2.091
	M&D	0.855	0.355	0.322	0.342
	Football	1.958	0.663	0.969	1.125
AVG CIF		1.870	0.477	0.876	1.200
AVG QCIF + CIF		2.407	0.685	1.160	1.572

Table 5.20 – Distortion-quantization model fitting standard deviation error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	1.040	0.230	0.626	0.532
	Stefan	0.810	0.683	0.540	0.629
	News	0.209	0.274	0.124	0.161
	Kayak	0.576	0.232	0.359	0.529
	M&D	0.280	0.141	0.105	0.089
	Football	1.160	0.329	0.552	0.778
AVG QCIF		0.679	0.315	0.384	0.453
CIF	Foreman	1.541	0.223	1.051	0.996
	Stefan	0.476	0.204	0.229	0.318
	News	0.098	0.059	0.059	0.074
	Kayak	0.547	0.144	0.338	0.381
	M&D	0.211	0.091	0.043	0.037
	Football	1.060	0.294	0.549	0.585
AVG CIF		0.656	0.169	0.378	0.399
AVG QCIF + CIF		0.667	0.242	0.381	0.426

Table 5.21 – Distortion-quantization model estimation complexity results

	SEQ	MODEL			
		I	II	III	IV*
QCIF	Foreman	4.7	18.2	4.6	4.3
	Stefan	4.9	5.2	4.8	4.2
	News	4.7	5.6	4.3	3.9
	Kayak	4.7	9.7	4.7	4.3
	M&D	4.1	5.6	4.2	4.0
	Football	5.7	15.0	5.5	4.6
AVG QCIF		4.8	9.9	4.7	4.2
CIF	Foreman	6.1	12.5	4.4	4.3
	Stefan	5.2	4.1	4.3	4.0
	News	5.1	5.5	4.2	3.8
	Kayak	5.3	9.8	4.5	4.1
	M&D	5.1	11.2	4.0	4.1
	Football	5.9	14.3	5.6	4.5
AVG CIF		5.5	9.6	4.5	4.1
AVG QCIF + CIF		5.1	9.7	4.6	4.2

* **Note:** Model IV can also be estimated in one single iteration using linear least squares estimation.

Table 5.22 – Distortion-quantization average model fitting error results with a reduced number of model parameters

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	2.618	0.793	1.924	3.456
	Stefan	4.712	5.016	7.355	12.957
	News	1.668	0.845	2.65	5.372
	Kayak	4.062	1.521	2.696	5.175
	M&D	1.414	0.979	1.065	0.968
	Football	3.143	1.157	1.975	3.518
AVG QCIF		2.936	1.719	2.944	5.241
CIF	Foreman	2.14	0.973	1.6	1.705
	Stefan	2.298	3.136	5.527	9.384
	News	0.83	0.243	1.382	3.032
	Kayak	3.182	1.099	1.901	3.939
	M&D	0.79	0.681	0.784	0.419
	Football	2.172	0.775	1.509	2.688
AVG CIF		1.902	1.151	2.117	3.528
AVG QCIF + CIF		2.419	1.435	2.531	4.384

RATE-DISTORTION MODEL

Although the rate-quantization and distortion-quantization models can be used both in single and multiple video objects encoding, the actual rate-distortion function, $R(D)$, may be useful in the context of multiple video objects encoding, where the video quality should be kept approximately constant among the different VOs in the scene, and consequently the available bit rate should be allocated to each VO according to its rate-distortion characteristics.

In this scenario, the rate-distortion characteristics of each of the N VOs in the scene, $R_i(D)$, for $i=1, \dots, N$, can be used in the bit allocation step to solve the following problem: given a global target scene bit allocation, R_T , find the individual VO bit allocations, R_i , such that $\sum_{i=1}^N R_i = R_T$ subject to the restriction that all VOs are encoded with the same distortion, \hat{D} , i.e., $D_i = \hat{D}$. Therefore, the main goal is to find a solution to

$$\sum_{i=1}^N R_i(\hat{D}) = R_T \quad (5.73)$$

In a scenario where the different VOs in the scene have different priorities (e.g., due to different perceptual relevance or different error protection needs) (5.73) can be generalized to

$$\sum_{i=1}^N R_i(\omega_i \cdot \hat{D}) = R_T \quad (5.74)$$

where ω_i is a weight reflecting the VO i priority, such that $\sum_{i=1}^N \omega_i = 1$.

Figure 5.15 illustrates the experimental rate-distortion function for a frame of the *Foreman* and *Stefan* sequences encoded in Intra mode.

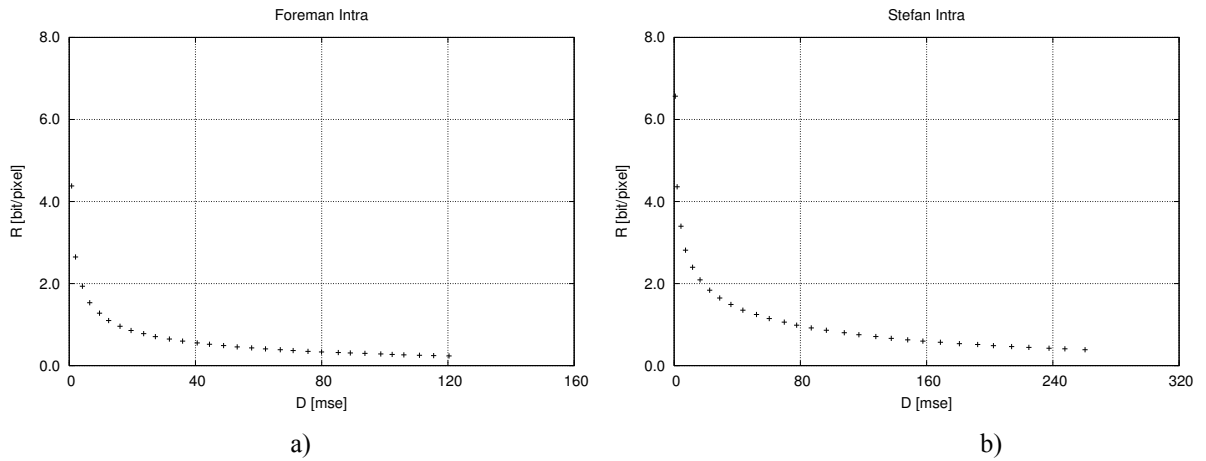


Figure 5.15 – Experimental rate-distortion function for Intra coding: a) *Foreman* sequence; b) *Stefan* sequence

Notice that the rate-distortion function, within certain restrictions, can be obtained either by combining the rate-quantization and distortion-quantization functions through the elimination of the Q variable, or estimated directly from the encoded data. In this section, this last approach is investigated. The following models are proposed for study and comparison:

Rate-Distortion Model I

$$R(D) = \exp\left(-a \cdot (\log_e D - d)^c + b\right) \quad (5.75)$$

Rate-Distortion Model II

$$R(D) = a \cdot \frac{1}{(D - d)^c} + b \quad (5.76)$$

Rate-Distortion Model III

$$R(D) = a \cdot \frac{1}{(D - d)^c + b} \quad (5.77)$$

Rate-Distortion Model IV

$$R(D) = a \cdot \frac{1}{D^2} + b \cdot \frac{1}{D} + c \quad (5.78)$$

Table 5.23 to Table 5.26 illustrate the distortion-quantization model parameters results for the *Foreman* and *Stefan* sequences in QCIF and CIF formats, indicating for each model parameter its minimum, maximum, and mean and standard deviation, measured over all encoded pictures for each sequence (the results for the other sequences mentioned in Figure 5.13 are included in Annex A).

Similarly to what occurs for the previous two modeling cases, parameter c of rate-distortion Models I (5.75), II (5.76), and III (5.77), also exhibits small standard deviations and can be kept constant if a simpler model is aimed. For Model I, c has an average value of 1.1; for Model II, c has an average value of 0.3; and for model III, c has an average value of 0.7.

Table 5.27 summarizes the *Average Model Fitting Error Criterion* results for the set of test sequences presented in Figure 5.13. As can be seen from this table, Model II exhibits the lowest average *stdfit* error in nearly all cases with an average *stdfit* error approximately 45% lower than Model I and Model III, the next best performing models in terms of this metric.

Regarding the *Model Fitting Error Deviation Criterion*, Model II is also the best performing model exhibiting generally a lower variation of the fitting error indicated by a lower standard deviation of the *stdfit* error along the sequences (see Table 5.28).

Regarding the *Model Estimation Complexity Criterion*, Model II converges typically faster than the other models, requiring usually less than half the number of iterations of Model I and approximately 35% less iterations than Model III to converge, all this for $\varepsilon = 10^{-3}$ (see Table 5.29).

Regarding the *Model Parameter Variation Criterion*, Models I, II, and III exhibit typically two changing parameters and two steady parameters, while Model IV exhibits two changing parameters and one steady parameter (see Table 5.23 to Table 5.26). Therefore, if a simpler model is aimed, these models can be simplified keeping the steady parameters constant and estimating the remaining parameters through linear least squares. However, reducing the number of model parameters to two, has a higher impact for Model II, than for the other Models as can be seen by comparing Table 5.27 and Table 5.30. While for Model II the average fitting error increases approximately 322% when the number of parameters is reduced to two, for the other models the average fitting error increases between 34% and 65%,

respectively for Model IV and Model I. For the results presented in Table 5.30, the following parameters have been set constant: $c = 1.1$ and $d = 1.4$ for Model I, $c = 0.3$ and $d = 0.5$ for Model II, $c = 0.7$ and $d = 0.8$ for Model III, and $c = 0.4$ for Model IV.

Table 5.23 – Rate-distortion model parameters for the Foreman sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	0.50	0.65	0.53	0.03
	b	0.07	0.94	0.67	0.19
	c	1.07	1.17	1.13	0.02
	d	1.19	1.97	1.63	0.16
II	a	2.33	6.10	4.46	0.98
	b	-1.49	-0.27	-0.72	0.39
	c	0.25	0.44	0.33	0.04
	d	0.16	0.39	0.25	0.04
III	a	4.33	15.51	10.32	3.10
	b	1.45	2.91	2.23	0.40
	c	0.66	0.83	0.72	0.03
	d	0.69	0.95	0.85	0.05
IV	a	-5.56	-1.17	-3.65	1.18
	b	3.65	11.09	8.04	1.93
	c	0.12	0.42	0.35	0.06

Table 5.24 – Rate-distortion model parameters for the Foreman sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	0.53	0.78	0.61	0.05
	b	-0.70	0.92	0.52	0.33
	c	1.03	1.18	1.13	0.04
	d	1.09	2.11	1.50	0.23
II	a	1.31	6.86	3.82	1.64
	b	-2.91	-0.10	-1.04	0.94
	c	0.16	0.45	0.30	0.08
	d	0.74	0.92	0.85	0.05
III	a	2.04	16.18	8.94	4.11
	b	0.85	3.07	2.15	0.55
	c	0.69	0.91	0.80	0.03
	d	0.76	1.03	0.92	0.05
IV	a	-5.73	-0.25	-3.10	1.53
	b	2.16	11.31	7.06	2.51
	c	0.05	0.39	0.22	0.09

Table 5.25 – Rate-distortion model parameters for the Stefan sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.38	0.42	0.41	0.01
	<i>b</i>	0.92	1.29	1.03	0.06
	<i>c</i>	1.14	1.16	1.15	0.00
	<i>d</i>	2.05	2.34	2.16	0.06
II	<i>a</i>	6.51	10.89	7.74	0.80
	<i>b</i>	-4.50	-1.72	-2.70	0.52
	<i>c</i>	0.13	0.21	0.17	0.02
	<i>d</i>	0.41	0.54	0.47	0.02
III	<i>a</i>	14.87	22.04	16.58	1.22
	<i>b</i>	2.16	2.61	2.46	0.08
	<i>c</i>	0.55	0.63	0.60	0.01
	<i>d</i>	0.72	0.81	0.77	0.01
IV	<i>a</i>	-8.07	-4.58	-5.44	0.52
	<i>b</i>	10.27	16.15	11.69	0.90
	<i>c</i>	0.71	1.11	0.83	0.07

Table 5.26 – Rate-distortion model parameters for the Stefan sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.41	0.48	0.44	0.01
	<i>b</i>	0.38	1.04	0.63	0.13
	<i>c</i>	1.07	1.14	1.10	0.01
	<i>d</i>	2.05	2.26	2.12	0.03
II	<i>a</i>	3.72	7.53	4.93	0.72
	<i>b</i>	-2.66	-0.90	-1.47	0.34
	<i>c</i>	0.16	0.23	0.19	0.02
	<i>d</i>	0.79	0.86	0.82	0.01
III	<i>a</i>	7.29	16.05	9.90	1.55
	<i>b</i>	1.59	2.29	1.84	0.13
	<i>c</i>	0.56	0.62	0.60	0.01
	<i>d</i>	0.74	0.80	0.77	0.01
IV	<i>a</i>	-5.86	-2.40	-3.40	0.63
	<i>b</i>	6.28	12.30	8.09	1.11
	<i>c</i>	0.45	0.85	0.59	0.08

Table 5.27 – Rate-distortion average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.034	0.019	0.037	0.189
	Stefan	0.085	0.019	0.078	0.429
	News	0.037	0.016	0.024	0.189
	Kayak	0.062	0.035	0.071	0.273
	M&D	0.022	0.019	0.028	0.098
	Football	0.055	0.036	0.068	0.260
AVG QCIF		0.049	0.024	0.051	0.240
CIF	Foreman	0.026	0.032	0.035	0.136
	Stefan	0.048	0.022	0.029	0.250
	News	0.021	0.011	0.006	0.082
	Kayak	0.049	0.015	0.052	0.218
	M&D	0.020	0.015	0.011	0.041
	Football	0.038	0.019	0.040	0.152
AVG CIF		0.034	0.019	0.029	0.147
AVG QCIF + CIF		0.041	0.022	0.040	0.193

Table 5.28 – Rate-distortion model fitting standard deviation error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.017	0.006	0.024	0.052
	Stefan	0.011	0.003	0.012	0.041
	News	0.002	0.002	0.002	0.007
	Kayak	0.008	0.006	0.009	0.040
	M&D	0.002	0.002	0.003	0.005
	Football	0.021	0.010	0.024	0.108
AVG QCIF		0.010	0.005	0.012	0.042
CIF	Foreman	0.027	0.007	0.031	0.083
	Stefan	0.004	0.003	0.007	0.041
	News	0.002	0.001	0.001	0.004
	Kayak	0.009	0.003	0.011	0.037
	M&D	0.001	0.001	0.001	0.002
	Football	0.018	0.007	0.024	0.082
AVG CIF		0.010	0.004	0.013	0.042
AVG QCIF + CIF		0.010	0.004	0.012	0.042

Table 5.29 – Rate-distortion model estimation complexity results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	3.5	2.5	2.3	2.8
	Stefan	4.0	5.3	3.9	4.9
	News	7.8	3.2	4.9	5.8
	Kayak	2.1	1.7	1.4	2.3
	M&D	3.5	2.5	2.3	2.8
	Football	4.0	5.3	3.9	4.9
AVG QCIF		7.8	3.2	4.9	5.8
CIF	Foreman	2.1	1.7	1.4	2.3
	Stefan	3.5	2.5	2.3	2.8
	News	4.0	5.3	3.9	4.9
	Kayak	7.8	3.2	4.9	5.8
	M&D	2.1	1.7	1.4	2.3
	Football	3.5	2.5	2.3	2.8
AVG CIF		4.0	5.3	3.9	4.9
AVG QCIF + CIF		7.8	3.2	4.9	5.8

Table 5.30 – Rate-distortion average model fitting error results with a reduced number of model parameters

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.049	0.086	0.042	0.198
	Stefan	0.106	0.234	0.146	0.572
	News	0.041	0.067	0.051	0.188
	Kayak	0.094	0.142	0.078	0.280
	M&D	0.029	0.046	0.052	0.218
	Football	0.101	0.148	0.084	0.302
AVG QCIF		0.070	0.121	0.076	0.293
CIF	Foreman	0.050	0.072	0.068	0.220
	Stefan	0.060	0.090	0.104	0.297
	News	0.059	0.032	0.027	0.147
	Kayak	0.061	0.090	0.055	0.221
	M&D	0.026	0.035	0.030	0.241
	Football	0.051	0.071	0.055	0.222
AVG CIF		0.051	0.065	0.057	0.225
AVG QCIF + CIF		0.061	0.093	0.066	0.259

5.3.2 Stationary Rate and Distortion Models for Inter Coding

In the case of Inter coding, the coding of a certain VOP does not depend only on its own content but also on its reference VOP(s), this means the coded versions of the VOPs used for prediction; therefore, its rate-distortion characteristics depend not only on the quantization parameter(s) and the picture statistics of the current VOP but also on the quantization parameter(s) and picture statistics of the reference VOPs (B-VOPs) or VOP (P-VOPs). For P-VOPs, the rate-quantization, distortion-quantization, and rate-distortion functions become, respectively, $R(Q, Q_{ref})$, $D(Q, Q_{ref})$, and $R(D, D_{ref})$. Since these bidimensional rate and distortion functions are difficult to obtain, at least for a wide range of Q and Q_{ref} values, it was decided in a first approach to estimate the different rate and distortion model parameters only for a reduced set of Q_{ref} to reduce the estimation complexity. In this case $Q_{ref} \in \{4, 8, 12, 16, 20, 24, 28\}$; this means that each model parameter a , b , and c , for the current picture becomes $a_{Q_{ref}}$, $b_{Q_{ref}}$, and $c_{Q_{ref}}$ since different values of a , b , and c will be obtained for each value of Q_{ref} . For the case where the rate distortion models can be considered approximately stationary regarding Q_{ref} then $a_{Q_{ref}}$, $b_{Q_{ref}}$, and $c_{Q_{ref}}$, become less dependent on the reference picture quantization parameter and the subscript can be dropped. This approximation will be further formalized and discussed in Section 0.

In this context, the main purpose of this section is to model the stationary rate and distortion component of the actual rate and distortion functions, i.e., for each Q_{ref} find the best analytical model that best fits the experimental data. Since the purpose of this work is to find for each type of rate and distortion models, the model that better approximates the experimental data, it is important to choose analytical models that resemble the typical behavior of the experimental curves as illustrated in Figure 5.11. This approach has already been followed in Section 5.3.1 focused on the Intra coding case. In this case, all models tested have three parameters a , b , and c .

Regarding the model comparison, solely the *Average Model Fitting Error Criterion* is used to simplify the model comparison, although the other criteria used in Section 5.3.1 could additionally be used. Notice, however, that in this case for each picture there are M different models estimated, i.e., one for each different Q_{ref} value, with $M = 7$. For this purpose, the set of sequences mentioned in Figure 5.13, in QCIF and CIF formats, at 15 Hz, have been encoded with the MPEG-4 reference software video encoder [32] without rate control, using the Inter coding mode for different values of the quantization parameter ($Q \in \{1, \dots, 31\}$) and different quantization parameters for the corresponding reference pictures with $Q_{ref} \in \{4, 8, 12, 16, 20, 24, 28\}$.

Similarly to the Intra coding approach, the model parameters have been estimated for each proposed model, for each encoded picture of each sequence, and for each Q_{ref} using the Levenberg-Marquardt algorithm [166].

STATIONARY RATE-QUANTIZATION MODEL

In this case, for the stationary component of the Inter rate-quantization model the same functions used for the Intra case are proposed for study and comparison:

Stationary Rate-Quantization Model I

$$R(Q) = \exp(-a \cdot Q^c + b) \quad (5.79)$$

Stationary Rate-Quantization Model II

$$R(Q) = a \cdot \frac{1}{Q^c} + b \quad (5.80)$$

Stationary Rate-Quantization Model III

$$R(Q) = \frac{a}{Q^c + b} \quad (5.81)$$

Stationary Rate-Quantization Model IV

$$R(Q) = a \cdot \frac{1}{Q^2} + b \cdot \frac{1}{Q} + c \quad (5.82)$$

Regarding the *Average Model Fitting Error Criterion* (see Table 5.31), the stationary rate-quantization model I (5.79) outperforms the other models exhibiting typically the lowest average fitting error for both QCIF and CIF formats. It is important to notice however that relatively to the Intra case, this model exhibits an average fitting error 100% higher than the best Intra rate-quantization model with three parameters (see Table 5.5) and similar to the best Intra rate-quantization model with only two parameters (see Table 5.13), which means that in the Inter case the rate-quantization characteristics are more difficult to model.

Table 5.31 – Stationary rate-quantization average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.035	0.054	0.057	0.058
	Stefan	0.053	0.082	0.103	0.071
	News	0.040	0.053	0.062	0.054
	Kayak	0.040	0.052	0.072	0.046
	M&D	0.027	0.043	0.043	0.048
	Football	0.032	0.054	0.062	0.044
AVG QCIF		0.038	0.056	0.067	0.054
CIF	Foreman	0.029	0.045	0.046	0.051
	Stefan	0.042	0.046	0.073	0.052
	News	0.033	0.030	0.041	0.033
	Kayak	0.041	0.033	0.068	0.034
	M&D	0.020	0.023	0.028	0.028
	Football	0.027	0.034	0.044	0.032
AVG CIF		0.032	0.035	0.050	0.038
AVG QCIF + CIF		0.035	0.046	0.058	0.046

STATIONARY DISTORTION-QUANTIZATION MODEL

For the stationary component of the Inter distortion-quantization model the same functions used for the Intra case are proposed for study and comparison:

Stationary Distortion-Quantization Model I

$$D(Q) = \exp(a \cdot Q^c + b) \quad (5.83)$$

Stationary Distortion-Quantization Model II

$$D(Q) = a \cdot (1 - \exp(-b \cdot Q^c)) \quad (5.84)$$

Stationary Distortion-Quantization Model III

$$D(Q) = a \cdot Q^c + b \quad (5.85)$$

Stationary Distortion-Quantization Model IV

$$D(Q) = a \cdot Q^2 + b \cdot Q + c \quad (5.86)$$

Regarding the *Average Model Fitting Error Criterion* (see Table 5.32), the stationary distortion-quantization model II (5.84) exhibits consistently the lowest average fitting error for both QCIF and CIF formats. Relatively to the Intra case, the best performing distortion-quantization model is the same but the average fitting error approximately duplicates (see Table 5.19).

Table 5.32 – Stationary distortion-quantization average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	6.449	0.973	4.604	2.532
	Stefan	11.565	2.055	8.321	6.321
	News	6.657	1.386	4.965	3.293
	Kayak	7.115	1.442	4.576	3.072
	M&D	4.419	0.499	3.222	1.769
	Football	5.590	1.613	3.394	2.640
AVG QCIF		6.966	1.328	4.847	3.271
CIF	Foreman	5.225	0.714	3.592	1.947
	Stefan	6.685	2.002	4.792	3.756
	News	3.932	0.775	2.849	1.840
	Kayak	5.748	1.155	3.770	3.342
	M&D	2.639	0.276	1.869	1.036
	Football	3.702	1.138	2.203	1.625
AVG CIF		4.655	1.010	3.179	2.258
AVG QCIF + CIF		5.811	1.169	4.013	2.764

Notice that for Inter coding the distortion-quantization function besides being highly dependent on the average quantizer of the reference picture for $Q \gg Q_{ref}$ (see Figure 5.16), it exhibits clearly three different zones of variation, indicating that it can be piecewise approximated by simpler functions than (5.84) (see Figure 5.17). Therefore, this Thesis proposes to approximate (5.84) for different quantization parameter ranges. For small quantization steps (i.e., high-resolution quantization), since $e^{-x} \approx 1 - x$, the distortion-quantization function should be approximated by

$$D_1(Q) = abQ^c \quad (5.87)$$

Notice, that with $ab=1/12$ and $c=2$, (5.87) corresponds to the theoretical distortion-quantization approximation expressed by (5.20).

For medium quantization steps, the distortion increases approximately linearly with the quantization step. Therefore, the distortion-quantization function should be approximated by

$$D_2(Q) = a'Q + b' \quad (5.88)$$

where (5.88) can be seen as a linearization of (5.84) in the neighborhood of a given quantization parameter value Q_0 , i.e.,

$$D_2(Q) = D'(Q_0)(Q - Q_0) + D(Q_0) \quad (5.89)$$

where $D'(Q_0)$ is the derivative of $D(Q)$ for $Q = Q_0$.

Finally, for high quantization steps the distortion-quantization function is approximately constant, and the distortion-quantization function could be approximated simply by

$$D_3(Q) = C^{\text{te}} \quad (5.90)$$

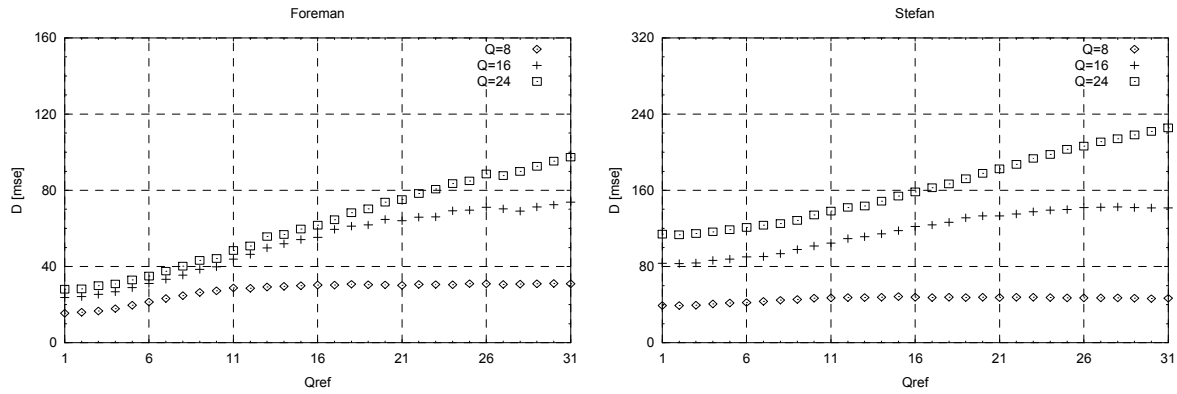


Figure 5.16 – Dependency of the distortion-quantization on the reference quantization parameter for Inter coding

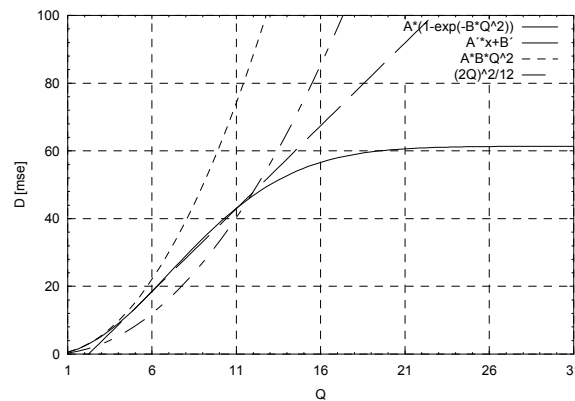


Figure 5.17 – Piecewise approximation of the distortion-quantization function

STATIONARY RATE-DISTORTION MODEL

While for the Inter rate and distortion models the functions used for the Intra coding case were

reused, for the rate-distortion models, it was found after an extensive analysis that the four parameter models (5.75) – (5.78) caused instability in the nonlinear estimation method; therefore slightly different versions of the Intra rate-distortion models were considered. Thus, for the Inter coding case, the following models are proposed for study and comparison:

Stationary Rate-Distortion Model I

$$R(D) = \exp(-a \cdot D^c + b) \quad (5.91)$$

Stationary Rate-Distortion Model II

$$R(D) = a \cdot \frac{1}{D^c} + b \quad (5.92)$$

Stationary Rate-Distortion Model III

$$R(D) = a \cdot \frac{1}{D^c + b} \quad (5.93)$$

Stationary Rate-Distortion Model IV

$$R(D) = a \cdot \frac{1}{D^2} + b \cdot \frac{1}{D} + c \quad (5.94)$$

Regarding the *Average Model Fitting Error Criterion* (see Table 5.33), the stationary rate-distortion model II (5.92) generally outperforms the other models exhibiting the lowest average fitting error.

Table 5.33 – Stationary rate-distortion average model fitting error results

	SEQ	MODEL			
		I	II	III	IV
QCIF	Foreman	0.057	0.041	0.092	0.086
	Stefan	0.113	0.070	0.166	0.296
	News	0.065	0.037	0.094	0.104
	Kayak	0.068	0.060	0.113	0.171
	M&D	0.044	0.021	0.067	0.046
	Football	0.058	0.070	0.103	0.175
AVG QCIF		0.068	0.050	0.106	0.146
CIF	Foreman	0.046	0.038	0.076	0.061
	Stefan	0.073	0.051	0.105	0.190
	News	0.038	0.023	0.055	0.047
	Kayak	0.062	0.047	0.096	0.174
	M&D	0.027	0.012	0.040	0.017
	Football	0.041	0.044	0.068	0.104
AVG CIF		0.048	0.036	0.073	0.099
AVG QCIF + CIF		0.058	0.043	0.090	0.123

5.3.3 Delta Rate and Distortion Models for Inter Coding

As mentioned in the previous section, rate and distortion modeling for Inter coding becomes

more complex since the rate and distortion model functions become bidimensional for P-VOPs, and thus harder to model with a simple model using a reduced number of parameters. In order to circumvent this problem, some assumptions need to be made.

When encoding a given VOP, typically any video encoder can choose one among, at least, two major encoding modes: Intra and Inter. It is up to the encoder control mechanism to select between these two coding modes aiming at achieving the best rate-distortion trade-off. While for off-line encoding typically a brute force approach can be followed, i.e., all possible combinations of coding modes and coding parameters are tested and the best one is selected, for real-time encoding the encoder control has to provide a good choice of the best encoding mode for each coding unit with a limited complexity (in this case a VOP, although it could also be a macroblock or a video packet⁸). For P-VOPs, this mode decision is typically based on the statistics of the prediction error between the VOP to be encoded and its reference.

Therefore it is reasonable to accept that under typical operation conditions the encoder control mechanism selects the Inter coding mode whenever the VOP to be encoded has a “good prediction” this means when the previous VOP is somehow similar to the current VOP (without loss of generality, it is assumed that a mechanism for deciding the encoding mode for each VOP based on some prediction error criteria is available). Consequently, whenever a given VOP is encoded in Inter mode it is assumed that its rate and distortion characteristics are similar to the previous VOP, i.e., its rate-distortion characteristics are stationary (in the case of scene changes, the encoder typically tends to switch to Intra coding mode). Moreover, since typically the rate control mechanism aims at achieving constant quality along time, under the assumption of stationary rate-distortion characteristics, the average quantization of consecutive VOPs should not change abruptly, since otherwise subjectively annoying quality fluctuations tend to occur.

In this case, the rate-quantization and distortion-quantization functions for the current VOP, if Inter coded, can be modeled with a stationary component $\tilde{R}(Q)$ and $\tilde{D}(Q)$ functions, plus an adaptation or dynamic component that depends on the difference between Q_T and Q_{ref} , where Q_T is the target quantization parameter to encode the current VOP, obtained from the $\tilde{R}(Q)$ or $\tilde{D}(Q)$ functions, e.g., $Q_T = \{Q : \tilde{R}(Q_T) = R_T\}$, i.e., Q_T is the target quantization parameter for a given target number of bits, R_T , or a given target distortion, D_T . $\tilde{R}(Q)$ and $\tilde{D}(Q)$ are approximated functions (just the stationary component) to the real rate-quantization or distortion-quantization functions. Thus, for small Q variations between successive VOPs, i.e., for $|\Delta Q| \leq \varepsilon$, where $\Delta Q = Q_T - Q_{ref}$, the rate and distortion functions can be approximated by the following equations

$$R(Q, Q_{ref}) = \tilde{R}(Q) + \Delta R(Q, Q_{ref}) \quad (5.95)$$

$$D(Q, Q_{ref}) = \tilde{D}(Q) + \Delta D(Q, Q_{ref}) \quad (5.96)$$

where $\tilde{R}(Q)$ and $\tilde{D}(Q)$ represent, respectively, the reference quantizer independent stationary components of $R(Q, Q_{ref})$ and $D(Q, Q_{ref})$, and $\Delta R(Q, Q_{ref})$ and $\Delta D(Q, Q_{ref})$ are delta functions that represent the difference between the actual and the approximated stationary functions this means the functions using only the stationary component. The rationale for such

⁸ In MPEG-4 video [10], a video packet corresponds to an integer number of MBs of a VOP.

decomposition comes from the fact that under typical Inter coding operation, the quantization parameter between successive time instants does not change abruptly, therefore $Q_{ref} \approx Q_0 = C^{te}$ along time. In this case, $\tilde{R}(Q)$ and $\tilde{D}(Q)$ can be seen as

$$\tilde{R}(Q) = R(Q, Q_{ref} = Q_0) \quad (5.97)$$

$$\tilde{D}(Q) = D(Q, Q_{ref} = Q_0) \quad (5.98)$$

Equations (5.97) and (5.98) represent the modeling approach where the dependency of the Inter rate and distortion functions on Q_{ref} is not considered [101, 134]. However, this fully stationary approach does not take into account the approximation error represented in equations (5.95) and (5.96) by $\Delta R(Q, Q_{ref})$ and $\Delta D(Q, Q_{ref})$. This situation is illustrated in Figure 5.18 for the rate-quantization case. The left side of Figure 5.18 represents the rate-quantization function for the reference VOP at time instant $t-1$ (solid line) while the right side shows three rate-quantization curves for the current VOP at time instant t , $R_t(Q, Q_{ref} = Q_0)$, $R_t(Q, Q_{ref} = Q_L)$, and $R_t(Q, Q_{ref} = Q_H)$ (notice that since the reference VOP has already been coded $R_{t-1}(Q, Q_{ref})$ corresponds to a single line depending only on Q since a certain Q_{ref} has been used). In this figure, $R(Q, Q_{ref} = Q_0)$ represents the stationary rate-quantization approximation for the current VOP that is approximated by the rate-quantization function of the reference VOP, based on the assumption that the quantization parameter along time is approximately constant, i.e., $Q = Q_0 \approx C^{te}$ (in this case $R(Q, Q_{ref}) \approx \tilde{R}(Q)$). However, if this assumption is not fully valid this means the reference VOP quantization parameter is not Q_0 , this will have an impact on the actual rate-quantization function of the current VOP. If the reference VOP is more finely encoded with a quantization parameter $Q = Q_L \leq Q_0$, the actual rate-distortion function of the current VOP, $R(Q, Q_{ref} = Q_L)$, will move down away from $\tilde{R}(Q)$; on the contrary, if the reference VOP is more coarsely encoded with $Q = Q_H \geq Q_0$, the actual rate-distortion function of the current VOP, $R(Q, Q_{ref} = Q_H)$, will move up away from $\tilde{R}(Q)$. This deviation is represented by the delta function $\Delta R(Q, Q_{ref})$.

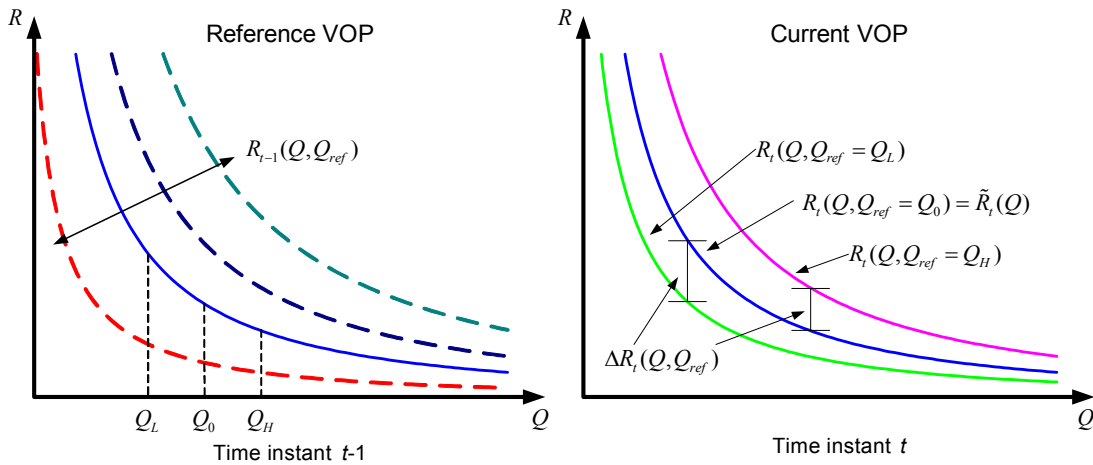


Figure 5.18 – Modification of the rate-quantization function when $Q_{ref} \neq Q_0$

Typically, these delta functions reveal a higher dependency on Q than on Q_{ref} , i.e., the error approximation of $R(Q, Q_{ref})$ and $D(Q, Q_{ref})$ by (5.97) and (5.98) is highly dependent on the quantization parameter Q and less dependent on Q_{ref} . This situation is illustrated in Figure 5.19 for the rate-quantization and distortion-quantization functions. Regarding the rate-quantization case, Figure 5.19a represents the actual $R(Q, Q_{ref})$ function for a frame of the *Foreman* sequence considering only some Q_{ref} values, while Figure 5.19b and Figure 5.19c represent, respectively, the approximation error resulting from approximating $R(Q, Q_{ref})$ by $R(Q, Q_{ref} + \Delta Q_{ref})$ for $\Delta Q_{ref} = 4$ and $\Delta Q_{ref} = 8$, respectively. As can be seen from these figures, the approximation error increases as the value of Q decreases, but along the Q axis is very similar for values of Q_{ref} for which $|Q - Q_{ref}| \leq \Delta Q_{ref}$.

Based on this experimental knowledge, the dependency of $\Delta R(Q, Q_{ref})$ and $\Delta D(Q, Q_{ref})$ on Q_{ref} can be dropped and they can be written solely as $\Delta R(Q)$ and $\Delta D(Q)$, i.e.,

$$\Delta R(Q) \approx R(Q, Q_{ref} + \Delta Q) - R(Q, Q_{ref}) \quad (5.99)$$

$$\Delta D(Q) \approx D(Q, Q_{ref} + \Delta Q) - D(Q, Q_{ref}) \quad (5.100)$$

Notice that the dependency of $\Delta R(Q)$ and $\Delta D(Q)$ on Q_{ref} has been alleviated under the assumption that $|Q - Q_{ref}| < \varepsilon$ (typically $\varepsilon \leq 8$); this would be less valid for Q and Q_{ref} rather different. As can be seen in Figure 5.19, $\Delta R(Q) \approx 0$ as Q increases. Similarly, $\Delta D(Q) \approx 0$ typically for $Q < Q_{ref}$ and tends to become constant as Q increases.

It is important to refer that when the target quantization parameter significantly differs from Q_{ref} , $\Delta R(Q)$ exhibits large amplitudes as can be seen in Figure 5.19 for $\Delta Q = 8$ and $Q_{ref} = 24$. In this case, for a target quantization parameter, Q_T , less than 8, i.e., significantly different from $Q_{ref} = 24$, the approximation above is no longer valid. Notice, however, as mentioned above, that the purpose of this refined modeling approach is still based on the assumption of approximately stationary conditions.

In this context, the main purpose of this section is to model the delta rate-quantization and delta distortion-quantization component of the actual rate and distortion functions, i.e., find an analytical model that adequately fits the experimental data representing the deviation between the stationary rate and distortion models and the actual rate and distortion experimental data. In terms of the modeling approach, it is important to choose analytical functions that resemble the experimental behavior of the actual rate and distortion characteristics as illustrated in Figure 5.19.

Regarding the rate-distortion function, based on a careful analysis of the experimental curves, it is possible to conclude that, generally, for small Q variations and $R \leq 0.5$ bit/pixel, the $R(D)$ characteristics can be considered approximately stationary, i.e., $R(D, Q_{ref}) \approx R(D)$. This assumption may reveal, however, in some cases to be a rough approximation, notably for higher bit rates.

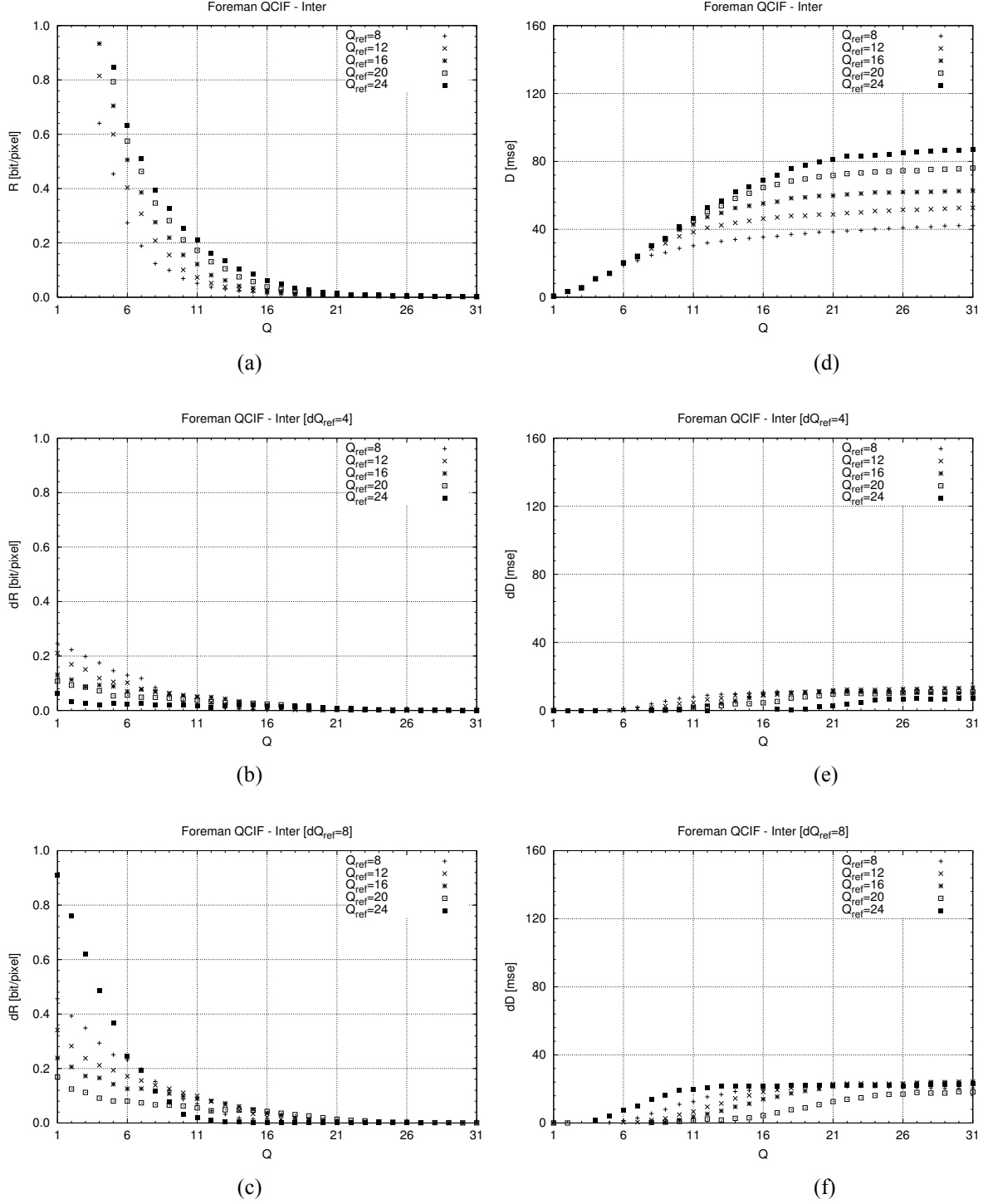


Figure 5.19 – Delta rate and distortion functions for the Foreman sequence: (a) $R(Q, Q_{ref})$; (b) $\Delta R(Q)$ for $\Delta Q_{ref} = 4$; (c) $\Delta R(Q)$ for $\Delta Q_{ref} = 8$; (d) $D(Q, Q_{ref})$; (e) $\Delta D(Q)$ for $\Delta Q_{ref} = 4$; (f) $\Delta D(Q)$ for $\Delta Q_{ref} = 8$

In order to compare the different delta rate and distortion models, it is important to choose meaningful comparison criteria. Similarly to the stationary models, since the main objective of this model analysis is to propose a model that closely approximates the experimental data over a wide range of source characteristics, the model comparison criterion proposed is the minimization of the average model fitting error.

In this context, the *stdfit* measure (5.64) is again used to evaluate how well each model matches the experimental data. In this case, however, for the delta rate-quantization models x_i and y_i in (5.64) are respectively, the quantization parameter Q_i and the rate difference ΔR_i , where ΔR_i is derived from the actual rate-quantization differences $R(Q, Q_{ref} + \Delta Q) - R(Q, Q_{ref})$; similarly, for the distortion-quantization models, $x_i = Q_i$ and $y_i = \Delta D_i$, where the distortion difference ΔD_i is derived from the actual distortion quantization differences $D(Q, Q_{ref} + \Delta Q) - D(Q, Q_{ref})$.

Thus, the model comparison criterion used is the minimization of the average *stdfit* over all pictures of each sequence and all different Q_{ref} for two different cases: $\Delta Q = 4$ and $\Delta Q = 8$.

To evaluate how well each model approximates the experimental data, the same test conditions used for the stationary models are used; again the model parameters have been estimated for each proposed model and for each encoded picture of each sequence using the Levenberg-Marquardt algorithm [166].

DELTA RATE-QUANTIZATION MODEL

Looking to the actual delta rate characteristics, the following $\Delta R(Q)$ models are proposed for study and comparison:

Delta Rate-Quantization Model I

$$\Delta R(Q) = \left(a \cdot \frac{1}{Q^c} + b \right) \cdot \Delta Q \quad (5.101)$$

Delta Rate-Quantization Model II

$$\Delta R(Q) = \left(a \cdot \frac{1}{Q^2} + b \cdot \frac{1}{Q} + c \right) \cdot \Delta Q \quad (5.102)$$

where a , b , and c , are the model parameters for each model.

Table 5.34 presents the model fitting results for each studied model and for each test sequence for the delta rate-quantization models for $\Delta Q = 4$ and $\Delta Q = 8$.

Regarding the delta rate-quantization models, it can be seen from Table 5.34 that model I (5.101) and model II (5.102) exhibit similar results in terms of the average fitting error both for $\Delta Q = 4$ and for $\Delta Q = 8$. Notice that for $\Delta Q = 8$ the average fitting error approximately duplicates indicating that as the difference between Q_{ref} and Q_r increases, the delta rate-quantization models proposed become less accurate. However, taking into account the model estimation complexity, it is important to refer that while model I (5.101) is intrinsically nonlinear, thus requiring an iterative nonlinear estimation method, model II (5.102) is a polynomial model that can be easily estimated through linear least squares estimation. Therefore it is possible to conclude that a simple linear model can adequately represent the delta rate-quantization experimental data.

Table 5.34 – Delta rate-quantization average model fitting error results

	SEQ	MODEL			
		I		II	
		$\Delta Q = 4$	$\Delta Q = 8$	$\Delta Q = 4$	$\Delta Q = 8$
QCIF	Foreman	0.009	0.015	0.008	0.014
	Stefan	0.011	0.019	0.010	0.016
	News	0.020	0.029	0.018	0.026
	Kayak	0.007	0.011	0.006	0.010
	M&D	0.009	0.015	0.009	0.013
	Football	0.006	0.008	0.005	0.007
AVG QCIF		0.010	0.016	0.009	0.014
CIF	Foreman	0.006	0.008	0.006	0.007
	Stefan	0.006	0.010	0.006	0.009
	News	0.010	0.016	0.010	0.014
	Kayak	0.004	0.005	0.004	0.004
	M&D	0.005	0.007	0.004	0.007
	Football	0.004	0.006	0.004	0.005
AVG CIF		0.006	0.009	0.006	0.008
AVG QCIF + CIF		0.008	0.012	0.008	0.011

DELTA DISTORTION-QUANTIZATION MODEL

Looking to the actual rate and distortion characteristics, the following $\Delta D(Q)$ models are proposed for study and comparison:

Delta Distortion-Quantization Model I

$$\Delta D(Q) = \begin{cases} 0 & \Leftarrow Q < Q_1 \\ (a \cdot Q + b) \cdot \Delta Q & \Leftarrow Q_1 \leq Q < Q_2, \quad Q_1 = -\frac{b}{a} \text{ and } Q_2 = \frac{c-b}{a} \\ c \cdot \Delta Q & \Leftarrow Q \geq Q_2 \end{cases} \quad (5.103)$$

Delta Distortion-Quantization Model II

$$\Delta D(Q) = (a \cdot Q^2 + b \cdot Q + c) \cdot \Delta Q \quad (5.104)$$

where a , b , and c , are the model parameters for each model.

Table 5.35 presents the model fitting results for each studied model and for each test sequence for the delta distortion-quantization models for $\Delta Q = 4$ and $\Delta Q = 8$.

Regarding the delta distortion-quantization models, it can be seen from Table 5.35 that model I (5.103) exhibits consistently a lower fitting error than model II (5.104) both for $\Delta Q = 4$ and for $\Delta Q = 8$. Moreover, similarly to the delta rate-quantization models, for $\Delta Q = 8$ the average fitting error increases for both models (approximately 35% for model I, and 65% for model II) indicating that as the difference between Q_{ref} and Q_T increases, the delta distortion-quantization models proposed become also less accurate. In this case, however, it is possible to conclude that a simple polynomial model as model II (5.104) does not give the same performance in terms of the modeling error as the nonlinear model, i.e., model I (5.103). Thus a fitting error – modeling complexity trade-off exists.

Table 5.35 – Delta distortion-quantization average model fitting error results

	SEQ	MODEL			
		I		II	
		$\Delta Q = 4$	$\Delta Q = 8$	$\Delta Q = 4$	$\Delta Q = 8$
QCIF	Foreman	0.500	0.763	1.388	2.554
	Stefan	1.056	1.749	2.277	4.188
	News	0.785	1.065	2.523	4.399
	Kayak	0.812	0.971	1.221	1.959
	M&D	0.355	0.464	1.173	2.055
	Football	0.859	0.910	1.044	1.309
AVG QCIF		0.728	0.987	1.604	2.744
CIF	Foreman	0.310	0.414	1.015	1.229
	Stefan	0.540	0.905	1.292	2.325
	News	0.468	0.612	1.341	2.385
	Kayak	0.478	0.535	0.736	0.912
	M&D	0.170	0.232	0.588	1.036
	Football	0.399	0.447	0.646	0.849
AVG CIF		0.394	0.524	0.936	1.456
AVG QCIF + CIF		0.561	0.756	1.270	2.100

5.4 Final Remarks

This chapter considered one fundamental modeling problem of the video coding rate control mechanism: rate-distortion modeling. In this context, a detailed analysis of the theoretical backgrounds of this problem and the practice of applying these concepts in a practical video coding system has been presented. Based on this analysis and the experimental video coding data, a new modeling framework for Intra (I) and Inter (P) coding modes has been proposed, which is composed of different rate and distortion models for each of these coding modes.

The main objective of this work was to develop a family of simple rate and distortion models capable of describing the rate and distortion characteristics of the encoding process as a function of some encoder control parameters, reflecting the lossy encoding rate-distortion trade-off. In this context, three different types of models have been studied and proposed:

- **Rate-quantization models** – These models are particularly useful for constant bit rate video encoding scenarios to compute the quantization parameters to encode each VOP given the bit allocation for the corresponding time instant.
- **Distortion-quantization models** – These models are particularly useful for approximately constant quality encoding scenarios to compute the VOP quantization parameters that lead to a target average VOP distortion.
- **Rate-distortion models** – These models are particularly useful for multiple VO encoding scenarios, where the rate control mechanism must keep the quality among the several VOs approximately constant; in this context these models can be used to guide the bit allocation module in order to produce a bit allocation for the various VOs in the scene that leads to a similar quality.

In the case of Intra coding, the rate and distortion characteristics of the VOP to be encoded depend exclusively on the current quantizer parameter and the VOP statistics, since in this

case the VOP to be encoded does not depend on other (past or future) VOPs; therefore, its rate and distortion characteristics depend exclusively on the current quantizer parameter(s) and VOP statistics. In this context, this chapter proposed, in Section 5.3.1, a set of different $R(Q)$, $D(Q)$, and $R(D)$ models. This work has been presented in [24].

In the case of Inter coding (P-VOPs), the VOP to be encoded does not depend only on its own content but also on its reference VOP(s); therefore, its rate and distortion characteristics depend not only on the quantization parameter(s) and the picture statistics of the current VOP but also on the quantization parameter(s) and picture statistics of the reference VOPs. Consequently, in this case, the rate and distortion models become bidimensional.

Since these bidimensional models are more difficult to obtain, this Thesis proposes a new approach for Inter coding rate and distortion modeling where the rate-quantization and distortion-quantization functions are modeled as stationary functions plus an adaptation model – *the delta model*. In this context, this chapter studied and compared several rate and distortion models in the form of rate-quantization, distortion-quantization, and rate-distortion functions, and proposed, in Sections 5.3.2 and 5.3.3, for each type of models, a model capable of accurately representing the actual Inter rate and distortion characteristics for a selected set of representative sequences. This modeling approach can be specially useful in the context of the rate control mechanism in order to provide more accurate bit allocations and consequently allow to achieve a better rate-distortion trade-off. This work has been presented in [25].

Chapter 6

Rate Control Algorithm for Low-Delay

Video Encoding

6.1 Introduction

As referred in Chapter 3, pleasant visual consumption requires that the video data is coded with approximately constant quality or, at least, with smoothly changing quality. However, due to the varying scene complexity, hybrid video coding schemes, such as the MPEG-4 video codecs, tend to produce a variable number of bits per each encoding time instant, even for slightly changing video quality. In the case of bit rate or delay constrained video encoding, the bit rate variability is handled through a smoothing buffer allowing to achieve a constant average bit rate measured over short intervals of time. Therefore, the maximum bit rate variability is constrained by the smoothing buffer size: a larger buffer will allow larger bit rate variability and, consequently, potentially smoother quality variations. In this context, a generic rate control mechanism involves at first glance, at least, the following steps:

- **Bit allocation** – Assigns an adequate number of bits for short encoding time periods and for each encoding time instant, according to the encoder bit production, the long-term target bit rate, the buffer fullness, and some quality goals.
- **Coding mode control** – Determines the best coding modes and quantization parameters to encode each coding unit (e.g., each MB) to meet the target number of bits and/or target quality.

In the previous chapter, a set of rate-distortion models for Intra and Inter coding were developed aiming at relating either the rate or the distortion with the quantization parameters.

In the ideal case, where the models are extremely accurate, the rate control mechanism could simply use these models to compute the encoder parameters to achieve a desired encoding objective (e.g., average bit rate, target video quality). However, usually, these models tend to deviate from the actual encoder behavior. Therefore, to deal with these deviations between theoretical models and actual coding results, it is necessary to develop: i) adequate compensation mechanisms (e.g., rate control decisions and actions) that are able to track these deviations and compensate them in order to allow a stable and efficient operation of the encoder, and ii) adequate adaptation mechanisms (e.g., estimation of model parameters) that are able to instantaneously represent the actual behavior of the encoder and its rate controller. These two problems, i.e., compensation of the undesired behavior of the encoder and adaptation of the rate-distortion and rate controller parameters, aiming at building a robust and efficient rate control algorithm, constitute the main focus of this chapter.

Control theory provides a very useful framework to solve these problems, respectively, through feedback control mechanisms and parameter estimation. In this context, this chapter proposes a rate control algorithm for object-based video encoding, notably for “low-delay encoding scenarios”, which encompasses these two features: compensation and adaptation.

The chapter is organized as follows: after this introduction, Section 6.2 gives a brief overview of the control process modeling theory; Section 6.3 analyzes some well-known rate control algorithms in terms of their compensation and adaptation mechanisms; Section 6.4 proposes a new rate control algorithm for object-based video encoding for low-delay encoding scenarios, describing its main features and building blocks; Section 6.5 maps the proposed rate control algorithm into the scene-level/object-level rate control framework described in Chapter 3; Section 6.6 highlights the main aspects of the proposed rate control algorithm addressing specifically the spatial and temporal quality constraints; Section 6.7 describes the experimental conditions and presents the results obtained with the proposed algorithm for single and multiple video object encoding; finally, Section 6.8 summarizes the main conclusions and contributions of this chapter.

6.2 Control Approaches for the Bit Rate Control Problem

As referred in Chapter 4, the rate control mechanism is responsible for controlling the encoder in order to produce a set of video elementary streams that do not violate the relevant profile and level constraints, notably the VBV constraints that restrict the production of bits. Obviously, an efficient rate control should allocate the available resources in a way that, besides preventing VBV violation, the subjective quality of the decoded video is also maximized. In this context, it is important to model adequately the rate control process in order to optimize the encoding decisions that will lead to the desired objectives.

With this objective in mind, the bit rate control problem can be seen as a generic control problem where a given process (the video scene encoder) is controlled by a controller mechanism (the rate controller) in order to produce an output that follows closely a desired input command.

6.2.1 Feedback versus Feedforward Control

The straightforward way to perform rate control is by adjusting the encoding parameters for each encoding time based on, for example, the occupancy of the VBV buffer – **feedback rate control**. The basic idea of feedback rate control is to compare the actual result with the desired result and take an action based on the difference, i.e., whenever the past encoding

decisions resulted in an increase of the encoder VBV buffer occupancy, the rate control method has to compensate that behavior by more coarsely encoding the next incoming data, thus decreasing the output rate and consequently increasing the distortion. On the contrary, when the past encoding decisions resulted in a decrease of the encoder VBV buffer occupancy, the rate controller has to compensate that behavior by more finely encoding the next incoming data, thus increasing the output rate and, consequently, decreasing the distortion.

Feedback methods can be implemented easily, since they do not rely on statistics of the incoming data; however, video quality may exhibit large variations along time or even inside each picture depending on the periodicity of the rate controller reaction (i.e., the sampling period of the control mechanism). Moreover, this type of methods may become instable if the amount of compensation is not adequate.

A possible alternative to feedback rate control methods is to control the encoder based on the characteristics of the input data and the desired output result – **feedforward rate control**. The basic idea of feedforward rate control is to plan in advance what will be the encoding result of a certain set of encoding parameters and act before deviations occur, i.e., based on all or a subset of the input data, the rate controller selects the set of encoding parameters that should produce the desired result. Feedforward methods can achieve very tight rate control by processing simultaneously large amounts of data and performing multiple encoding passes. However, besides being computationally very expensive, these methods are not usually suitable for real-time encoding since they generally involve high end-to-end delay. Table 6.1 summarizes the main differences between these two types of rate control.

Table 6.1 – Feedback versus feedforward rate control

Feedback	Feedforward
• Closed loop	• Open loop
• Output driven (reaction)	• Input driven (planning)
• Acts only when there are deviations	• Acts before deviations appear
• Robust to model errors	• Not robust to model errors
• Risk of instability	• No risk of instability

For real-time rate control, notably, for low-delay video encoding scenarios, it is convenient to combine the advantages of both methods, notably the reaction capabilities of feedback rate control methods with the prediction capabilities of feedforward rate control methods. Thus, the rate control algorithm besides taking into account the encoding results of previous time instants and the VBV buffer status, should also take into account the characteristics of the incoming data to better predict future encoding results. In this context, two different tasks can be identified:

- **Generation of a control signal (compensation process¹)** – In the context of this Thesis, the rate control mechanism is responsible for defining the appropriate encoding

¹ According to the Merriam-Webster dictionary, to compensate means “to provide with means of counteracting variation”.

time instants, the MB coding modes, and the MB quantization parameters, i.e., the rate control mechanism is responsible for generating a multidimensional signal that regulates the encoding process, compensating the deviations between the ideal encoder behavior and the actual outcome of the encoding process. The rate control actions (compensation process) depend essentially on the amount of the deviations, i.e., the control error, and the controller parameters estimated during the adaptation process.

- **Estimation of model parameters (adaptation process²)** – In the context of this Thesis, both the encoder and the rate control mechanism are described by a set of models (e.g., in the case of the encoder, the rate-distortion models) including several model parameters that change depending on the operational conditions (e.g., image content, available channel bit rate, VBV buffer occupancy, etc.); consequently, these parameters need to be estimated during encoding to better describe these two entities throughout the changing encoding conditions.

The following sections introduce the basic approaches to implement effectively these two tasks: compensation and adaptation.

6.2.2 Linear Feedback Control

Linear feedback control provides an adequate theoretical background for the rate control problem as expressed by the first task described above.

Figure 6.1 illustrates a generic, linear feedback control model, which consists of a process with one manipulated control variable $u(t)$, one controlled variable $y(t)$, and a reference command variable $u_c(t)$. In this system, the controllable variable $y(t)$ must follow the reference variable $u_c(t)$ as closely as possible, resulting in control errors $e(t) = u_c(t) - y(t)$ that should be as small as possible, i.e., $e(t) \approx 0$. If the reference variable changes with time, the control system is denominated a *tracking control system* [170]. The control block in Figure 6.1 can be described, for example, by the general equation of a Proportional Integral Derivative (PID) controller [170], i.e.,

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (6.1)$$

where $e(t)$, $\int_0^t e(\tau) d\tau$, and $\frac{de(t)}{dt}$ represent, respectively, the instantaneous error at time instant t , the accumulated error until t , and an estimate of the future error.

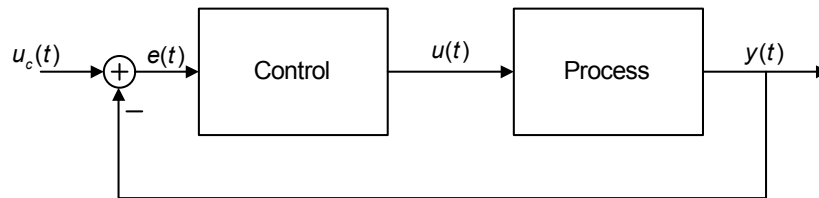


Figure 6.1 – Generic, linear feedback control model

² According to the Merriam-Webster dictionary, to adapt means “to bring one thing into correspondence with another; implies a modification according to changing circumstances”.

Equation (6.1) can be discretized and converted into a difference equation, replacing the integral term by a sum and the derivative term by a first-order difference. For a sampling interval T_0 and rectangular integration, the discrete version of (6.1) is given by

$$u[n] = K_p \left[e[n] + \frac{T_0}{T_i} \sum_{k=0}^n e[k] + \frac{T_D}{T_0} (e[n] - e[n-1]) \right] \quad (6.2)$$

Typically, adopting a linear feedback control model involves selecting a given operation point and designing the controller with constant parameters, e.g., in [171] a simple PI-controller is used to perform VBV control in the context of a rate control mechanism. This approach is adequate for many applications since feedback systems are inherently insensitive to modeling errors and disturbances [172]. This is typically a tuning problem, where the system is viewed as having constant but unknown parameters; thus, controller design consists in the off-line computation of the optimal controller parameters, e.g., PID parameters.

However, sometimes the process dynamics is affected by variations of the operational conditions (e.g., scene changes, channel rate variation, etc.). In this case, constant-gain feedback controllers are usually insufficient due to stability problems, e.g., large picture quality fluctuations or imminent VBV violations.

In this case, it is important to adapt the rate control mechanism to the unknowns of the process. Adaptive controllers can be a good alternative in such cases, where the adaptation process consists in the estimation of the changing parameters during the control process itself.

6.2.3 Adaptive Control

As referred in [172], “an adaptive controller is a controller with adjustable parameters and a mechanism for adjusting the parameters”. Moreover, adaptive control is a special type of nonlinear feedback control where the nonlinear nature of the controller comes precisely from the parameter adjustment mechanism.

An adaptive control system consists usually of two loops: the normal feedback loop with the controller and the process – *inner loop* – and the parameter adjustment loop – *outer loop*. These two loops are usually characterized by two different time scales: a fast time scale for the ordinary feedback and a slower one for updating the parameters. Moreover, it is assumed that there is some kind of feedback from the inner loop performance in reducing the control error for the parameter adjustment loop, in order that the parameter adjustment can lead to a more accurate control.

Therefore, adaptive control becomes a good approach to address simultaneously the two tasks identified in Section 6.2.1, i.e., to generate the video scene encoder control signal (compensation) and to estimate the video encoder and rate controller model parameters (adaptation). There are several possible approaches for implementing an adaptive control mechanism. Below four different types of adaptive schemes are briefly described: gain scheduling, model-reference adaptive control, self-tuning regulators, and dual control [172].

GAIN SCHEDULING

A gain scheduling system can be viewed as a feedback system in which the feedback parameters are adjusted by feedforward compensation [172]. The main advantage of this type of controllers is to reduce the effect of changing operational conditions (e.g., image content, channel rate, VBV buffer occupancy, etc.) by adjusting the controller parameters as functions of those conditions (see Figure 6.2).

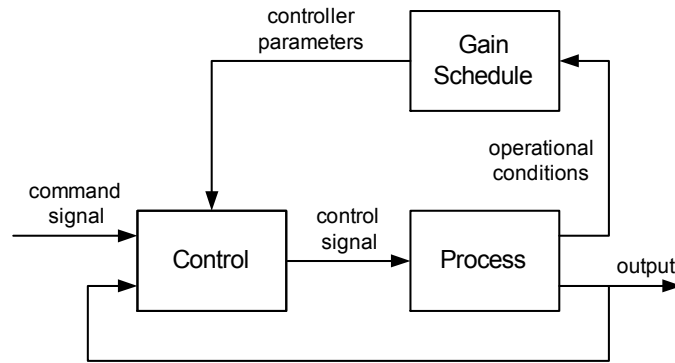


Figure 6.2 – Block diagram of a system with gain scheduling [172]

The main advantage of gain scheduling controllers is the possibility to have different controller behaviors according to the different operational conditions; however, it requires the definition of scheduling variables, i.e., the variables that define the operational conditions, and the computation of the controller parameters for a number of meaningful operational conditions.

MODEL-REFERENCE ADAPTIVE SYSTEMS (MRAS)

In a model-reference adaptive control system, the model (see Figure 6.3) specifies how the process output shall ideally behave in response to a given command signal u_c . Again, in this case, the controller can be viewed as composed by two loops: an ordinary feedback loop consisting of the process and the controller – *inner loop* – and a controller parameter adjustment loop – *outer loop*. Notice that the main objective of the outer loop is to adjust the controller parameters in a way that the difference between the process output, y , and the model output, y_m , is minimized.

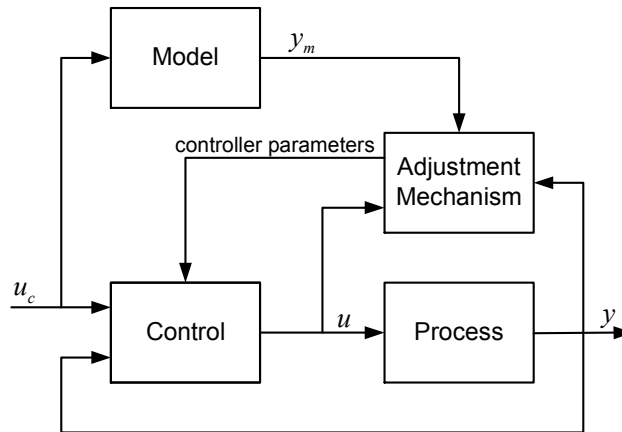


Figure 6.3 – Block diagram of a model-reference adaptive system [172]

A MRAS controller is particularly useful in scenarios where the process to be controlled can be described by a relatively accurate mathematical model; however, the main problem in using a MRAS solution is to develop an adjustment mechanism that is stable in zeroing the error, $e = y - y_m$.

SELF-TUNING REGULATORS (STR)

Gain scheduling and MRAS controllers are usually called direct methods because the adjustment mechanism defines how the controller parameters should be adjusted. However, a more generic case can be identified, where the process parameters are estimated and the controller parameters are computed based on a given controller specification and the estimated process parameters (see Figure 6.4). A control mechanism following this approach is usually called a self-tuning regulator (STR).

Self-tuning regulator systems can be viewed as an automation of process modeling and design, in which the process model and the control design are updated at each sampling period. This approach involves the simultaneous estimation of the process model and the control signal that minimizes the output error.

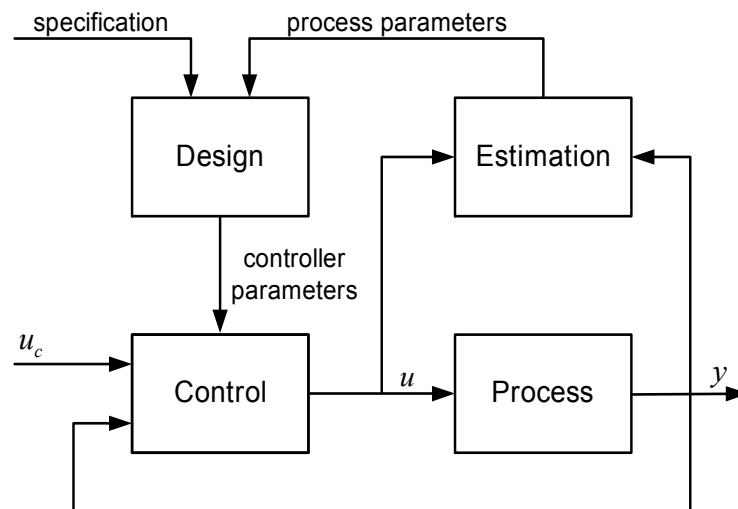


Figure 6.4 – Block diagram of a self-tuning regulator [172]

The main advantage of this approach is the flexibility that it provides; however, it requires an explicit controller behavior specification and involves solving a control design problem for each sampling period.

DUAL CONTROL

Dual control is another possible approach for adaptive control where the process and its parameters are described through a stochastic model, i.e., the parameter estimation uncertainties are also taken into account in the controller design approach, what does not happen for the previous approaches. As referred in [172], the problem of obtaining an adaptive dual controller is an optimal control theory problem that can be solved through nonlinear stochastic control, which besides being a mathematically complex approach is also time-consuming. This is one of the reasons why this approach, although, conceptually useful, has not been successfully used for practical problems [172] and, consequently, does not appear as a suitable approach for a video encoder rate controller for low-delay video encoding, where the computational complexity of the rate control process should be kept low.

6.3 Reviewing Major Rate Control Solutions from a Compensation and Adaptation Perspective

In designing a rate control mechanism, it is important to adopt an adequate control law, i.e., the function that relates the encoder output information with the required reference command signal in order to produce a control signal that adequately drives the encoder, compensating its deviations regarding the ideal behavior.

In order to highlight the mains aspects of such function, the compensation and adaptation mechanisms of several well-known rate control algorithms are analyzed below. In this Thesis, it is proposed to analyze each algorithm by checking the way the following tasks are performed:

- Type of compensation mechanism, e.g., buffer occupancy, target bits, quality, etc.
- Use of feedforward information.
- Sampling period of compensation and adaptation actions.
- Buffer control³.
- Quality control.
- Parameter estimation (adaptation process).

This analysis will be organized according to the algorithm complexity, starting with the less complex algorithms. This analysis does not intend to provide an exhaustive description of each algorithm but to highlight its major control features, highlighting benefits and drawbacks. Notice, that as mention before (see Section 3.2), video coding rate control is not normatively specified in any of the currently available and emerging video coding standards.

6.3.1 H.261 RM8 Rate Control

The ITU-T H.261 [7] Reference Model 8 (RM8) [173] specifies a simple rate control algorithm which aims at controlling the encoder output bit rate by providing basically only encoder rate buffer fullness control, notably to avoid encoder rate buffer overflow.

TYPE OF COMPENSATION MECHANISM

This algorithm relies essentially in a compensation mechanism that relates the encoder rate buffer fullness with the MB quantization parameter through the following equation

$$Q = \min \left\{ \left\lfloor 32 \times \frac{B}{B_s} \right\rfloor + 1, 31 \right\} \quad (6.3)$$

where Q is the quantization parameter ($Q \in \{1, \dots, 31\}$), B is the encoder rate buffer occupancy, and B_s is the buffer size defined as $p \times 6.4$ kbit ($p \in \{1, \dots, 30\}$ defines the channel bit rate as $p \times 64$ kbit/s), corresponding to a maximum buffering delay of 100 ms.

The buffer occupancy is updated at every MB, according to the following recursive equation

³ Since not all rate control algorithms analyzed in this section use a video buffering verifier mechanism to guide their operations, the term buffer control was chosen to refer actions related to the control of an implicit or explicit video rate buffering mechanism.

$$B[i] = B[i-1] + b_{MB}[i] - T_{MB} \quad (6.4)$$

where $b_{MB}[i]$ represents the number of bits used to encode MB i , and T_{MB} represents the average target number of bits per MB that is drained from the buffer.

Equation (6.4) can also be expressed as

$$B[i] = \sum_{k=1}^i (b_{MB}[k] - T_{MB}) \quad (6.5)$$

Substituting (6.5) in (6.3), and forgetting momentarily the nonlinearity introduced by the $\min[\cdot]$ function, leads to the following equation

$$Q[i] = K \times \sum_{k=1}^{i-1} (b_{MB}[k] - T_{MB}) \quad (6.6)$$

which can be seen as an integral feedback law, where Q is the manipulated control variable (u in Figure 6.1), T_{MB} is the reference command variable (u_c in Figure 6.1), b_{MB} is the controllable variable (y in Figure 6.1), $\sum_{k=1}^{i-1} (b_{MB}[k] - T_{MB})$ is the accumulated error, i.e., the integral of the control error, and $K = 32$ is the integral gain.

Notice that, due to the nonlinearities in (6.3), resulting from the finite number of quantizers available, the control error may increase indefinitely (theoretically). This is avoided by skipping information, i.e., when the accumulated error exceeds a certain threshold (in this case, the buffer size B_s), no information is sent for the corresponding MB(s). Furthermore, in order to avoid that the accumulated error becomes negative, which corresponds to an encoder rate buffer underflow, the encoder may add stuffing bits to the buffer.

It is important to notice that the RM8 compensation mechanism has a severe drawback, resulting from the direct relation of the quantization parameter with the buffer occupancy, which can lead to an equilibrium operational point at very high or very low buffer occupancies. In this case, a sudden scene change or a very low scene activity can lead to imminent buffer violations (respectively, an overflow or an underflow), which may require extreme actions to avoid and lead typically to large quality fluctuations.

USE OF FEEDFORWARD INFORMATION

The RM8 rate control algorithm relies exclusively on the feedback information from the buffer occupancy to adjust the quantization parameter; consequently, no feedforward information is used, i.e., the incoming data complexity is not taken into account in the control actions.

SAMPLING PERIOD

There are two sampling periods in this algorithm: the buffer occupancy update sampling period and the MB quantization parameter update sampling period, i.e., the control signal computation sampling period. The first task is performed at the end of each MB encoding, while the second one is performed once every 11 MBs. It is important to notice, however, that the H.261 Recommendation allows changing the quantization parameter at every MB, if desired. The RM8 quantization parameter adjustment period reflects a trade-off between the number of bits necessary to signal the quantization parameter modification and the reaction speed of the controller.

BUFFER CONTROL

The RM8 rate control algorithm is essentially a buffer control mechanism in the sense that the primary rate control goal is to keep the buffer occupancy within permitted bounds. However, two types of buffer control can still be identified in this algorithm: they will be called, in the context of this Thesis, soft buffer control and hard buffer control.

I) Soft Buffer Control

The soft buffer control is provided by the compensation mechanism expressed by equation (6.3), which aims at reducing the number of bits produced by the encoder when the buffer occupancy tends to increase by increasing the quantization parameter and at increasing the number of bits produced by the encoder when the buffer occupancy tends to decrease by decreasing the quantization parameter.

II) Hard Buffer Control

The hard buffer control is provided by the skipping and stuffing mechanisms. As referred above, when the number of bits produced by the encoder leads to an encoder rate buffer overflow, no information is sent, i.e., the MB data is skipped. Conversely, if not enough coded bits are being generated by the encoder, leading to a possible encoder rate buffer underflow, the rate control mechanism inserts stuffing bits in the bitstream in the form of special codewords⁴ that should be discarded by the decoder.

QUALITY CONTROL

This algorithm does not provide any direct or indirect encoding picture quality control, since the quantization parameter adjustment is performed based exclusively on the buffer occupancy and no feedforward information is used. Moreover, no feedback information about the decoded picture quality is used.

PARAMETER ESTIMATION

This algorithm is essentially a constant parameter control algorithm, where the main controller parameter is the integral gain, $K = 32$ in equation (6.6). This constitutes another fundamental drawback of this algorithm since the rate control algorithm does not handle efficiently changes in the operational conditions, such as scene changes or imminent buffer violations.

6.3.2 MPEG-4 VM4 Rate Control

The MPEG-4 Video Verification Model 4.0 (VM4)⁵ [98] specifies a rate control algorithm for independent single video object encoding with arbitrary shape, which aims at controlling the encoder output bit rate by controlling the VOP quantization parameter based on the deviation between the actual number of coded bits used and a pre-defined bit allocation for each VO VOP. This way, the VM4 algorithm circumvents the major drawback of the RM8 algorithm avoiding a direct relation between the encoder output buffer occupancy and the quantization parameter.

⁴ In the case of the H.261 Recommendation, this is accomplished with the MBA stuffing codeword (11 bits).

⁵ This algorithm has been superseded by VM5 [102] and VM8 [108] rate control algorithms analyzed below.

TYPE OF COMPENSATION MECHANISM

The general idea of the VM4 compensation mechanism is to increase the quantization parameter by a given amount, whenever the number of bits spent in the previous VO VOP is higher than the average number of bits available to encode the remaining VO VOPs. Conversely, the quantization parameter is decreased by a certain quantity, whenever the number of bits spent in the previous VOP is lower than the average number of bits available to encode the remaining VOPs.

The VM4 algorithm attempts to allocate uniformly the available number of bits to encode a given sequence of VOPs, T_{SEQ} , over all VOPs of the sequence, irrespective of their coding type. For a sequence of N VOPs, the nominal target number of bits to encode each VOP is then T_{SEQ}/N .

For each encoding time instant, i , the algorithm estimates the actual target number of bits to encode the remaining VOPs as

$$T_{AVG}[i] = \frac{T_{SEQ} - \sum_{k=1}^{i-1} S[k]}{N - i + 1} \quad (6.7)$$

where $S[k]$ is the number of bits spent when encoding VOP k . Equation (6.7) can also be written as

$$T_{AVG}[i] = \frac{T_{SEQ}}{N} + \frac{1}{N - i + 1} \sum_{k=1}^{i-1} \left(\frac{T_{SEQ}}{N} - S[k] \right) \quad (6.8)$$

where the first term represents the nominal VOP target and the second term represents the accumulated VOP bit allocation error (i.e., the integral of the error) multiplied by an integration gain that depends on the encoding time instant. The difference, $\Delta T[i] = T_{AVG}[i] - S[i-1]$, is used as an intermediate control variable to update the quantization parameter for each VOP based on the following equation

$$Q[i] = \begin{cases} Q[i-1] + \Delta Q[i] & \Leftarrow \Delta T[i] < \beta_1 S[i-1] \\ Q[i-1] & \Leftarrow \beta_1 S[i-1] \leq \Delta T[i] \leq \beta_2 S[i-1] \\ Q[i-1] - \Delta Q[i] & \Leftarrow \Delta T[i] > \beta_2 S[i-1] \end{cases} \quad (6.9)$$

where, $\Delta Q[i] = \max(1, \alpha Q[i-1])$ (in [98], $\alpha = 0.1$, $\beta_1 = -3/23$ and $\beta_2 = 3/20$). Notice that the quantization parameter obtained through (6.9) is subsequently limited between 1 and 31.

$\Delta T[i]$ can also be written as a function of the control errors $e[i] = \frac{T_{SEQ}}{N} - S[i]$, as follows

$$\Delta T[i] = e[i-1] + \frac{1}{N - i + 1} \sum_{k=1}^{i-1} e[k] \quad (6.10)$$

where the first term in (6.10) represents the proportional control action with $K_p = 1$ and the second term represents the integral control action with $K_I = 1/(N - i + 1)$.

Notice, however, that the encoder control variable, Q , incorporates also an on-off control action with a dead-zone [174], as expressed by (6.9), i.e., the control signal either remains

unchanged or can only have two different values for each control time instant.

The VM4 quantization parameter update algorithm (6.9) can be seen as a gain scheduling adaptive controller, where the scheduling variables are the quantization parameter, Q , and the output bits, S , and the controller parameters are: ΔQ , $\beta_1 S$ and $\beta_2 S$, which depend on the operational conditions of the encoder (in this case, the quantization parameter and the output number of bits per VOP).

A major drawback of the VM4 algorithm results from the uniform bit allocation assumption, which neglects the VOP coding type, i.e., Intra (I), Inter (P), or Bidirectional (B), which can lead to large quality fluctuations and to instable control due to the frequent and large changes of the quantization parameter.

USE OF FEEDFORWARD INFORMATION

The VM4 rate control algorithm does not use any feedforward information, relying exclusively on the feedback information from the number of encoded bits per VOP and the operational conditions to adjust the quantization parameter.

SAMPLING PERIOD

The original version of the VM4 algorithm [98] does not include a MB quantization parameter update step (i.e., the same quantization parameter is used for the whole VOP), containing only one sampling period, i.e., the VOP period, during which the encoder is controlled and the control parameters are adjusted through a gain scheduling type of approach.

BUFFER CONTROL

The VM4 algorithm does not track directly the encoder rate buffer occupancy status and does not adopt any explicit buffer control mechanism, which means that there is no guarantee that the video buffering verifier constraints are not violated.

QUALITY CONTROL

This algorithm does not provide any direct encoding picture quality control since no feedback information about the decoded picture quality is used to compute the control signal; however, the on-off control action with a dead-zone favors a constant quantization parameter and, consequently, contributes to a more uniform picture quality.

PARAMETER ESTIMATION

The VM4 algorithm establishes a direct relation between the variables that define the operational conditions, Q and S , and the controller parameters, ΔQ , $\beta_1 S$, and $\beta_2 S$, as presented above.

6.3.3 MPEG-2 Video TM5 Rate Control

The MPEG-2 Video Test Model 5 (TM5) [134] rate control algorithm, developed for frame-based video coding, aims essentially at controlling the encoder output bit rate through an adaptive picture bit allocation step, followed by a MB quantization parameter adjustment step. This algorithm avoids the main drawback of the VM4 algorithm uniform bit allocation, by allocating to each picture a number of bits proportional to its estimated complexity (depending, essentially, on the picture coding type).

TYPE OF COMPENSATION MECHANISM

To achieve its goal, the TM5 rate control algorithm uses three levels of feedback: I) GOP-level, II) Picture-level (a picture here is the same as a frame), and III) MB-level.

I) GOP-level Feedback

In this algorithm, the sequence of video frames is divided in GOPs, i.e., groups of pictures starting with an I-picture followed, typically, by a regular repetitive structure of P- and B-pictures. The algorithm aims at allocating a number of bits to each GOP that is proportional to the GOP duration, according to the following equation

$$T_{GOP}[i] = R \times t_{GOP}[i] + T_{GOP}[i-1] - S_{GOP}[i-1] \quad (6.11)$$

where R is the channel bit rate in bit/s, $t_{GOP}[i]$ is the GOP i duration in seconds, $T_{GOP}[i-1]$ is the GOP $i-1$ target number of bits, and $S_{GOP}[i-1]$ is the actual number of bits used to encode all pictures of GOP $i-1$.

Assuming that the bit budget to encode a complete sequence is T_{SEQ} , and the sequence is divide into N_{GOP} GOPs of equal duration, then equation (6.11) can be written as

$$T_{GOP}[i] = \frac{T_{SEQ}}{N_{GOP}} + \sum_{k=1}^{i-1} \left(\frac{T_{SEQ}}{N_{GOP}} - S_{GOP}[k] \right) \quad (6.12)$$

where the first term represents the nominal GOP target number of bits and the second term represents the accumulated GOP bit allocation error (i.e., the integral of the error). Therefore, at the GOP-level, the TM5 rate control algorithm uses an integral feedback law with an integration gain $K_I = 1$, which means that, at the GOP-level, the algorithm attempts to react immediately to deviations regarding the nominal target by distributing the accumulated bit allocation error by the next GOP to be encoded.

II) Picture-level Feedback

In order to obtain approximately constant picture quality along time, the TM5 algorithm assumes, at the picture-level, that each picture should get a fraction of the GOP target number of bits proportional to the picture type complexity, i.e., the relation between the nominal number of bits targets assigned to each picture encoding type, respectively T_I , T_P , and T_B , are the following

$$\frac{T_I}{X_I} = \beta_P \frac{T_P}{X_P} = \beta_B \frac{T_B}{X_B} \quad (6.13)$$

with

$$T_I + N_P T_P + N_B T_B = T_{GOP} \quad (6.14)$$

where X_I , X_P , and X_B , are the corresponding picture type estimated complexity; β_P and β_B are adjustment parameters depending only on the quantization matrices used (in [134], $\beta_P = 1.0$ and $\beta_B = 1.4$); and N_P and N_B are, respectively, the number of P- and B-pictures in the GOP.

Therefore, at the beginning of each GOP, the nominal targets for each picture encoding type, \bar{T}_I , \bar{T}_P , and \bar{T}_B , can be computed combining (6.13) and (6.14) as follows

$$\begin{aligned}
 \bar{T}_I &= \frac{T_{GOP}}{X_I + N_P \frac{X_P}{\beta_P} + N_B \frac{X_B}{\beta_B}} X_I \\
 \bar{T}_P &= \frac{T_{GOP}}{X_I + N_P \frac{X_P}{\beta_P} + N_B \frac{X_B}{\beta_B}} \frac{X_P}{\beta_P} \\
 \bar{T}_B &= \frac{T_{GOP}}{X_I + N_P \frac{X_P}{\beta_P} + N_B \frac{X_B}{\beta_B}} \frac{X_B}{\beta_B}
 \end{aligned} \tag{6.15}$$

In order to cope with deviations between the nominal targets and the actual number of bits spent, the target number of bits to encode the next picture in the GOP is adjusted for each picture encoding type as follows:

$$\begin{aligned}
 T_P &= \frac{R_{GOP}}{n_P \frac{X_P}{\beta_P} + n_B \frac{X_B}{\beta_B}} \frac{X_P}{\beta_P} \\
 T_B &= \frac{R_{GOP}}{n_P \frac{X_P}{\beta_P} + n_B \frac{X_B}{\beta_B}} \frac{X_B}{\beta_B}
 \end{aligned} \tag{6.16}$$

where R_{GOP} is the remaining number of bits to encode the remaining GOP pictures (the target for the first I-picture of the GOP corresponds to the nominal target); and n_P and n_B are, respectively, the number of remaining P- and B-pictures to encode in the GOP. In TM5 [134], the targets given by (6.16) have a minimum value of $bit_rate / (8 \times picture_rate)$ bits to avoid large picture quality variations.

To illustrate the behavior of this picture bit allocation mechanism in terms of feedback compensation, the coding cost of picture k in the GOP is defined as C_k , with $C_I = X_I$, $C_P = X_P / \beta_P$, and $C_B = X_B / \beta_B$, and

$$C_{GOP} = \sum_{k=1}^N C_k \tag{6.17}$$

In this case, the target number of bits to encode each picture of the GOP can be written as

$$T[i] = \frac{R_{GOP}[i]}{\sum_{k=i}^N C_k} C_i \tag{6.18}$$

Since $R_{GOP}[i] = T_{GOP} - \sum_{k=1}^{i-1} S[k]$, (6.18) can be rewritten as

$$T[i] = \frac{T_{GOP}}{C_{GOP}} C_i + \frac{C_i}{\sum_{k=i}^N C_k} \sum_{k=1}^{i-1} \left(\frac{T_{GOP}}{C_{GOP}} C_k - S[k] \right) \tag{6.19}$$

where $S[k]$ represents the number of bits used to encode picture k . The first term of (6.19)

represents the nominal target to encode each picture, i.e., the command signal, which is proportional to the picture encoding complexity. Since the command signal is not constant, the picture-level control can be seen as a tracking control system (see Section 6.2.2). The second term of (6.19) represents the compensation to the deviations between the actual encoding results and the nominal target. Since the compensation term depends on the accumulated error up to picture $i-1$, the control action expressed by (6.19) configures an integral type of controller.

Notice that while at the GOP-level, $K_I = 1$, at picture-level, $K_I < 1$, which leads to a slower reaction to deviations between the nominal target and the actual number of encoded bits, since the accumulated error is distributed by the remaining pictures to encode, according to their encoding type complexity.

III) MB-level Feedback

At the MB-level, the TM5 algorithm uses a feedback law that relates a reference MB quantization parameter, Q , with the occupancy of a virtual buffer, according to the following equation

$$Q[i] = Q_0 + K \sum_{k=1}^{i-1} (b_{MB}[k] - T_{MB}) \quad (6.20)$$

where Q_0 is the reference quantization parameter of the last encoded MB of the previous encoded picture of the same type; $b_{MB}[k]$ is the number of bits generated by MB k ; T_{MB} is the target number of bits per MB, i.e., the target number of bits for the picture divided by the number of MBs in the picture, N_{MB} ; and K is a constant that depends on the channel bit rate and the picture frame rate, according to the following relation

$$K = 31 \frac{\text{picture_rate}}{2 \times \text{bit_rate}} \quad (6.21)$$

Equation (6.20) represents an integral feedback law, since it relates directly the accumulated MB bit allocation error with the MB quantization parameter. It is important to notice that, as in the RM8 and VM4 algorithms, the TM5 MB-level feedback law assumes a uniform MB bit allocation within the picture.

USE OF FEEDFORWARD INFORMATION

The original TM5 rate control algorithm [134] relies mainly on the feedback information from the number of bits generated at each level, and some parameters estimated from previous encoding time instants, such as the picture type complexities and the MB average activity, described below. However, in order to take into account the lower sensitivity of the human visual system to the quantization error in highly texture zones and the higher sensitivity to the quantization error in uniform zones, the quantization parameter is further feedforward adjusted to the local characteristics of the picture using the MB texture activity computed from the luminance information of the MB being encoded, according to the following equation

$$Q_{MB}[i] = Q[i] \times \bar{A}_{MB}[i] \quad (6.22)$$

where $Q_{MB}[i]$ is the actual MB quantization parameter, $Q[i]$ is the reference quantization parameter obtained through (6.20), and $\bar{A}_{MB}[i]$ is the MB normalized activity computed

according to the following equations

$$\bar{A}_{MB} = \frac{2 \times A_{MB} + \mu_A}{A_{MB} + 2 \times \mu_A} \quad (6.23)$$

where μ_A is the average MB activity of the last encoded picture and A_{MB} is the MB activity computed as

$$A_{MB} = 1 + \min_{b=1,\dots,8} (\sigma_b^2) \quad (6.24)$$

where σ_b^2 is the variance of each of the eight 8×8 luminance sub-blocks associated to a MB, i.e., four sub-blocks frame-organized and four sub-blocks field-organized⁶, computed as

$$\sigma_b^2 = \frac{1}{64} \sum_{k=1}^{64} (p_k - \mu_b)^2 \quad (6.25)$$

where p_k is the luminance value of pixel k , and μ_b is the average luminance value in block b .

SAMPLING PERIOD

As referred above, the TM5 algorithm uses three embedded feedback loops, each corresponding to a different sampling period: GOP, Picture, and MB.

BUFFER CONTROL

This algorithm does not perform explicit buffer control. Moreover, the algorithm does not track the occupancy of the actual encoder rate buffer. Therefore, the TM5 algorithm cannot guarantee full compliance with the video buffer verifier mechanism. This fact constitutes one of the main limitations of this algorithm if used in real implementations.

QUALITY CONTROL

The TM5 algorithm does not provide an explicit encoding picture quality control since no feedback information about the decoded picture quality is used to compute the control signal. However, the MB quantization adaptation based on the MB normalized activity tends to coarsely mimic the human visual system perception by favoring higher distortions in highly textured zones and lower distortions in more uniform zones.

PARAMETER ESTIMATION

The TM5 algorithm requires the estimation of two types of parameters: the picture type complexities, i.e., X_I , X_P , and X_B , and the MB average activity.

The picture type complexity parameter, X_T , for the current picture of type $T \in \{I, P, B\}$ is estimated from the results of the previous encoded picture of the same type as

$$X_T = S_T \times Q_T \quad (6.26)$$

where S_T is the number of bits generated by encoding the previous picture of type T , and Q_T

⁶ A frame-organized sub-block is a 8×8 pixels sub-block that contains lines alternating from top field and bottom field; a field-organized sub-block is a 8×8 pixels sub-block that contains lines only from one field (either from top field or bottom field).

is the average MB quantization parameter over all MBs of that picture. Alternatively, the picture type complexity could be estimated directly from the encoding results of each picture; however, this would require, at least, two encoding passes for each picture in the GOP, which is usually not suitable for real-time applications both in terms of computational complexity and delay.

The MB average activity parameter for the current picture is estimated as the average MB activity of the last encoded picture. It is important to notice however that, in this case, this limitation is introduced exclusively to reduce the computational complexity of the algorithm since it avoids processing the whole picture before encoding it.

6.3.4 MPEG-4 Visual Annex L Rate Control

As mention in Chapter 2, the MPEG-4 Visual standard [29] includes on its Annex L, an informative description of a rate control mechanism composed of three main algorithms: 1) Frame rate control, 2) MVO rate control, and 3) MB rate control

The frame rate control algorithm can be applied for SVO and independent MVO encoding. The MVO rate control algorithm is an extension of the frame rate control algorithm for jointly encoding multiple VOs. These algorithms became stable in the MPEG-4 Video Verification Model 5.0 (VM5) [102] since they showed superior performance (better bit rate control) and extra functionalities (MVO rate control) relatively to VM4 rate control.

The frame rate control and multiple object rate control algorithms use a fixed quantization parameter for all the MBs in a VOP, which is computed based on a given target number of bits and the corresponding VO rate-quantization model. However, in certain situations, notably in low-delay video encoding, using a fixed quantization parameter for all MBs in a VOP may lead frequently to imminent buffer violations due to eventual large deviations between the actual and the estimated number of encoded bits. Therefore, the MB rate control algorithm attempts to circumvent that problem by providing a way to achieve a more accurate rate control, allowing the quantization parameter to change from MB to MB inside a VOP. This algorithm became stable in the MPEG-4 Video Verification Model 8.0 (VM8) [108].

TYPE OF COMPENSATION MECHANISM

The general idea of these algorithms is to control the output bit rate by allocating an appropriate number of bits to each encoding unit (e.g., VOP or MB) and computing the corresponding quantization parameters through rate-quantization models. To cope with deviations between the intended and the actual results, these algorithms adjust the bit allocation at each VOP through a feedback mechanism. Moreover, when the MB rate control is used, a fine adjustment is performed at the MB-level.

I) VOP-level Feedback

As in VM4, the frame rate control algorithm attempts to allocate uniformly the available number of bits to encode a given sequence of VOPs, T_{SEQ} , over all VOPs of a given VO irrespective of their coding type. For a sequence of N VOPs, the nominal target number of bits to encode each VOP is then T_{SEQ}/N . However, in this case, for each encoding time instant, i , the algorithm estimates the target number of bits to encode the remaining VOPs, based on the number of bits left to encode the remaining VOPs and the number of bits generated in the previous encoding time instant as

$$T[i] = (1 - \alpha) \frac{T_{SEQ} - \sum_{k=1}^{i-1} S[k]}{N - i + 1} + \alpha S[i - 1] \quad (6.27)$$

where α is an algorithm parameter (in [29], $\alpha = 0.05$).

The rationale for equation (6.27) is that, if the previous VOP was complex and used a higher than foreseen number of bits, then the number of bits to encode the upcoming VOP should be increased relatively to the nominal target number of bits. However, since the remaining number of bits is lower, fewer bits can be allocated to this VOP. The weighted average controlled by α reflects this trade-off. In [29], in order to guarantee a minimum quality, a lower bound of $R_s/30$ bits is allocated to each VOP, where R_s is the bit rate for the sequence, independently of the source temporal resolution.

Equation (6.27) can also be written as

$$T[i] = \frac{T_{SEQ}}{N} + \frac{1 - \alpha}{N - i + 1} \sum_{k=1}^{i-1} \left(\frac{T_{SEQ}}{N} - S[k] \right) - \alpha \left(\frac{T_{SEQ}}{N} - S[i - 1] \right) \quad (6.28)$$

Equation (6.28) expresses a proportional-integral feedback compensation law, where the first term represents the command signal, i.e., the nominal target, the second term represents the integral action with $K_I = (1 - \alpha)/(N - i + 1)$, and the third term represents the proportional action with $K_P = -\alpha$.

II) MB-LEVEL FEEDBACK

When the MB rate control algorithm is used, another compensation mechanism is used to provide a fine control of the number of bits generated while encoding the corresponding VOP. In this case, the fraction of bits assigned to each MB is controlled by its coding cost, expressed through its prediction error mean absolute difference (MAD), according to the following equation

$$T_{MB}[i] = \frac{\omega_i \cdot MAD_i}{C_i} \left(T_{VOP} - \sum_{k=1}^{i-1} b_{MB}[k] \right) \quad (6.29)$$

where ω_i is a MB weight that can be made dependent on the MB perceptual importance (in [29], $\omega_i = 1.0$), C_i is the remaining accumulated coding cost of the MBs still to be encoded, defined by (6.30), T_{VOP} is the target number of bits for encoding the current VOP, and $b_{MB}[k]$ is the number of bits generated when coding MB k .

$$C_i = \sum_{k=i}^{N_{MB}} \omega_k \cdot MAD_k \quad (6.30)$$

Equation (6.29) can be rewritten as

$$T_{MB}[i] = \frac{\omega_i \cdot MAD_i}{C_{VOP}} T_{VOP} + \frac{\omega_i \cdot MAD_i}{C_i} \sum_{k=1}^{i-1} \left(\frac{\omega_k \cdot MAD_k}{C_{VOP}} T_{VOP} - b_{MB}[k] \right) \quad (6.31)$$

where C_{VOP} , defined by (6.32), is the coding cost of the N_{MB} MBs in the current VOP, i.e.,

$$C_{VOP} = \sum_{k=1}^{N_{MB}} \omega_k \cdot MAD_k \quad (6.32)$$

Notice that, in this case, the bit allocation inside each VOP is not uniform but depends on the MB complexity. Moreover, the feedback compensation mechanism defined by equation (6.31) can be seen as a tracking control system with integral action, since the command signal changes along time and the compensation mechanism relies on the accumulated error.

USE OF FEEDFORWARD INFORMATION

The MPEG-4 Visual Annex L rate control algorithm uses feedforward information into three different situations: 1) to compute the quantization parameters in the frame rate control; 2) to allocate the VOP target among the several MBs in the VOP in the MB rate control; and 3) to distribute the allocated bits among the several VOs in the MVO rate control.

In the first case, the algorithm uses the VOP MAD computed during the motion estimation step in the VOP rate-quantization model to compute the quantization parameter to encode the VOP.

In the second case, the MB MAD is used in the MB-level feedback law to perform the MB bit allocation, and in the MB rate-quantization model to compute the MB quantization parameter.

Finally, in the third case, the MVO rate control algorithm uses feedforward information about the size, motion vectors amplitude, and VOP MAD to distribute the allocated number of bits for a given encoding time instant among the several VOs to encode for that time instant. Since the different VOPs to encode may have different sizes, and different complexities, it is important to take that into account. In this case, this is done by assigning a weight to each VOP

$$\omega_i = \omega_M \cdot M_i + \omega_S \cdot S_i + \omega_V \cdot V_i \quad (6.33)$$

where M_i , S_i , and V_i are, respectively, the normalized motion, size and variance (MAD^2) of VOP i , and ω_M , ω_S , and ω_V are weights that control the importance of the different types of data in the target number of bits distribution. Other types of data can also be considered for distributing the bit rate, such as the VO priority and the VOP coding mode [14].

In [29], the values of ω_M , ω_S , and ω_V depend on the mode of operation of the algorithm. The MVO rate control defines two modes of operation: LowBitRate and HighBitRate; these modes depend on the number of encoding time instants skipped since the last encoding time instant. If the number of skipped time instants is higher than two, the algorithm operates in the LowBitRate mode; otherwise, it operates in HighBitRate mode. For the LowBitRate mode, $\omega_M = 0.6$, $\omega_S = 0.4$, and $\omega_V = 0.0$; for the HighBitRate mode, $\omega_M = 0.25$, $\omega_S = 0.25$, and $\omega_V = 0.50$.

The target number of bits for a given encoding time instant is, finally, distributed between each VOP to encode using the weight computed through (6.33) as

$$T_i = \omega_i \cdot T \quad (6.34)$$

Notice that, in the MPEG-4 Visual Annex L MVO rate control algorithm, there is no feedback for the distribution of the target number of bits among the several VOs in the scene, which means it is a pure input driven strategy with no compensation mechanism. This is a major drawback of this algorithm that will be tackled in this Thesis (see Section 6.4.5).

SAMPLING PERIOD

When the MB rate control is not used, the MPEG-4 Visual Annex L rate control algorithm uses only a feedback loop at the VOP-level. If the MB rate control is activated, the algorithm uses two embedded feedback loops, corresponding to two different sampling periods: VOP-level and MB-level.

BUFFER CONTROL

The MPEG-4 Visual Annex L rate control algorithms use two types of buffer control, called in the context of this Thesis, soft buffer control and hard buffer control.

I) Soft Buffer Control

In this case, the algorithm targets a buffer occupancy equal to half the encoder rate buffer size, this immediately before encoding a given VOP. To achieve this goal, the VOP target obtained through (6.27) is further adjusted according to the following equation

$$T_{SBC}[i] = T[i] \frac{B[i] + 2(B_s - B[i])}{2B[i] + (B_s - B[i])} \quad (6.35)$$

where $B[i]$ represents the buffer occupancy for time instant i and B_s the buffer size.

Notice, that equation (6.35) embeds another feedback compensation loop with a command signal equal to $B_s/2$ and an error signal equal to $B[i] - B_s/2$, which can be more clearly viewed if, alternatively, (6.35) is written as

$$T_{SBC}[i] = T[i] \frac{3B_s + 2(B_s/2 - B[i])}{3B_s - 2(B_s/2 - B[i])} \quad (6.36)$$

or equivalently

$$T_{SBC}[i] = T[i] \frac{3 + e[i]}{3 - e[i]} \quad (6.37)$$

with $e[i] = \frac{B_s/2 - B[i]}{B_s/2}$ being the normalized buffer occupancy error, i.e., $e[i] \in [-1, 1]$.

As can be inferred from (6.37), if $B[i] = B_s/2$, then $T_{SBC}[i] = T[i]$, which means that the control error $e[i] = 0$ and no compensation is needed for that time instant.

Notice that (6.37) configures a nonlinear feedback law with a steady state equal to half the buffer size. The goal of having for each encoding time instant a buffer occupancy equal to half the buffer size, as expressed by (6.37), means that encoder rate buffer overflows and encoder rate buffer underflows are treated equally. However, in real encoding situations, encoder rate buffer underflows are typically easy to avoid through stuffing data, while encoder rate buffer overflows require more extreme measures, such as skipping data, which may lead to subjectively annoying artifacts. Moreover, setting the target buffer occupancy independently of the coding modes of the VOPs being encoded and their relative position with respect to the beginning of the GOV, may be inappropriate. These are major drawbacks of this algorithm that will be tackled in this Thesis by proposing an algorithm for soft buffer control that defines a target buffer occupancy depending on the content complexity and the relative encoding time instant with respect to the beginning of the GOV. This algorithm will be presented in Section 6.4.6.

II) Hard Buffer Control

In some situations, the soft buffer control may not be sufficient to guarantee that the video buffering verifier mechanism is not violated. Therefore, whenever the target bit allocation given by (6.37) added to the occupancy exceeds a certain threshold (overflow threshold), the bit allocation is decreased by the amount in excess; similarly, if the foreseen encoder occupancy is below a certain threshold (underflow threshold), before the next encoding time instant, the bit allocation is increased by the corresponding amount, i.e.,

$$T_{HBC}[i] = \begin{cases} \beta B_s - B[i] & \Leftarrow B[i] + T_{SBC}[i] > \beta B_s \\ R_p + (1 - \beta) B_s - B[i] & \Leftarrow B[i] + T_{SBC}[i] - R_p < (1 - \beta) B_s \end{cases} \quad (6.38)$$

where β is a control parameter (in [29], $\beta = 0.9$) and R_p is the number of bits drained from the buffer between two consecutive encoding time instants.

It is important to highlight, however, that even the hard buffer control mechanism expressed by (6.38) cannot guarantee full VBV compliance, since the sampling period for this mechanism is the VOP encoding period. Therefore, reactions at the end of the VOP encoding may not be sufficient to avoid buffer violations. Moreover, in [29], no stuffing strategy is proposed, which means that for scenes with little motion the encoder can easily experience encoder rate buffer underflows.

Additionally to the buffer control action expressed by (6.38), if after each encoding time instant the buffer occupancy is above a certain threshold, the next VOP(s) will be skipped until the buffer occupancy reaches again a nominal point of operation, in order to prevent encoder rate buffer overflow for the upcoming VOPs. This means that, while $B[i] > \delta B_s$ (in [29], $\delta = 0.8$), the next VOP to be encoded is skipped and the buffer occupancy is updated as $B[i+1] = B[i] - R_p$.

QUALITY CONTROL

The MPEG-4 Visual Annex L rate control algorithms do not provide an explicit encoding picture quality control since no feedback information about the decoded picture quality is used to compute the control signal. However, two mechanisms can be identified in these algorithms that aim to provide some indirect quality control.

The first mechanism, used in the frame rate control, is the quantization parameter variation range limitation between consecutive VOPs of the same VO to avoid large picture quality variations between consecutive encoding time instants, i.e.,

$$Q[i] \in [\max(1, (1 - \gamma)Q[i-1]), \min(31, (1 + \gamma)Q[i-1])] \quad (6.39)$$

where $Q[i]$ is the quantization parameter for the current VOP, $Q[i-1]$ is the quantization parameter of the previous VOP, and γ is a control parameter (in [29], $\gamma = 0.25$).

The second mechanism, used in the MVO rate control, consists in skipping VOPs in order to achieve a better trade-off between motion smoothness and spatial quality. When the bit rate resources are scarce, notably in low bit rate encoding, the target number of bits for the current time instant that emerges from the buffer control may not be sufficient to even encode the auxiliary data (i.e., header, motion vector, and shape data); thus, no bits would be left for encoding the texture data. In these cases, the MVO rate control algorithm issues an alert for the following algorithmic steps to signal this scarceness in the form of pre-encoding skip information, i.e., the number of encoding time instants to skip before encoding the next

VOP(s). The rate control algorithm estimates the number of encoding time instants to skip, $N_{pre_skip}[i]$, such that

$$T_{HBC}[i] - H[i-1] + N_{pre_skip}[i] \cdot R_p \geq \varepsilon \quad (6.40)$$

where $T_{HBC}[i]$ is the target number of bits for the current encoding time instant resulting from the hard buffer control, $H[i-1]$ is the total number of bits used in the previous encoding time instant for encoding the auxiliary data (i.e., header, motion vector, and shape data), R_p is the number of bits drained from the buffer between two consecutive encoding time instants, and ε is a control threshold greater or equal than zero.

PARAMETER ESTIMATION

The MPEG-4 Visual Annex L rate control algorithms require the estimation of the rate-quantization parameters.

I) VOP Rate-Distortion Model Parameters

In the case of the frame rate control algorithm, the rate-quantization model used is described by the following equation

$$R(Q) = \left(X_1 \frac{1}{Q} + X_2 \frac{1}{Q^2} \right) \cdot MAD \quad (6.41)$$

where X_1 and X_2 are the model parameters, which are estimated by linear least squares estimation after each encoding time instant. A first estimate of the model parameters is obtained through the minimization of

$$\chi^2 = \sum_{k=i-w[i]+1}^i \left(y[k] - (X_1 + X_2 x[k]) \right)^2 \quad (6.42)$$

with $y[i] = \frac{S[i] - H[i]}{MAD[i]} Q[i]$ and $x[i] = \frac{1}{Q[i]}$, where $S[i]$ represents the total number of bits used to encode VOP i ; $H[i]$ represents the header, motion vector, and shape (if applicable) bits; and $w[i] = \min[i, w_{\max}]$ is a sliding window size that controls the number of data points used to estimate the model parameters (in [29], $w_{\max} = 20$). Notice that, since the header, motion vector, and shape bits do not directly depend on the quantization parameter used, they are not taken into account to estimate the rate-quantization model parameters.

If the encoding VOP MAD changes significantly, a smaller window with the more recent data points is used, i.e.,

$$w[i] = \begin{cases} \min \left[\left\lceil \frac{MAD[i]}{MAD[i-1]} w[i-1] + 1 \right\rceil, w_{\max} \right] & \Leftarrow MAD[i] < MAD[i-1] \\ \min \left[\left\lceil \frac{MAD[i-1]}{MAD[i]} w[i-1] + 1 \right\rceil, w_{\max} \right] & \Leftarrow MAD[i] \geq MAD[i-1] \end{cases} \quad (6.43)$$

In order to avoid model biasing due to outlier data points, a second estimate of the model parameters is obtained, considering now only the data points for which the prediction error is less than a rejection threshold, i.e., the data points that verify the following condition

$$\left| (S[k] - H[k]) - \left(X_1 \frac{1}{Q[k]} + X_2 \frac{1}{Q^2[k]} \right) MAD[k] \right| < \sqrt{\frac{\sigma^2[i]}{w[i]}} \quad (6.44)$$

where

$$\sigma^2[i] = \sum_{k=i-w+1}^i \left[(S[k] - H[k]) - \left(X_1 \frac{1}{Q[k]} + X_2 \frac{1}{Q^2[k]} \right) MAD[k] \right]^2 \quad (6.45)$$

Notice that a new sliding window size is also computed for the second estimation step that directly derives from the number of data points in the previous window verifying (6.44).

It is important to highlight that the window adaptation mechanism specified by (6.43) leads usually to sudden changes of the window size after scene changes, which may lead to instabilities in the parameter estimation.

II) MB Rate-Distortion Model Parameters

The MB rate control assumes that the encoder rate-quantization function can be modeled as

$$R_{MB}(Q) = \begin{cases} A_1 \frac{1}{Q^2} MAD_{MB}^2 & \Leftarrow R_{bpp} > \varepsilon_T \\ \left(A_2 \frac{1}{Q^2} + A_3 \frac{1}{Q} \right) MAD_{MB} & \Leftarrow R_{bpp} \leq \varepsilon_T \end{cases} \quad (6.46)$$

where R_{MB} represents the number of bits to encode the MB texture data, A_1 , A_2 , and A_3 are the model parameters, Q is the MB quantization parameter, MAD_{MB} is the mean absolute difference for the MB, R_{bpp} is the target number of bits per pixel for encoding the MB texture data, and ε_T is a control parameter (in [29], $\varepsilon_T = 0.085$).

After encoding MB i , the rate-quantization model parameters A_1 , or A_2 and A_3 , are updated based on the encoding results for the current and previous MBs.

If $b_{MB}[i] = 0$ (i.e., no texture data bits were produced), only the number of MBs without texture data bits, n_s , is updated, i.e., $n_s = n_s + 1$; otherwise, the two situations of (6.46) should be considered:

1. For $R_{bpp} > \varepsilon_T$, only parameter A_1 is updated, i.e.,

$$A_1 = A'_1 \times \beta + A_1^{prev} \times (1 - \beta) \quad (6.47)$$

with $\beta = i / N_{MB}$, $A'_1 = \hat{A}_1 \frac{1}{i - n_s} + A'_1 \frac{i - n_s - 1}{i}$ and $\hat{A}_1 = \frac{b_{MB}[i]}{MAD_{MB}^2[i]} Q^2[i]$, and A_1^{prev}

being the last encoded VOP model parameter (in [29], for the first VOP of a given VO, $A_1^{prev} = 100$).

2. For $R_{bpp} \leq \varepsilon_T$, only parameters A_2 and A_3 are updated; in this case, A_2 and A_3 are estimated through linear least squares estimation, similarly to the frame rate control model parameter estimation, using a sliding window containing the last w MBs for which $b_{MB} > 0$, with $w_{max} = 20$.

6.4 Proposal for a Low-Delay Rate Control Algorithm

This section proposes a new rate control algorithm for SVO and MVO constant bit rate encoding under low-delay constraints. This rate control algorithm follows the architecture presented in Figure 6.5, where the proposed rate controller, corresponding to the major contribution of this Thesis, is responsible for jointly controlling the encoding of multiple video objects composing a given scene, in order to meet the relevant constraints of the underlying encoding scenario. This architecture is composed of three major building blocks:

- **Scene Encoder** – Responsible for encoding the original video content (i.e., the video objects composing the scene) into a set of bitstreams (one for each VO⁷). This block is composed by a scene buffer where the original VOPs are stored; a symbol generator that reduces the redundancy and irrelevancy of the original video data generating adequate coded symbols; an entropy coder that reduces the statistical redundancy of the coded symbols converting them into bit codes; and, finally, a video multiplexer responsible for organizing the coded symbols according to the adopted video coding syntax.
- **Video Buffering Verifier** – A series of normative models, each one defining rules and limits to verify if the amount required for a specific type of decoding resource is within the bounds allowed by the corresponding profile and level specification (see Chapter 4). The rate control algorithm must use these rules and limits to define the control actions that will drive the scene encoder without violating this mechanism.
- **Rate Controller** – The mechanism responsible for controlling the scene encoder aiming at efficiently encoding the original video data while producing a set of bitstreams that does not violate the video buffering verifier mechanism. Essentially, this mechanism is composed by six modules (described below) that, based on statistics computed from the input data stored in the scene buffer (feedforward information) and statistics computed through the different video buffering verifier models (feedback information), compute a multidimensional control signal (e.g., encoding time instants, MB coding modes and quantization parameters) that will command the encoding process.

The performance of this rate control algorithm will be analyzed later in this Thesis (see Section 6.7), in comparison to relevant alternatives, regarding its ability to meet the relevant rate control objectives, notably, not violating the video buffering verifier constraints, keeping the VBV buffer close to its target occupancy, and, not the least, its ability to maximize the output video quality without much variations.

⁷ In case scalability is used, one bitstream for each VOL of each VO is generated.

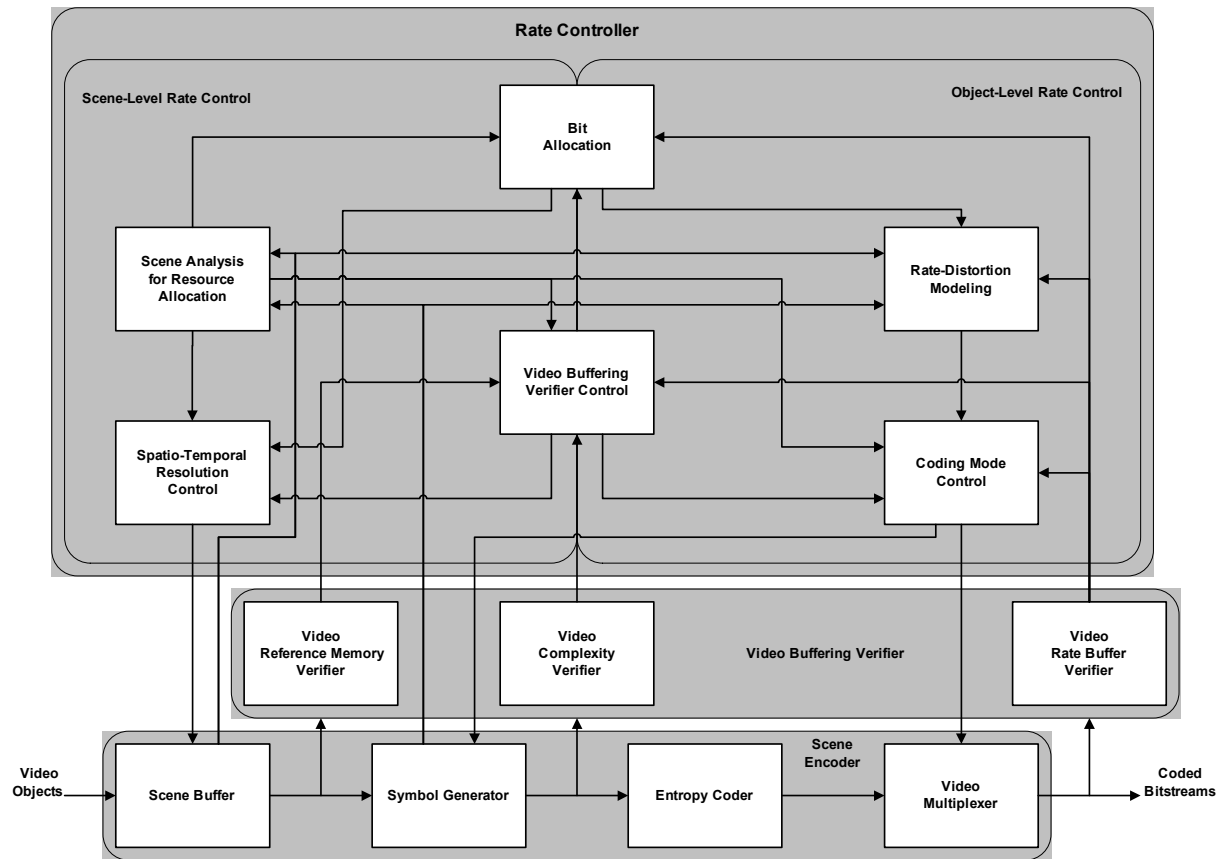


Figure 6.5 – Architecture of the proposed rate control algorithm

6.4.1 Application Scenario Requirements

Before presenting the main modules of the proposed rate control algorithm, it is important to define the relevant application scenario requirements, since they will strongly conditioned the solutions to be adopted.

LOW-DELAY ENCODING

As referred in Chapter 5, the more relevant delay constraints in terms of rate control are the processing and the buffering delay, since these are the ones directly related to the encoding process.

The rate control algorithm proposed in this Thesis aims at controlling scene encoders operating under low-delay constraints, i.e., under low processing delay and low buffering delay. In terms of processing delay, it means that the rate controller can only process the VOPs corresponding to each encoding time instant, i.e., no future VOPs are analyzed, and a single pass encoding is performed, i.e., each coding unit is encoded only once. In terms of buffering delay, it means that the rate control algorithm should be able to cope with small buffer sizes for which the number of bits per picture needs to be kept nearly constant and, consequently, a very tight control of the VBV buffer occupancy and fine coding mode control (e.g., quantization parameter selection) is required.

CONSTANT BIT RATE ENCODING

In the context of this Thesis, it is assumed that the rate controller uses a combined VBV

control with shared bandwidth resources (see Section 4.7), i.e., the scene encoding is constrained by a single VBV buffer where the corresponding buffer at the encoder side is filled at variable bit rate depending on the encoder bit production and is drained at constant bit rate (CBR). In this case, the drain rate of the encoder rate buffer, $R(t)$, is simply described by

$$R(t) = R, \quad t \geq 0. \quad (6.48)$$

Notice, that although this rate control algorithm does not target explicitly other types of channels, it can be straightforwardly extended to deal with piecewise constant bit rate⁸ (PBR) and variable bit rate⁹ (VBR).

For PBR channels, the channel rate is time varying and the drain rate of the encoder rate buffer is described by

$$R(t) = R_i, \quad t \in [t_i, t_{i+1}[\quad (6.49)$$

where $[t_i, t_{i+1}[$ is the interval between consecutive channel changing rates or consecutive VOP encoding time instants.

For VBR channels, it is difficult to define the drain rate of the encoder rate buffer by a simple function. Notice, however, that from a rate control point of view what is important is to track the amount of bits drained from the encoder rate buffer (or, equivalently loaded into the VBV buffer).

For each case, the amount of bits, d_i , drained from the encoder rate buffer or equivalently loaded into the VBV buffer between time instants t_i and t_{i+1} can be described by

$$d_i = \begin{cases} R \cdot (t_{i+1} - t_i) & \Leftarrow CBR \\ R_i \cdot (t_{i+1} - t_i) & \Leftarrow PBR \\ \int_{t_i}^{t_{i+1}} R(\tau) d\tau & \Leftarrow VBR \end{cases} \quad (6.50)$$

Notice that, in terms of decoder operation, it is assumed that the decoder has a single VBV buffer for all elementary streams that is filled at constant bit rate¹⁰ and that each VOP is instantaneously removed from this common buffer at the corresponding VOP decoding time. The VBV buffer occupancy for CBR encoding is given by the following recurrence

$$\begin{aligned} B_0 &= B_{\max} \\ B_{i+1} &= B_i + R \cdot (t_{i+1} - t_i) - b_i \end{aligned} \quad (6.51)$$

where B_{\max} is the initial VBV buffer occupancy before the removal of the first VOP, B_i is the VBV buffer occupancy before the removal of VOP i from the buffer, R is the channel bit

⁸ In this scenario, it is assumed that the channel bit rate is constant over short periods of time (several VOP intervals).

⁹ The variable bit rate channel can be seen as a piecewise constant bit rate channel where the channel varies between each two VOP encoding time instants. The variation is constrained by some type of leaky-bucket technique.

¹⁰ In the case of PBR or VBR encoding, this buffer is filled at piecewise constant and variable bit rate, respectively.

rate, $(t_{i+1} - t_i)$ is the time elapsed since the last VOP removal, and b_i is the number of bits spent for VOP i .

To prevent VBV buffer underflow and overflow, the VBV buffer occupancy must always verify the following conditions

$$b_i \leq B_i \leq B_{\max} \quad (6.52)$$

where B_{\max} is the VBV buffer size, i.e., the maximum VBV buffer occupancy.

An immediate consequence of (6.51) is that, in order to prevent possible future VBV buffer underflows, the decoder must delay the initial picture presentation until the VBV buffer reaches its maximum occupancy. Figure 6.6 illustrates the situation where, due to an early decoding start, the given VOP bits are not yet in the VBV buffer for some decoding time instants, leading, therefore, to a VBV underflow.

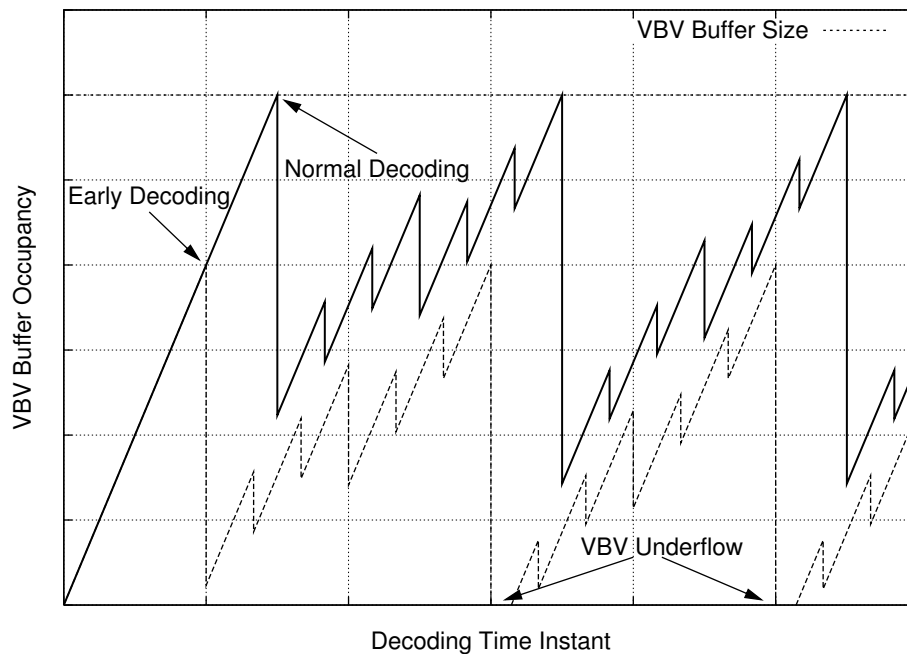


Figure 6.6 – Early decoding start leading to VBV underflows

In a VBR scenario, the channel is switched-off whenever there is no information to transmit; thus the VBV buffer never overflows (i.e., the corresponding buffer at the encoder never underflows) and, consequently, VBV overflows can be easily prevented.

The VBV buffer occupancy model for VBR encoding is given by the following recurrence

$$\begin{aligned} B_0 &= B_{\max} \\ B_{i+1} &= \min \left[B_{\max}, \max \left[0, B_i + R_{\max} \cdot (t_{i+1} - t_i) - b_i \right] \right] \end{aligned} \quad (6.53)$$

Notice that, in order to fully characterize a VBR scenario, additional parameters are needed, besides the maximum channel bit rate, R_{\max} , notably, the average channel bit rate, R , and a temporal window to compute the average bit rate [88].

In VBR conditions, the main constraint is the VBV underflow, which occurs whenever a given complete picture is not yet in the VBV buffer at the corresponding removal time instant.

To prevent the VBV buffer from underflowing, the encoder must guarantee always that the VBV buffer meets the following condition

$$B_i \geq b_i \quad (6.54)$$

For the VBR case, the VBV buffer overflow constraint does not apply because this would correspond to an encoder rate buffer underflow that does not occur since whenever the encoder rate buffer becomes empty the corresponding channel rate is zero. An important feature of VBR encoding is that the bitstream stuffing is not needed and can be disabled.

In terms of VBV control, PBR channels can be treated either as CBR, and in this case VBV buffer underflows and overflows must be avoided, or as VBR where only VBV buffer underflows need attention.

Notice that, while in CBR scenarios bit rate variations are experienced on a frame-by-frame basis (i.e., short-term bit rate variations), in VBR scenarios, besides frame-by-frame bit rate variations, there are also long-term variations at the scale of minutes. Such fluctuations complicate the admission and scheduling of video transmission under QoS transmission scenarios. These problems, however, are not dealt with in the context of this Thesis.

6.4.2 Scene Analysis for Resource Allocation

A mandatory feature of the proposed rate control algorithm is to be able to extract relevant information for the control process from the input video data before really encoding it. Since the addressed scenario requires low-delay encoding, only the input data for the corresponding encoding time instant is analyzed. Therefore, the scene analysis for resource allocation is carried out prior to each encoding step, i.e., before encoding any VOP for a given time instant, all VOPs of all VOs to be encoded at the time instant under consideration are analyzed.

This scene analysis module follows a basic principle for each time instant: all analysis is performed before all coding for that time instant. In this way, the actual data reflecting the instantaneous characteristics of each VO can be used, for example, to decide the best spatio-temporal resolution trade-off and efficiently distribute the available resources before really encoding the VOPs. This is especially useful when the scene or a particular VO changes its characteristics rapidly and thus the allocation and distribution of resources should quickly reflect these changes. This is not so well handled when the statistics of the previous time instant are used as in the case of the VM5 and VM8 rate control algorithms implemented in the MPEG-4 Visual reference software [32]. To properly tackle this problem, a new analysis architecture is proposed and has been implemented, allowing to efficiently perform the necessary scene analysis functions [27].

This scene analysis module receives as input, from the scene buffer, the set of original VOPs to be encoded, for each particular time instant, and, from the symbol generator, the corresponding set of previously reconstructed VOPs which are stored in the prediction memory. After performing motion estimation, shape coding¹¹, and counting the number of MBs of each type (i.e., transparent, opaque and boundary) the following information is extracted for each VO:

- Size, $S_{VO}[n]$ – the number of non-transparent MBs in the current VOP, reflecting the

¹¹ The proposed rate control algorithm assumes the shape is lossless encoded. Although the MPEG-4 Visual standard [29] supports lossy shape coding, the artifacts introduced are usually subjectively very annoying.

number of textured MBs to be coded in the current encoding time instant.

$$S_{VO}[n] = \#\{\text{non-transparent MBs in VOP } n\} \quad (6.55)$$

- Object Activity, $A_{VO}[n]$ – the sum of the absolute values of the current VOP motion vectors, reflecting the VO activity for the current encoding time instant.

$$A_{VO}[n] = \sum_{k=1}^{N_{MB}[n]} \sum_{l=1}^4 |mv_x[k][l]| + |mv_y[k][l]| \quad (6.56)$$

- Texture Complexity, $C_{VO}[n]$ – proportional to the variance of the prediction error for the current VOP, reflecting the VO texture coding complexity.

$$C_{VO}[n] = \sum_{k=1}^{N_{MB}[n]} C_{MB}[k] \quad (6.57)$$

where

$$C_{MB}[k] = \begin{cases} \sum_{l=1}^{64} p_Y^2[l] \Leftarrow \text{Inter Coded MB} \\ \sum_{l=1}^{64} (p_Y[l] - \bar{p}_Y)^2 \Leftarrow \text{Intra Coded MB} \end{cases} \quad (6.58)$$

$p_Y[l]$ is, respectively, for Inter and Intra coded MBs, the luminance¹² motion compensated prediction error and the luminance value of pixel l , while \bar{p}_Y is its corresponding average value, for MB k .

The relevant MB data, i.e., MB prediction error, motion vectors and shape modes, and the information given by (6.55), (6.56), and (6.57), are then feedforwarded to the other rate controller modules in order to guide their actions, notably, the spatio-temporal resolution control, the bit allocation, and the coding mode control modules (these modules are described in the following sections).

6.4.3 Spatio-Temporal Resolution Control

This module is conceptually responsible for deciding the appropriate spatial¹³ and temporal resolutions of the input video data, trying to achieve the best trade-off between spatial quality and motion smoothness.

In the proposed rate control algorithm, the spatial resolution of the input data is not changed during the scene encoding. Regarding the temporal resolution, the scene encoder assumes a base scene temporal resolution – scene rate (SR) – equal to the minimum common divider of all VO temporal resolutions in the scene. For example, for a scene with three VOs, encoded at 15, 10, and 7.5 VOPs/s, the scene rate is 30Hz, which means that at each 33,3 ms the rate controller has to inspect the scene buffer to check if there are any VOPs to encode and

¹² Since, generally, the Human Visual System (HVS) is more sensitive to the luminance component and most of the texture detail resides in this component, the chrominance is not used for these calculations.

¹³ Changing the spatial resolution during the encoding process is not allowed in the MPEG-4 video profiles used in this Thesis (i.e., Simple, Core, and Main Profiles – see Section 2.6.4). Therefore, this possibility was not considered in the work presented here.

perform the scene analysis to extract the relevant information.

Therefore, for each possible encoding time instant, this module receives input from the bit allocation module regarding the available number of bits to encode the next SP (see definition in Section 6.4.5), $T_{sp}[p]$; receives input from the video buffering verifier control regarding the estimated status of the different video buffering verifier buffers after the current set of VOPs had been encoded, i.e., $E[VMV[p]]$, $E[VCV[p]]$, $E[B-VCV[p]]$, and $E[VBV[p]]$; and, finally, can also receive input from the scene analysis module regarding the motion smoothness of each VO, although this possibility has not been addressed in this work.

Based of this information, the spatio-temporal resolution control module decides whether to skip or to proceed with the encoding process for the current time instant. This decision is primarily based on the estimated status of the different video buffering verifier models. If it is foreseen that none of these models will be violated, this verification is successful; otherwise, the current encoding time instant is skipped (see Section 6.4.8).

If the above verification is successful, the spatio-temporal resolution control module investigates if the allocated number of bits coming from the bit allocation module is enough to encode the estimated auxiliary data (i.e., header¹⁴, motion, and shape data) for all VOPs to be encoded. If the allocated number of bits for the current encoding time instant is enough to encode this data, then this verification is considered successful; otherwise, the current encoding time instant is skipped (see Section 6.4.8).

The spatio-temporal resolution control module may also decide to skip some VOPs¹⁵ of the current encoding time instant based not on the scarceness of bits to encode the input VOs, but on the low object activity and low texture complexity signaled by the scene analysis module, this way saving bits for the more demanding VOs¹⁶. A recent work on this topic considered explicitly skipping VOPs in rate-distortion optimizations, which in this Thesis is not considered, in order to obtain a trade-off between spatial and temporal qualities [109].

The main novelty of the spatio-temporal resolution control module proposed in this Thesis is the control of the temporal resolution of the different VOs in the scene based on the status of the video buffering verifier mechanism buffers, which is typically not considered in the literature for the VMV and VCV models. Of course, this module could be improved, considering also the possibilities of dynamically changing the spatial resolution of the different VOs (only for some video profiles) and introducing motion smoothness criteria.

6.4.4 Rate-Distortion Modeling

In order to be able to predict in advance the behavior of the scene encoder, notably the joint behavior of the symbol generator and the entropy coder (see Figure 6.5), it is necessary to have some mathematical description of this behavior that can be adapted to the actual encoding results during the encoding process, typically through parameter estimation. This configures a model-reference adaptive control system, as described in Section 6.2.3, since the

¹⁴ The header bits of the previous encoding time (of the same VOP coding type).

¹⁵ In the proposed rate control algorithm, skipping is currently performed for all VOPs of a given encoding time instant. When only some VOPs are skipped during the encoding of segmented scenes, the decoded composed scene may exhibit holes due to the temporal changes in the shape of the different VOs. In these circumstances, the compositor should handle this problem to avoid these subjectively annoying artifacts.

¹⁶ This feature is not currently implemented in the proposed rate control algorithm.

rate controller has the task to minimize the error between the output of the model and the actual output of the scene encoder.

In terms of the rate controller operation, each VOP encoder, i.e., the symbol generator and entropy coder of each VO in the scene, can be seen as the process to be controlled. For this process, the input control signal is the target number of bits for each encoding time instant, $T_{VOP}[i]$ ¹⁷, while for the system output it is convenient to consider three different types of output: 1) the number of bits used to encode the VOP, $S_{VOP}[i]$; 2) the VOP average pixel distortion, $D_{VOP}[i]$; and 3) the encoder rate buffer occupancy, $B[i]$ (see Figure 6.7). These three different types of output are closely related to the rate control goals for CBR encoding, i.e.,

- Achieve a pre-defined average number of bits per second – in this case, the rate controller should compensate deviations between the encoder output $S_{VOP}[i]$ and the command signal $T_{VOP}[i]$.
- Maximize the spatial quality and minimize quality fluctuations – in this case, the rate controller should take into account the encoder output $D_{VOP}[i]$ along time and between VOPs to compensate, for example, future bit allocations.
- Keep the VBV buffer occupancy within permitted bounds – in this case, the rate controller should keep track of the VBV buffer occupancy and correct the deviations.

Notice that, in terms of the dynamics of the system, the main issue is to be able to adequately compensate the deviations relatively to the target values (i.e., target output bits, target quality, or target buffer occupancy).

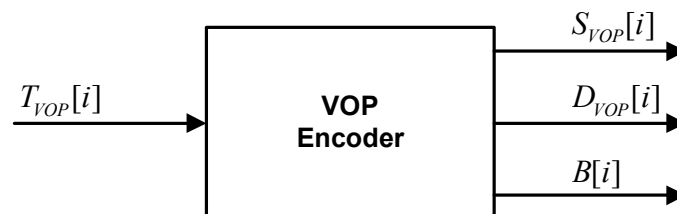


Figure 6.7 – VOP-level encoder model

Therefore, for rate control purposes, this Thesis proposes that the encoder be modeled by rate-distortion functions at the different levels of rate control operation, i.e., VOP-level, MB-level, and, although not considered in this Thesis, DCT-level. With these functions, the rate controller is able to feedforward predict the behavior of the scene encoder with relative accuracy aiming at reducing the amount of compensation needed to bring the scene encoder to the ideal behavior.

Rate-distortion models are used essentially to map bit allocations into coding parameters, notably, the quantization parameter that will be used to encode each coding unit, i.e., a given VOP or MB inside a VOP.

¹⁷ For the cases where the main goal has to achieve a given target quality instead of a given target output average bit rate, the command signal can be exchanged with $D_{VOP}[i]$.

I) VOP-level Encoder Rate-Distortion Modeling

At the VOP-level, the VOP encoder is modeled by its VOP rate and distortion functions, $R(Q)$, $D(Q)$, and $R(D)$, for each VO, and for each VOP coding type (see Chapter 5).

In the proposed rate control algorithm, the following $R(Q)$ model, derived from the results of Chapter 5, is used for the Intra and Inter VOP coding types in order to estimate the initial quantization parameter at the beginning of each VOP encoding

$$R(Q) = \exp(-aQ^c + b) \quad (6.59)$$

where a , b , and c are model parameters¹⁸, and Q is the VOP quantization parameter.

Since (6.59) is a nonlinear function, at the end of each VOP encoding, the model parameters are estimated through the Levenberg-Marquardt nonlinear least squares estimation algorithm [175] by minimizing

$$\chi_{RQ}^2 = \sum_{k=i-W_{RQ}+1}^i \left(\frac{S_{VOP}^{\text{text}}[k]}{N_{PIX}[k]} - \exp(-a(\bar{Q}_{VOP}[k])^c + b) \right)^2 \quad (6.60)$$

where $S_{VOP}^{\text{text}}[k]$ is the number of bits used to encode the VOP texture, $N_{PIX}[k]$ is the number of non-transparent pixels, $\bar{Q}_{VOP}[k]$ is the average quantization parameter over all encoded MBs in the VOP¹⁹, and $W_{RQ} = 10$ is the sliding window size over which the minimization is carried on.

The stop criterion on the relative change of the fitting error between two iterations is set to $\varepsilon_{\text{FIT}} = 10^{-3}$ and, additionally, in order to prevent the iterative process to run infinitely, the maximum number of iterations is limited to $N_{\text{ITER}} = 10$.

For Inter-coded VOPs, the following delta rate-quantization model is used

$$\Delta R(Q) = \left(a \cdot \frac{1}{Q^2} + b \cdot \frac{1}{Q} + c \right) \cdot \Delta Q \quad (6.61)$$

where a , b , and c are model parameters²⁰, Q is the VOP quantization parameter, and ΔQ is the difference between the reference VOP average quantization parameter, \bar{Q}_{ref} , and the weighted average quantization parameter of the last three encoded VOPs of the same coding type, Q_0 , i.e.,

$$\Delta Q = \bar{Q}_{\text{ref}} - Q_0 \quad (6.62)$$

where

$$Q_0 = \frac{2Q[i-1] + Q[i-2] + Q[i-3]}{4} \quad (6.63)$$

can be seen as an estimation of the stationary quantization parameter (see Section 5.3.2).

¹⁸ Notice there is a different set of model parameters for each VOP coding type and for each VO in the scene.

¹⁹ Skipped and transparent MBs are not counted.

²⁰ Notice there is a different set of model parameters for each VO in the scene.

The delta rate-quantization parameters are estimated at the end of each VOP encoding through the linear least squares estimation algorithm by minimizing

$$\chi_{\Delta RQ}^2 = \sum_{k=i-W_{RQ}+1}^i \left(\frac{\Delta R[k]}{\Delta Q[k]} - \left(a \left(\frac{1}{\bar{Q}_{VOP}[k]} \right)^2 + b \frac{1}{\bar{Q}_{VOP}[k]} + c \right) \right)^2 \quad (6.64)$$

where

$$\Delta R[k] = \frac{S_{VOP}^{\text{text}}[k]}{N_{PIX}[k]} - \frac{S_{VOP}^{\text{text}}[k-1]}{N_{PIX}[k-1]} \quad (6.65)$$

is the difference between the number of texture bits per pixel generated by VOP k and its reference of the same coding type²¹; and

$$\Delta Q[k] = \bar{Q}_{VOP}[k] - Q_0[k] \quad (6.66)$$

Notice that in (6.64) only the encoding time instants k for which $\Delta Q[k] \neq 0$ can be considered. In fact, to avoid numerical problems, only the points for which $|\Delta Q[k]| \geq 1$ are considered.

II) MB-level Encoder Rate-Distortion Modeling

The purpose of this level of modeling is to provide a fine level of rate control, i.e., with a lower sampling period – MB period – and, consequently, provide the rate controller with a faster reaction to deviations relatively to the nominal operation.

After assigning a certain target number of bits to encode a given VOP, the rate controller has to guarantee that this target is met. A straightforward method is to divide uniformly the VOP target number of bits by the number of MBs in the VOP; however, this will lead, typically, to large quality fluctuations inside the VOP. Therefore, it is convenient to assign a number of bits for each MB that will minimize the quality fluctuations inside the VOP. For this purpose, it will be assumed that each MB can be modeled by the following rate and distortion functions:

$$R_{MB}(Q) = a_1 \frac{1}{Q^{c_1}} X_{MB} \quad (6.67)$$

where a_1 and c_1 are the model parameters estimated during encoding and X_{MB} is a MB texture complexity measure²².

$$D_{MB}(Q) = \min \left[a_2 Q^{c_2}, \sigma_{MB}^2 \right] \quad (6.68)$$

where a_2 and c_2 are the model parameters estimated during encoding and σ_{MB}^2 is the MB prediction error variance.

$$R_{MB}(D) = a_3 \frac{1}{D^{c_3}} \quad (6.69)$$

²¹ For P-VOPs, immediately after I-VOPs, the delta RQ model is not applied.

²² In the context of this Thesis, the MB MAD has been used.

where a_3 , and c_3 are the model parameters estimated during encoding.

In terms of the proposed rate control algorithm, only (6.67) is used to estimate the quantization parameter of each MB in a given VOP. Notice, however, that in the MPEG-4 Visual standard [29], the quantization parameter has a limited range of variation between consecutive MBs, i.e., $|\Delta Q_{MB}[i]| \leq 2$ (with $\Delta Q_{MB}[i] = Q_{MB}[i] - Q_{MB}[i-1]$). Therefore, it is useless to compute a very accurate MB quantization parameter for each MB since, given the quantization parameter of the previous MB, only a reduced number of values can be used for the current MB (at most five different values can be used).

Although from (6.67) it is possible to obtain any MB quantization parameter value, $Q_{MB}[i]$, for a given target number of bits, $T_{MB}[i]$, this value needs to be clipped. Consequently, the MB $R(Q)$ function is used mainly to compute the quantization parameter variation between consecutive MBs, $\Delta Q_{MB}[i]$, inside each VOP; therefore, it is not required to be very accurate.

The MB-level rate-distortion function aims, thus, at modulating the VOP-level quantization parameter in order to provide a fine adjustment of the VOP encoded bits. Therefore, more than a precise rate-quantization function, it is important to specify a stable modeling of the MB rate-quantization characteristics. For this purpose, parameter c_1 in (6.67) has been set constant, i.e., $c_1 = 1$ resulting in the rate-quantization function

$$R_{MB}(Q) = a \frac{1}{Q_{MB}} X_{MB} \quad (6.70)$$

In order to adapt this model to the results of the encoding process, after each MB encoding, the parameter a is estimated through the following steps:

1. Compute an estimate of a , $\hat{a}[i]$, using data from the current VOP up to MB i , which minimizes the following quadratic error

$$\chi_{MB}^2 = \sum_{k=1}^i \left(b_{MB}^{\text{text}}[k] - a \frac{X_{MB}[k]}{Q_{MB}[k]} \right)^2 \quad (6.71)$$

that is

$$\hat{a}[i] = \frac{\sum_{k=1}^i \left(b_{MB}^{\text{text}}[k] \frac{X_{MB}[k]}{Q_{MB}[k]} \right)}{\sum_{k=1}^i \left(\frac{X_{MB}[k]}{Q_{MB}[k]} \right)^2} \quad (6.72)$$

2. Compute the new estimate of a as a weighted average of (6.72) and the rate-quantization model parameter of the last encoded VOP of the same type, a^{prev} , i.e.,

$$a[i] = \hat{a}[i] \times \beta + a^{\text{prev}} \times (1 - \beta) \quad (6.73)$$

where

$$\beta = \left(\frac{i}{N_{MB}} \right)^2 \quad (6.74)$$

The main reason for using a quadratic instead of a linear dependency on the MB index i for

β , as in (6.47), is to decrease the dependency of (6.73) on the current VOP data, i.e., on (6.72), while there are only a few encoded MBs. This way, a higher weight can be given to a^{prev} , for the first MBs of a given VOP, which has been computed based on all coded data of the previous VOP, being therefore a more stable value. Consequently, a more stable estimation of parameter a is achieved.

For the first VOP of each coding type, a is initialized as

$$a^{\text{init}} = Q_{\text{VOP}} \frac{T_{\text{VOP}}}{\sum_{k=1}^{N_{\text{MB}}} X_{\text{MB}}[k]} \quad (6.75)$$

where Q_{VOP} is the target quantization parameter for the corresponding VOP.

Figure 6.8 shows the value of parameter a computed through (6.73) at the end of each VOP, for the *Kayak* sequence, in QCIF format, encoded at 15Hz, with an Intra period of 1 second and a target bit rate of 160 kbit/s. As can be seen from Figure 6.8, after a few VOPs have been encoded, the MB rate-quantization parameter reaches a relatively stable value around $a \approx 175$.

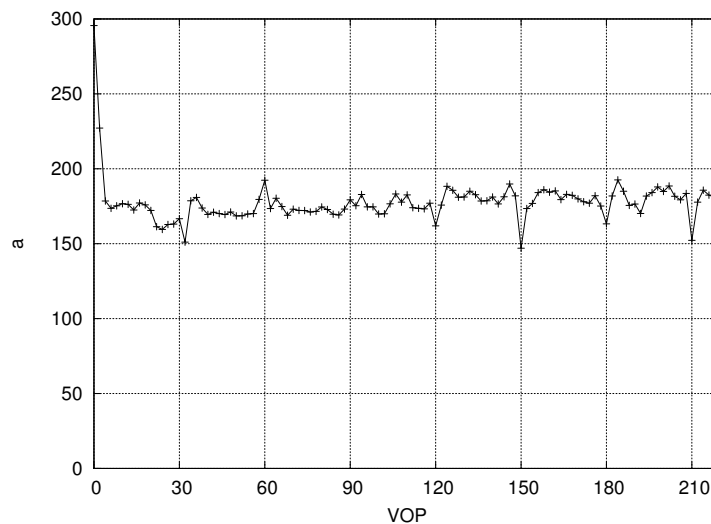


Figure 6.8 – Evolution of parameter a for the *Kayak* sequence [QCIF@15Hz; 160 kbit/s]

Figure 6.9 illustrates the evolution of parameter a inside a given VOP for two VOPs of the *Kayak* sequence, where

$$a_{\text{MB}}[i] = \frac{b_{\text{MB}}^{\text{text}}[i] Q_{\text{MB}}[i]}{X_{\text{MB}}[i]} \quad (6.76)$$

As can be seen in Figure 6.9a and Figure 6.9b, a_{MB} tends to have large fluctuations between consecutive MBs inside a VOP, as the estimate of a , \hat{a} , based only on data from the current VOP as in (6.72). However, using the weighted function (6.73) produces a stable estimation of a .

Notice that, in both cases (Figure 6.9a and Figure 6.9b), a converges smoothly to its final value at the end of the VOP.

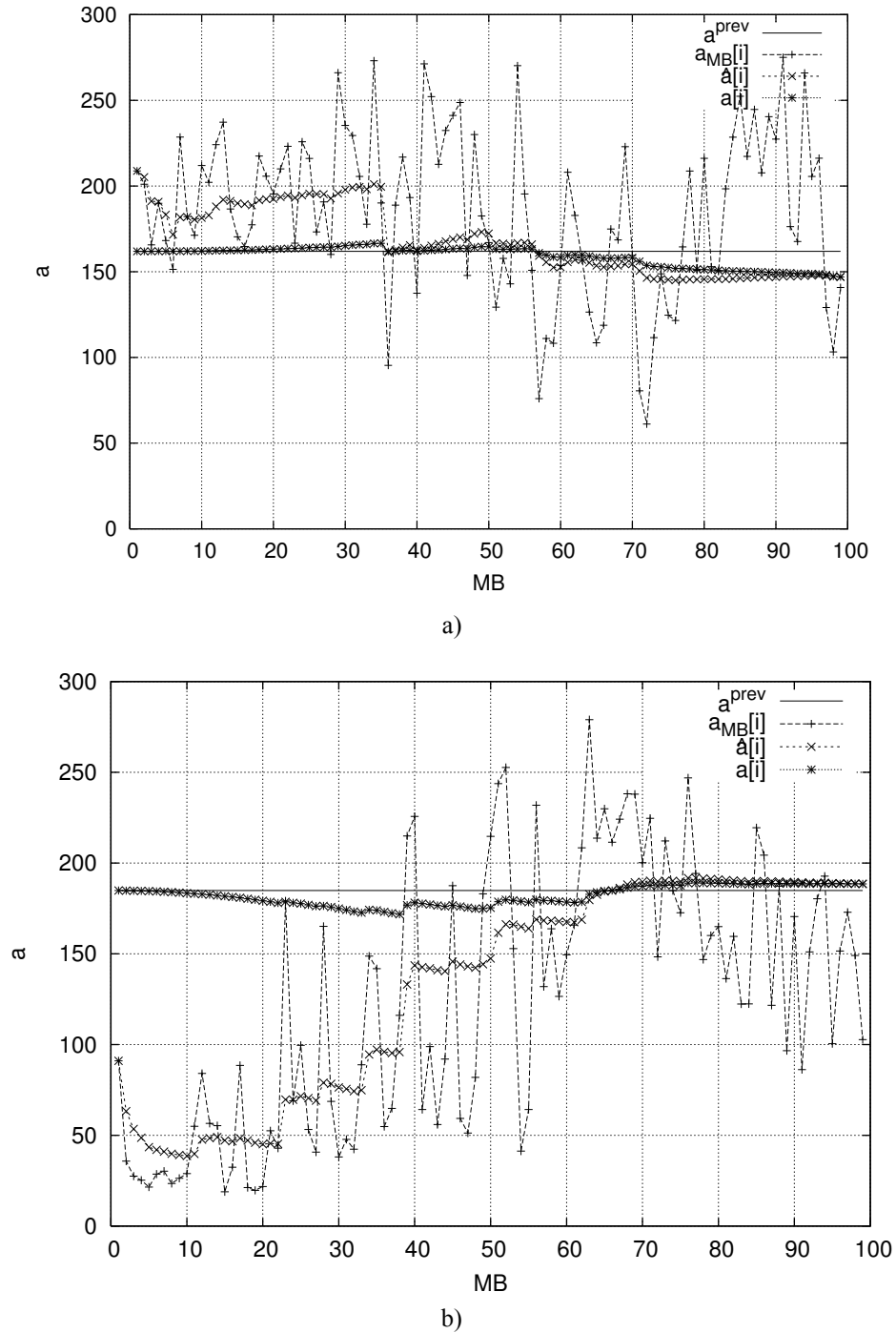


Figure 6.9 – Estimation of parameter a for two VOPs of the Kayak sequence [QCIF@15Hz; 160 kbit/s]: a) VOP 60 Intra-coded; b) VOP 100 Inter-coded

The main novelty of this rate-distortion modeling module is the used of two modeling levels (VOP-level and MB-level) for obtaining a good trade-off between accurate bit allocation and spatio-temporal quality smoothness. The VOP-level rate-distortion models are used to obtain a first approximation of the average VOP quantization parameter, while the MB-level rate-distortion models are used for fine adjustment of the MB quantization parameter in order to allow a fine control of the VBV constraints and simultaneously avoid excessive MB quantization parameter variations that would penalize the spatial (and possibly also the temporal) quality smoothness.

6.4.5 Bit Allocation

As referred previously, the purpose of the rate control algorithm proposed in this Thesis is twofold: first, to control the output of a MVO encoder in a way that the conformance mechanisms are not violated, and, second, to maximize the subjective quality of the decoded video. To achieve the second goal, it is necessary to maintain the quality of the decoded VOPs approximately constant, or at least, smoothly varying, both along time and among the several VOs composing the video scene for each encoding time instant.

To achieve these goals, it is necessary to properly allocate the available bit rate resources. This Thesis proposes that this task to be handled in the bit allocation module (see Figure 6.5). However, in a MVO encoding scenario, the bit allocation task can be rather complex, since the different VOs composing a scene may have different characteristics in terms of shape, motion, and texture complexity, and thus may be encoded with different temporal resolutions. Consequently, following a strategy of dividing to conquer, this Thesis proposes that the bit allocation be partitioned into several hierarchical levels, similarly to the syntactic organization of the encoded video data.

In the context of this Thesis, it is assumed that the different VOs composing a scene can be encoded at different frame rates; however, in order to allow easy random access in a MVO encoding scenario, it is imposed here that all the VOs composing a scene are encoded with GOVs of identical duration, i.e., the time distance between two consecutive I-VOPs is equal for all VOs of a given scene and coincident in time, although the number of VOPs in each GOV may be different²³. Therefore, this Thesis proposes that the bit allocation will be performed using the following hierarchical levels:

- **Group of Scene Planes (GOS)** – The set of all encoding time instants between two random access points, typically encoded with a constant number of bits (in CBR scenarios); GOSs may be composed by VOs with different VOP rates. In the case of a single VO, a GOS becomes a Group of Video Object Planes (GOV). The bit rate control algorithm ensures that the available bit rate for encoding the GOS is adequately allocated among the VOPs in the GOS (for all VOs).
- **Scene Plane (SP)** – The set of all VOPs of all VOs to be encoded at a particular encoding time instant. The bit rate control algorithm ensures that the allocated bits for each encoding time instant are adequately distributed among the several VOPs to be encoded for that particular time instant. Notice that not all VOs of a given scene have VOPs to be encoded in every scene plane (see Figure 6.10).
- **Video Object Plane** – The sample of each video object at a particular encoding time instant. The bit rate control algorithm computes the adequate coding parameters for the VOP to meet the target number of bits to encode the given VOP.
- **Macroblock** – The smallest coding unit for which the quantization parameter can be changed. The bit rate control algorithm selects a quantization parameter for each MB based on the MB-level rate-distortion function for a fine allocation of bits inside each VOP, taking into account the complexity of the several MBs to encode.
- **Block** – Subdivision of the MB with 8×8 samples; each MB is composed by four luminance blocks and two chrominance blocks for 4:2:0 content. The quantization

²³ A restriction is imposed that the VOP rates have a common time base.

parameter selection must take into account the statistical properties of each block in a MB in order to select the most adequate quantization parameter for that MB.

- **DCT Coefficient** – The smaller texture coding unit. The DCT coefficients are encoded for each Block and represented as Run-Level codewords, relying on the assumption that each DCT block is composed by long runs of zero valued coefficients. An efficient bit rate control algorithm should optimally select the DCT coefficients to be transmitted. The rate control algorithm may select the coefficients with more energy, although in a rate-distortion sense can be advantageous to not transmit a higher energy coefficient in order to not break a long sequence of zero value coefficients. The proposed rate control algorithm only implements this level to finely adjust the number of bits/VOP to avoid encoder rate buffer overflow.

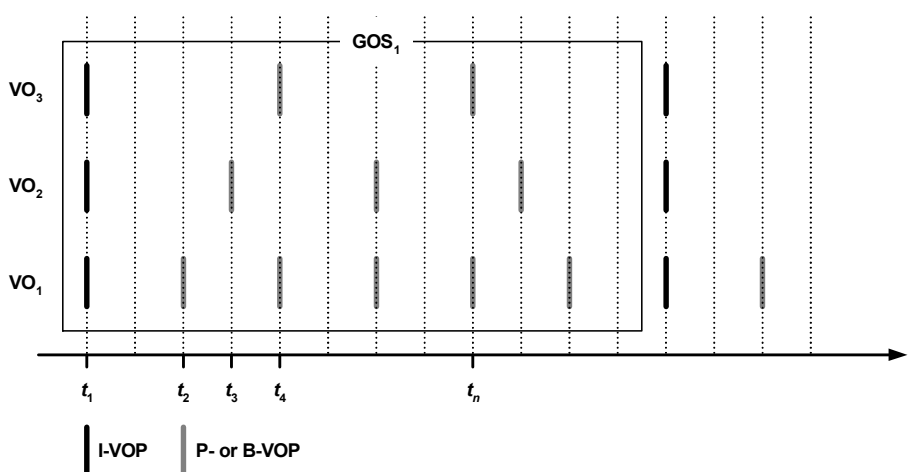


Figure 6.10 – Multiple video objects encoding with different VOP rates

For each encoding time instant, for the various VOs to encode, the bit allocation module at the GOS-, SP-, and VOP-level, should take into the account the following set of fixed (scene encoder configuration) and variable (operational conditions) parameters:

- Fixed parameters (scene encoder configuration)
 - Channel bit rate.
 - Each VO target encoded temporal resolution.
 - VBV buffer size.
- Time-varying parameters (operational conditions)
 - Current VBV buffer occupancy.
 - Target VBV buffer occupancy.
 - Number of bits for the previous VOPs of each VO

Additionally, the bit allocation module should distribute the available resources (bits) among the several VOPs in the SP, and, inside each VOP, among the several MBs, in a way the overall quality (perceptual) is maximized.

Notice that deviations between the intended and the actual encoding results should be compensated through adequate feedback compensation mechanisms. Since multiple arbitrarily shaped video objects rate control is, in principle, much more complex than typical frame-

based rate control [17], it is convenient to separate the different sources of uncertainty. For each source of uncertainty, an adequate compensation mechanism should be provided, in a way that, together, the different parts contribute to the overall rate control goals. Therefore, for each bit allocation level, a different feedback compensation mechanism is proposed in this Thesis.

In this context, each feedback compensation mechanisms should be able to control deviations for the reference signals (i.e., target number of bits, buffer occupancy, and/or quality) at the same hierarchical levels defined above.

GOS-LEVEL BIT ALLOCATION

In the context of this Thesis, a given video scene composed of N_{VO} VOs is divided in N_{GOS} groups of scene planes where each GOS is composed of N_{VO} GOVs (one for each VO in the scene) of equal duration²⁴.

In terms of bit allocation at the GOS-level, the algorithm aims at allocating a nominal number of bits to each GOS, \bar{T}_{GOS} , that is proportional to the GOS duration, as follows

$$\bar{T}_{GOS}[m] = R[m] \times (t_{GOS}[m+1] - t_{GOS}[m]), \quad m = 1, \dots, N_{GOS} \quad (6.77)$$

where $R[m]$ and $t_{GOS}[m]$ are, respectively, the output average target bit rate and the starting time instant for GOS m .

Therefore, deviations from the expected results are compensated through an integral feedback compensation law as follows

$$T_{GOS}[m] = \bar{T}_{GOS}[m] + K_{GOS}[m] \cdot \sum_{k=1}^{m-1} (\bar{T}_{GOS}[k] - S_{GOS}[k]), \quad m = 1, \dots, N_{GOS} \quad (6.78)$$

where $S_{GOS}[k]$ is the number of bits used to encode the GOS k , and K_{GOS} is the integration factor given by

$$K_{GOS}[m] = \frac{1}{\min(\alpha_N, N_{GOS} - m + 1)} \quad (6.79)$$

where $\alpha_N = \max \left[1, \left\lceil 3 \frac{B_s}{\bar{T}_{GOS}[m]} \right\rceil \right]$ and B_s is the VBV buffer size.

The rationale for setting $K_{GOS} \leq 1$ is to avoid large quality fluctuations between adjacent GOSs, notably when scene changes occur inside a given GOS, and the bit allocation error compensation would penalize essentially the upcoming GOS, if $K_{GOS} = 1$. With this approach, GOS bit allocation deviations are smoothed through α_N GOSs, if the buffer size is sufficiently large to accommodate these bit production variations.

Notice, however, that for low bit rate encoding forcing a tight GOS bit allocation leads, typically, to larger quality variations, e.g., for $K_{GOS} = 2/3$ larger quality fluctuations are obtained relatively to $K_{GOS} = 1/3$.

²⁴ Since the different VOs in the scene may be encoded at different VOP rates, the number of VOPs in each GOV may be different.

SP-LEVEL BIT ALLOCATION

At the SP-level, in order to obtain approximately constant quality along a GOS, each SP should get a nominal target number of bits that is a fraction of the GOS target (6.78), proportional to the amount and complexity of the VOs to be encoded in that particular time instant. Therefore, in a MVO encoding scenario, this Thesis proposes that each VO in each SP be assigned a coding complexity weight, $\omega[n]$, reflecting its relative coding difficulty in the current SP, computed as follows

$$\omega[n] = \bar{S}_{VO}[n] \cdot \alpha_S + \bar{A}_{VO}[n] \cdot \alpha_A + \bar{C}_{VO}[n] \cdot \alpha_C \quad (6.80)$$

where $\bar{S}_{VO}[n]$ is the normalized size, $\bar{A}_{VO}[n]$ is the normalized activity, $\bar{C}_{VO}[n]$ is the normalized texture complexity of VO n in the current SP, and α_S , α_A , and α_C are weights such that $\alpha_S + \alpha_A + \alpha_C = 1$ (in this Thesis $\alpha_S = 0.2$, $\alpha_A = 0.5$, $\alpha_C = 0.3$, as described in [14]).

Representing by Γ the set of VOs to be encoded in current SP, the normalized VO size is given by

$$\bar{S}_{VO}[n] = \frac{S_{VO}[n]}{\sum_{k \in \Gamma} S_{VO}[k]} \quad (6.81)$$

where $S_{VO}[n]$ is given by (6.55).

Similarly, the normalized VO activity is given by

$$\bar{A}_{VO}[n] = \frac{A_{VO}[n]}{\sum_{k \in \Gamma} A_{VO}[k]} \quad (6.82)$$

where $A_{VO}[n]$ is given by (6.56).

Finally, the normalized VO texture complexity is given by

$$\bar{C}_{VO}[n] = \frac{C_{VO}[n]}{\sum_{k \in \Gamma} C_{VO}[k]} \quad (6.83)$$

where $C_{VO}[n]$ is given by (6.57).

The coding complexity of a given VOP n in SP p of GOS m is, given by²⁵

$$X_{VOP}[n] = \alpha_T[n] \cdot \omega[n], \quad n = 1, \dots, N_{VO} \quad (6.84)$$

where N_{VO} is the number of VOs in the scene and $\alpha_T[n]$ and $\omega[n]$ are, respectively, the coding type weight ($T \in \{I, P, B\}$) and coding complexity weight (6.80) – reflecting the texture, shape, and motion (if applicable) coding complexity – of VO n in SP i of GOS m . Notice, that $\alpha_T[n] = 0$, if VO n does not have a VOP in SP i of GOS m .

In SVO encoding scenarios, since there is only one VO to encode in each SP, $\omega = 1$; therefore, the VOP complexity depends exclusively on the VOP coding type weight, α_T .

In MVO encoding scenarios, the coding complexity of a given SP p of GOS m is defined as the sum of its VOP complexities defined according to (6.84), i.e.,

²⁵ The SP and GOS indexes have been dropped for simplicity.

$$X_{SP}[p] = \sum_{n=1}^{N_{VO}} X_{VOP}[n][p], \quad p = 1, \dots, N_{SP}[m] \quad (6.85)$$

Therefore, the GOS m coding complexity is the sum of its SP complexities defined according to (6.85), i.e.,

$$X_{GOS}[m] = \sum_{p=1}^{N_{SP}[m]} X_{SP}[p], \quad m = 1, \dots, N_{GOS} \quad (6.86)$$

where $N_{SP}[m]$ is the number of scene planes in the GOS m , with $N_{SP}[m] = \lceil t_{GOS}[m] \times SR \rceil$, where t_{GOS} is the GOS duration and SR is the scene rate.

The nominal target number of bits allocated for each SP in a given GOS is set by the following equation (the GOS index has been dropped for simplicity)

$$\bar{T}_{SP}[p] = T_{GOS} \frac{X_{SP}[p]}{X_{GOS}}, \quad p = 1, \dots, N_{SP} \quad (6.87)$$

Notice that, in the general case, the SP control scheme is a tracking control scheme since the target number of bits changes for each SP based on the number of VOPs in each SP. In fact, also the coding weights ω will be updated after each encoding time instant. Therefore, the actual SP target can be expressed by the following equation

$$T_{SP}[p] = T_{GOS} \frac{X_{SP}[p]}{X_{GOS}} + \frac{X_{SP}[p]}{\sum_{k=p}^{N_{SP}} X_{SP}[k]} \sum_{k=1}^{p-1} \left(T_{GOS} \frac{X_{SP}[k]}{X_{GOS}} - S_{SP}[k] \right), \quad p = 1, \dots, N_{SP} \quad (6.88)$$

Notice that, since the proposed rate control algorithm aims at low-delay encoding scenarios, both X_{SP} and X_{GOS} can only be computed with data from the current or past encoding time instants; therefore, these parameters must be adapted for each encoding time instant, i.e., for each SP, based on the actual data. In this context, it is also necessary to adapt the different VOP coding weights in order to obtain approximately constant distortion among consecutive encoding time instants for each VO.

When a VO is encoded using different VOP coding types, each VOP coding type should get a bit allocation that leads to approximately constant coding quality along time for that VO. Moreover, since the VOP complexity weight changes along time, it is important to estimate the relative coding complexity of each VOP coding type during the encoding process.

For a given VO to reach an average distortion per pixel for the different VOP coding types approximately constant, i.e., $D_I \approx D_P \approx D_B$, the following relation should approximately hold

$$\frac{b_I}{\alpha_I} \approx \frac{b_P}{\alpha_P} \approx \frac{b_B}{\alpha_B} \quad (6.89)$$

where, as referred above, α_I , α_P , and α_B , are the VOP coding type weights, and b_I , b_P , and b_B , are the average number of bits per pixel for each corresponding VOP coding type. At the beginning of the encoding process, the following values are used

$$\alpha_I = \min \left[\frac{B_S}{R_{GOV}/SR}, 2.7 \right], \quad \alpha_P = 1.0, \text{ and } \alpha_B = 0.5 \quad (6.90)$$

where B_s is the VBV buffer size, R_{GOV} is the average target bit rate for the current VO GOV, and SR is the scene rate, i.e., the number of scene planes encoded per time unit.

Therefore, at the beginning of each GOV, in order that the resulting distortion is kept approximately constant, the nominal target number of bits for each VOP coding type is set as

$$\begin{aligned} T_I &= \frac{\alpha_I}{\alpha_I + \alpha_P N_P + \alpha_B N_B} T_{GOV} \\ T_P &= \frac{\alpha_P}{\alpha_I + \alpha_P N_P + \alpha_B N_B} T_{GOV} \\ T_B &= \frac{\alpha_B}{\alpha_I + \alpha_P N_P + \alpha_B N_B} T_{GOV} \end{aligned} \quad (6.91)$$

where N_P , and N_B are, respectively, the number of P- and B-VOPs in the GOV, and T_{GOV} is the nominal target number of bits for the given VO GOV.

The relation expressed by (6.89) is highly dependent on the scene complexity, spatial and temporal resolutions and target bit rate, and, consequently, the VOP coding weights should be estimated from the actual encoded data. To achieve this goal along time, the coding type weights, α_I , α_P , and α_B , are estimated at the beginning of each GOS for each VO, i.e., at the beginning of each VO GOV, according to the recent coding results. Therefore, in this Thesis, it is proposed that before encoding each I-VOP, i.e., at the beginning of each GOV, of a given VO, α_I be estimated through the following equation

$$\alpha_I = \frac{\bar{b}_I}{\bar{b}_P} \left(\frac{\bar{D}_I}{\bar{D}_P} \right)^{\gamma_T} \quad (6.92)$$

where \bar{b}_I , \bar{D}_I , \bar{b}_P , and \bar{D}_P are, respectively, the average number of bits per pixel and the average pixel distortion for I- and P-VOPs, computed over window sizes W_I and W_P of past I- and P-VOPs encoding results, and γ_T is a parameter that controls the impact of the average distortion ratios on the estimation of the VO coding weight (in this Thesis, $W_I = 3$ and $W_P = N_{SP} - 1$, and $\gamma_T = 0.5$).

The rationale for (6.92) is the following: at the beginning of each GOS, α_I is estimated based on the ratio between the windowed average number of bits per pixel used to encode previous I- and P-VOPs with a correction factor $(\bar{D}_I/\bar{D}_P)^{\gamma_T}$ that intends to bring closer the average pixel distortion between I- and P-VOPs; in order to avoid an overcompensation that may lead to instabilities, the ratio \bar{D}_I/\bar{D}_P is raised to the power γ_T that controls the amount of compensation (for $\gamma_T = 0$, no distortion compensation is applied).

Figure 6.11 shows the evolution of α_I for two sequences: *Foreman* and *Stefan*. For comparison, the estimation of α_I with and without distortion correction is presented. As can be seen in Figure 6.11, the correction factor tends to produce a more stable α_I value. Moreover, in Figure 6.11a, when $\gamma_T = 0.5$ (i.e., with correction) the value of α_I decreases until GOS 7 and increases after that, relatively to the case with $\gamma_T = 0$ (i.e., without correction). This means that until GOS 7 too many bits were being allocated to I-VOPs

leading to I-VOPs being encoded with higher quality than P-VOPs, while after GOS 6 too few bits were being allocated to I-VOPs leading to I-VOPs being encoded with lower quality than P-VOPs. The correction factor compensates this behavior by reducing the relative allocation of bits between I- and P-VOPS before GOS 7, and increasing it after that. In Figure 6.11b, the phenomenon is similar; however, in this case, I-VOPs were being encoded systematically with higher quality than P-VOPs (without correction) and the correction factor compensates this behavior by reducing α_I .

To better illustrate this phenomenon, Figure 6.12 shows the PSNR of the luminance component for the first three GOS of the *Football* sequence with and without distortion correction, and in comparison with the MPEG-4 Visual VM8 [108]. As can be seen in Figure 6.12, the distortion correction allows achieving a more uniform quality along time, relatively to VM8 and to the case without distortion correction.

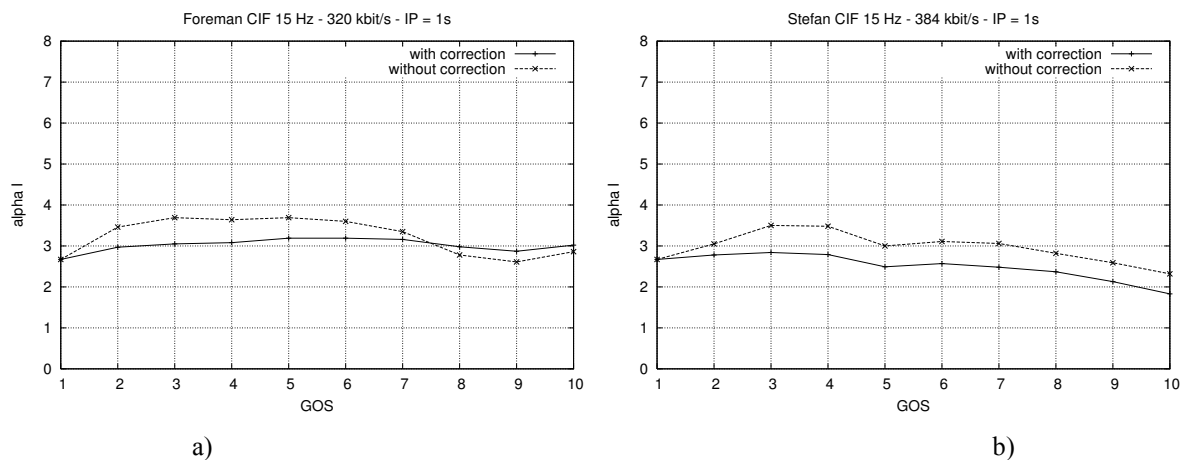


Figure 6.11 – Evolution of α_I along the sequence with and without distortion correction:
a) Foreman; b) Stefan

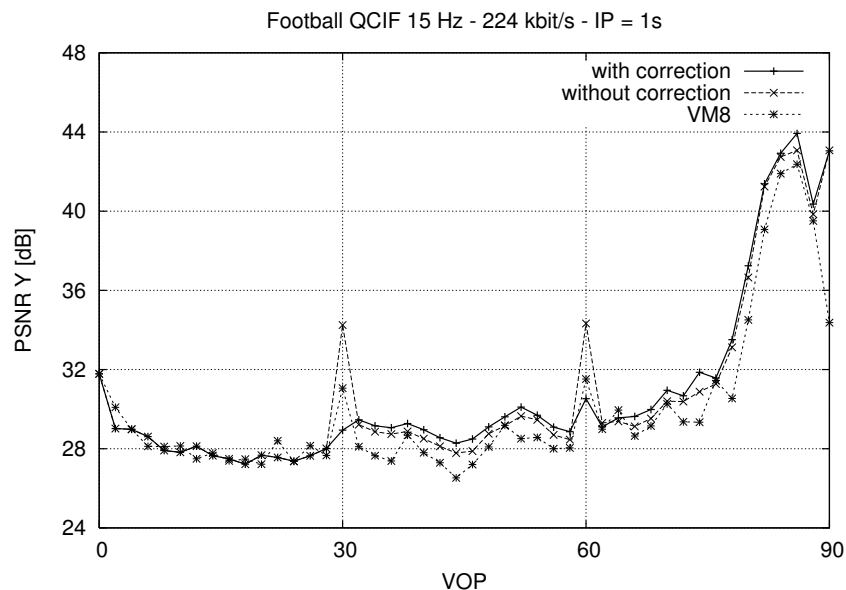


Figure 6.12 – PSNR for various VOP coding type weight adaptations for three GOS of the *Football* sequence

Although in this Thesis B-VOPs are not considered for the performance evaluation to be presented in Section 6.7 due to implementation limitations, the proposed solution is also applicable for B-VOPs, i.e.,

$$\alpha_B = \frac{\bar{b}_B}{\bar{b}_P} \left(\frac{\bar{D}_B}{\bar{D}_P} \right)^{\gamma_r} \quad (6.93)$$

In this case, at the beginning of each B-VOP encoding, α_B should be updated according to (6.93).

VOP-LEVEL BIT ALLOCATION

At the VOP-level, i.e., inside each SP²⁶, in order to obtain approximately constant quality among the several VOPs composing the SP, each VOP should get allocated a nominal target number of bits that is a fraction of the SP target (6.88), proportional to the relative complexity of the VOP to be encoded in that particular time instant. Therefore, the nominal target number of bits for the VO n VOP in a given SP p of GOS m is given by the following equation

$$\bar{T}_{VOP}[n] = T_{SP} \frac{X_{VOP}[n]}{X_{SP}}, \quad n = 1, \dots, N_{VO} \quad (6.94)$$

For MVO encoding, it is important to guarantee that the spatial quality among the different VOs in the scene is kept approximately constant, i.e., an important goal is to encode all the objects in the scene with approximately constant quality. This goal can hardly be achieved when only a pure feedforward approach is used to compute the VO weights used to distribute the SP target among the several VOPs in the given SP. This is the approach followed in [77] and [29], where there is no compensation to deviations on the bit rate distribution among the several VOPs for a given encoding time instant. Therefore, it is important to adapt the VO coding complexity weights along time and to compensate the bit allocation deviations through the feedback adjustment of these parameters in order to meet the requirement of spatial quality smoothness.

In this Thesis, the following compensation mechanism is proposed, aiming at reducing the deviations in the average distortion among the several VOs composing the scene for a given SP. For this purpose, the SP average luminance pixel distortion is defined as the weighted sum of the various VOs distortions, i.e.,

$$D_{SP}[p] = \frac{1}{\sum_{k=1}^{N_{VO}} N_{PIX}[k]} \sum_{k=1}^{N_{VO}} N_{PIX}[k] \cdot D_{VO}[k] \quad (6.95)$$

where $N_{PIX}[k]$ is the number of pixels in VO k VOP in SP p .

Using (6.95) as the reference target SP distortion, a complexity weight adjustment is computed for each VO as follows

²⁶ Notice that, for single VO scenes, this level is not needed since each SP contains only one VOP. Therefore, no feedback compensation is required among temporal co-located VOPs.

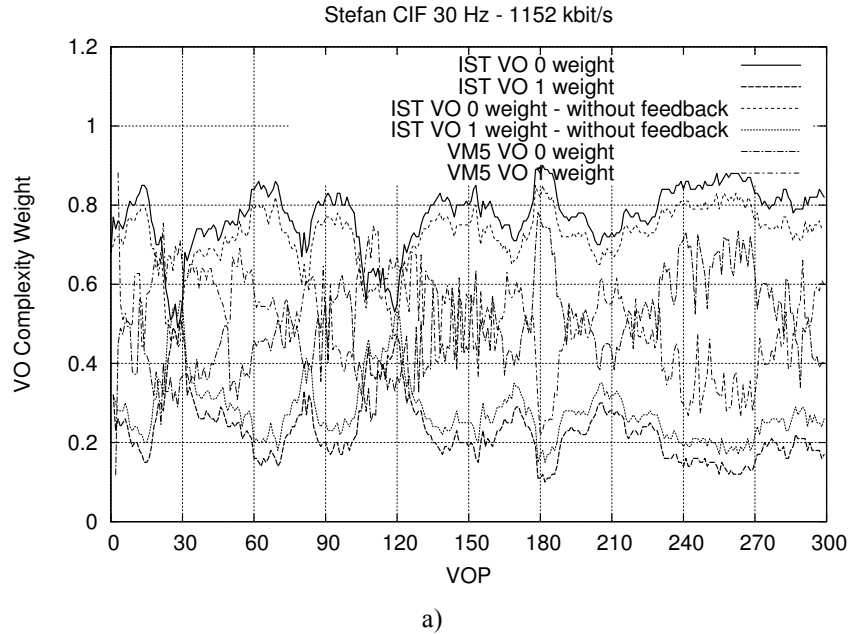
$$\phi_D[p][n] = \left(\frac{S_{VOP}[p-1][n]}{\alpha_T[p-1][n] \times \sum_{k=1}^{N_{VO}} S_{VOP}[p-1][k]} \times \frac{D_{VOP}[p-1][n]}{D_{SP}[p-1]} \right)^{\gamma_D} \quad (6.96)$$

where γ_D is a control parameter to control the impact of ϕ_D in the VOP bit allocation feedback compensation (typically, $0.1 \leq \gamma_D \leq 0.5$; in this Thesis, $\gamma_D = 0.2$ has been used).

Since the main objective of this compensation mechanism is to smooth the spatial quality differences among the different VOS, this VO complexity weight adjustment mechanism can be seen as a way to perform spatial quality control inside each SP.

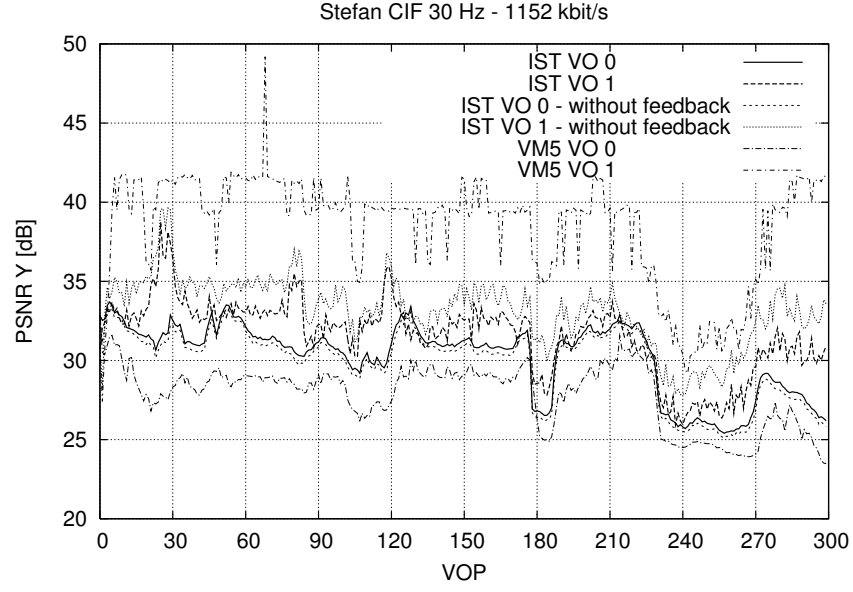
Figure 6.13 shows the VO weights, VO VOP luminance PSNR, and Scene PSNR²⁷ for two VOs of the *Stefan* sequence (VO 0 – *Background*, and VO 1 – *Player*), in CIF format, encoded at 30Hz with a target bit rate of 1152 kbit/s.

As can be seen in Figure 6.13, the feedback adaptation of the VO weights, as proposed in this Thesis²⁸, leads to a larger fluctuation of the VO weights (see Figure 6.13a) reflecting a better adaptation to the different VOs characteristics expressed by smoother VO PSNR differences (see Figure 6.13b) when compared with the MPEG-4 Visual VM5 solution [108] and with no feedback adaptation. Moreover, the Scene PSNR is not penalized by this adaptation, as can be seen in Figure 6.13c. In fact, a higher Scene PSNR is obtained by the proposed solution, when compared with the other two alternative solutions.

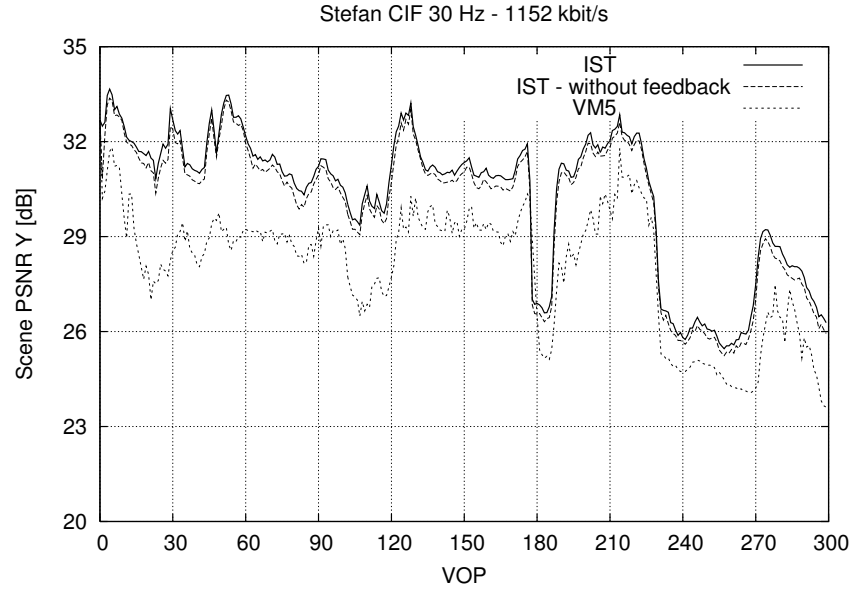


²⁷ See Section 6.7.

²⁸ The algorithm proposed in this Thesis is labeled IST.



b)



c)

Figure 6.13 – VO complexity weight estimation for the Stefan sequence: a) VO complexity weights; b) VO VOP PSNR; c) Scene PSNR

Based on (6.84) and (6.96), the VO feedback-adjusted complexity can be computed as follows

$$\eta_D[n] = \phi_D[n] \cdot X_{VOP}[n] \quad (6.97)$$

and subsequently

$$T_{VOP}[n] = T_{SP} \frac{\eta_D[n]}{\sum_{k=1}^{N_{VO}} \eta_D[k]} + \frac{\eta_D[n]}{\sum_{k=n}^{N_{VO}} \eta_D[k]} \sum_{k=1}^{n-1} \left(T_{SP} \frac{\eta_D[k]}{\sum_{k=1}^{N_{VO}} \eta_D[k]} - S_{VOP}[k] \right), \quad n = 1, \dots, N_{VO} \quad (6.98)$$

MB-LEVEL BIT ALLOCATION

At the MB-level, i.e., inside each VOP, in order to obtain approximately uniform quality among the several non-transparent MBs, each MB should get a nominal target number of bits that is a fraction of the VOP target (6.98), proportional to the relative complexity of the MB to be encoded in that particular VOP.

Recognizing, however, that a simple function can hardly relate a simple MB complexity measure with the number of bits generated when encoding the given MB with a certain quantization parameter, it is proposed in this Thesis to adopt the MB MAD as the MB complexity measure, X_{MB} , and compensate the deviations relatively to the ideal behavior with an integral feedback law with an adaptive integration factor.

The main advantage of this approach is to have a simple way to finely adjust the VOP quantization parameter with a single encoding step, which is a strong requirement for low-delay encoding, and additionally to be able to adaptively compensate the possible deviations relatively to the nominal target bit allocations before the end of the VOP, where the consequence of these deviations could be adverse (e.g., VBV violations) and the type of reaction required in this case would be extreme (e.g., skipping the next VOPs to encode).

Notice that, for Intra-coded MBs, the MB MAD measure is the mean absolute difference to the average MB luminance pixel values, while for Inter-coded MBs the MB MAD measure is the mean absolute difference of the motion estimation prediction luminance error.

Therefore, the nominal target number of bits to encode each MB in a given VOP is given by the following equation

$$\bar{T}_{MB}[i] = T_{VOP} \frac{X_{MB}[i]}{\sum_{k=1}^{N_{MB}} X_{MB}[k]}, \quad i = 1, \dots, N_{MB} \quad (6.99)$$

where N_{MB} is the number of MBs in the VOP being encoded.

As for the higher levels, i.e., the SP- and VOP-level, the MB feedback control scheme is a tracking control scheme since the nominal target number of bits for each MB changes according to the MB complexity measure, $X_{MB}[i]$. Therefore, the actual MB target number of bits can be expressed by the following equation

$$T_{MB}[i] = T_{VOP} \frac{X_{MB}[i]}{\sum_{k=1}^{N_{MB}} X_{MB}[k]} + K_{MB}[i] \cdot \sum_{k=1}^{i-1} \left(T_{VOP} \frac{X_{MB}[k]}{\sum_{k=1}^{N_{MB}} X_{MB}[k]} - S_{VOP}[k] \right), \quad i = 1, \dots, N_{MB} \quad (6.100)$$

where K_{MB} is the feedback integration factor.

Since in MPEG-4 Visual [29], the quantization parameter is restricted to change between adjacent MBs to a maximum absolute value of two, large deviations from the nominal MB target tend to saturate the compensation mechanism. Consequently, it is convenient to have a K_{MB} factor that does not lead to abrupt or very slow reactions to modeling errors.

In this context, K_{MB} is given by the following equation

$$K_{MB}[i] = \max \left[\frac{X_{MB}[i]}{\sum_{k=i}^{N_{MB}} X_{MB}[k]}, \frac{1}{\min[N_{MB} - i + 1, 16]} \right] \quad (6.101)$$

The rationale for (6.101) is the following: at the beginning of the VOP encoding, the bit allocation errors are compensated at most along the subsequent 16 MBs²⁹, avoiding slowly reactions for MBs with low complexities, i.e., low MADs; as the encoding proceeds to the last MBs, the K_{MB} factor distributes the accumulated bit allocation error through the remaining MBs to be encoded, according to their relative complexities.

As can be seen in Figure 6.14 and Figure 6.15, although the number of bits as a function of the MB MAD resembles a linear function, there is a considerable dispersion around this linear tendency that justifies the introduction of the feedback compensation mechanism.

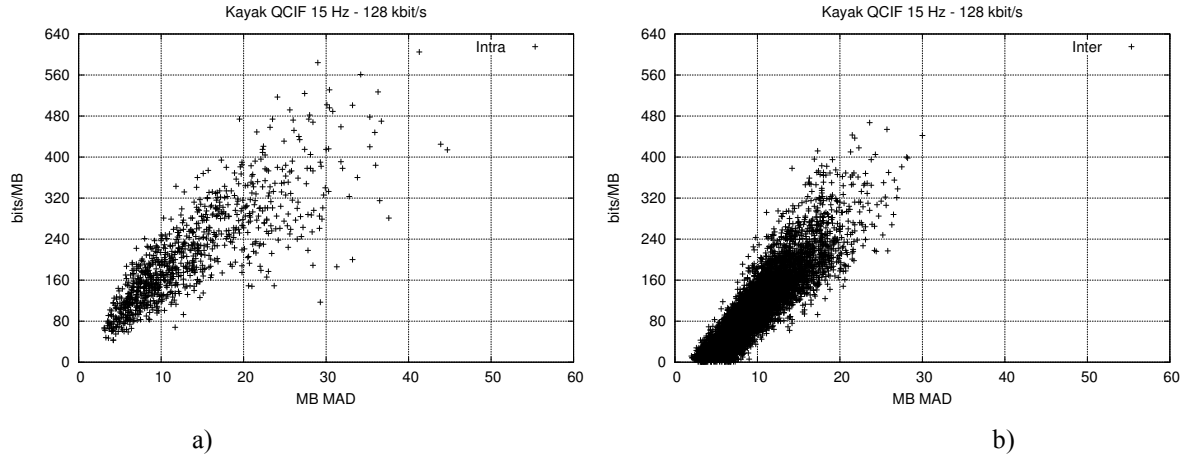


Figure 6.14 – Number of texture bits per MB as a function of the MB MAD for the Kayak sequence encoded with SP@L2: a) Intra-coded MBs; b) Inter-coded MBs

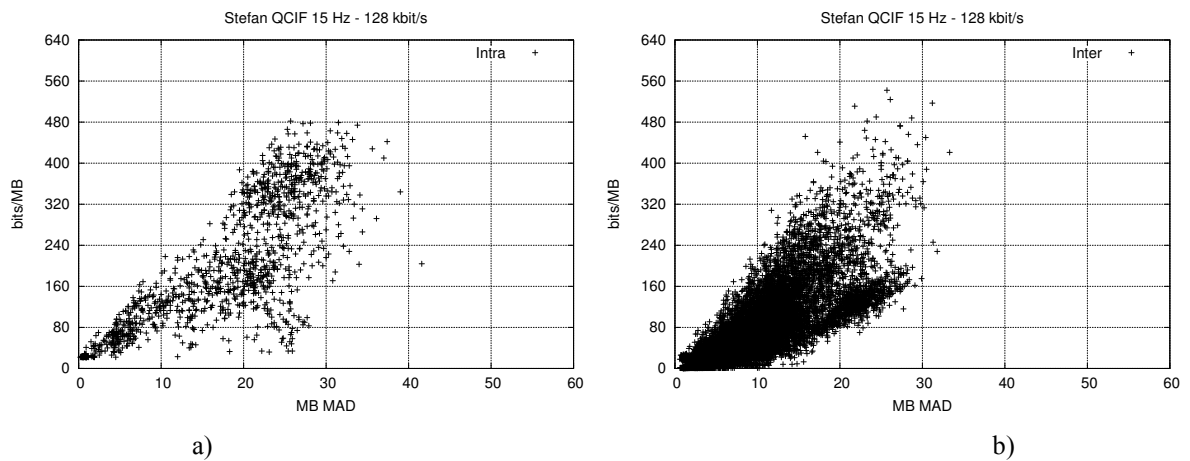


Figure 6.15 – Number of texture bits per MB as a function of the MB MAD for the Stefan sequence encoded with SP@L2: a) Intra-coded MBs; b) Inter-coded MBs

²⁹ This value has been set empirically.

BLOCK- AND DCT-LEVEL BIT ALLOCATION

In the context of this Thesis, there is no bit allocation at the Block- and DCT-level, which means that if enough space in the encoder bitstream buffer is available³⁰, all quantized DCT coefficients, for all blocks of a given MB are encoded, otherwise they are skipped (except Intra DC coefficients, for which the rate control algorithm allocates always enough space in the encoder rate buffer to transmit them).

The main novelty of this bit allocation module comes from its hierarchical nature that allows to develop adequate compensation mechanisms for each level and, consequently, achieve accurate bit allocations both among the different VOs composing the scene (MVO scenario) and along the sequence (SVO and MVO scenario). This bit allocation module constitutes one of the main strong points of the proposed rate controller.

6.4.6 Video Buffering Verifier Control

This rate controller module is responsible for controlling the scene encoder in such a way that the video buffering verifier mechanism is not violated and, therefore, the set of bitstreams produced by the encoder can be considered compliant with the profile@level selected to encode the given scene. The main purpose of this module is to provide guidance to the spatio-temporal resolution control, the bit allocation, and the coding mode control modules relatively to the status of the various video buffering verifier models, and consequently assist these modules in their respective tasks regarding the constraints of the video buffering verifier mechanism.

This controller module provides VMV, VCV, and VBV control, following the three verification steps described in Section 4.4, i.e.,

- Video Reference Memory Verification.
- Video Complexity Verification.
- Video Rate Buffer Verification.

Regarding the VMV and VCV control, this module is responsible for estimating, respectively, the total amount of memory and the total computational power required at the decoder, according to the VCV and VMV models. If not enough resources are available at the decoder according to these models, the video buffering verification control module signals this imminent violation to the spatio-temporal resolution control module in order adequate actions are taken.

With respect to the VBV control, the proper control of this mechanism requires more careful attention in order to avoid situations where extreme actions need to be carried out, leading to severe drops in the quality of the decoded data. Therefore, this Thesis proposes that VBV control should aim at preventing situations where strong measures have to be taken in order to avoid violations of the VBV mechanism, that is to say soft VBV control strategies should be favored over hard VBV control strategies (see Section 6.3).

To achieve accurate VBV control, this Thesis proposes to use a combined feedforward and feedback compensation mechanism. For that, the video buffering verification control module

³⁰ This means that no VBV buffer underflow will occur.

sets feedforwardly a target VBV buffer occupancy for each encoding time instant based on the amount and complexity of the VO data to encode and the relative position of the SP in the GOS. This way, the amount of compensation introduced by the feedback mechanism can be limited in order to avoid oscillations that can lead to buffer violations. Notice that [29] and [77] simply aim at achieving for every encoding time instant a VBV buffer occupancy equal to half the VBV buffer size, treating equally VBV buffer underflows and overflows, which is an oversimplification of the VBV control. This Thesis proposes that VBV control shall be conducted at the following levels:

- SP-level (MVO) or VOP-level (SVO).
- MB-level.

At each of these levels, the video buffering verifier control module tracks the deviation of the actual VBV buffer occupancy regarding the target occupancy and, based on this deviation, defines a compensation action. Whenever the algorithm detects a large deviation from the target VBV buffer occupancy, it considers that an imminent violation is foreseen and an extreme action, such as skipping VOPs or stuffing data, is required. Notice that, in order to maintain approximately constant spatial quality, the encoder should not dramatically reduce bit allocations in order to favor temporal continuity.

SP-LEVEL VBV CONTROL

For each target encoding time instant, the video buffering verifier control mechanism attempts to avoid imminent situations of VBV violations. First, in a defensive manner, i.e., through soft SP-level VBV control; secondly, in a more aggressive way, i.e., through hard SP-level VBV control.

I) Soft SP-level VBV Control

Soft SP-level VBV control is mainly achieved by a careful definition of a target VBV buffer occupancy that depends on the following parameters:

- The target number of bits to encode the current GOS.
- The relative time instant to the beginning of the GOS of each SP.
- The number of VOs and complexity of each VOP to be encoded in each SP.

For each SP of GOS m , the target VBV buffer occupancy immediately before removing all VOPs for the corresponding SP from the buffer is given by

$$B_T[p] = B_S - \left(T_{GOS} \frac{\sum_{k=1}^{p-1} X_{SP}[k]}{X_{GOS}} - (t_{SP}[p] - t_{SP}[1]) \times R \right) - B_L, \quad p = 1, \dots, N_{SP} \quad (6.102)$$

where B_S is the VBV buffer size, T_{GOS} is the target number of bits for encoding the whole GOS m given by (6.78), $X_{SP}[k]$ is the SP k complexity given by (6.85), X_{GOS} is the GOS m complexity given by (6.86), $t_{SP}[p]$ is the time instant of SP p , R is the average output target bit rate for GOS m , and B_L is the VBV underflow margin as explained in the following paragraphs.

Since at the beginning of each GOS all VOPs are Intra coded, this will typically lead to the

highest level in terms of encoder rate buffer occupancy, as Intra coded VOPs require typically more bits for achieving the same spatial quality than Inter coded VOPs. Consequently, in terms of VBV occupancy, this will correspond to the highest occupancy immediately before removing the first VOPs of a GOS and the lowest VBV occupancy immediately after removing these VOPs from the VBV buffer (see Figure 6.16).

In nominal terms, the available VBV margin is defined by the available encoder rate buffer space immediately after adding the encoded bits of the first SP in the GOS, or, in terms of VBV occupancy, by the occupancy of the VBV buffer immediately after removing the bits of the first SP VOPs.

Since VBV buffer underflow (encoder rate buffer overflow) is more critical than VBV buffer overflow (encoder rate buffer underflow), it is convenient to unequally distribute this nominal margin over these two critical zones. Therefore, at the beginning of each GOS, the VBV margin is computed as follows

$$B_M = B_S - T_{GOS} \frac{X_{SP}[1]}{X_{GOS}} \quad (6.103)$$

The target free space in the buffer is unequally partitioned as follows

$$B_L = \beta_{VBV} \times B_M \quad (6.104)$$

and

$$B_U = (1 - \beta_{VBV}) \times B_M \quad (6.105)$$

with $\beta_{VBV} = 0.9$ in the current implementation.

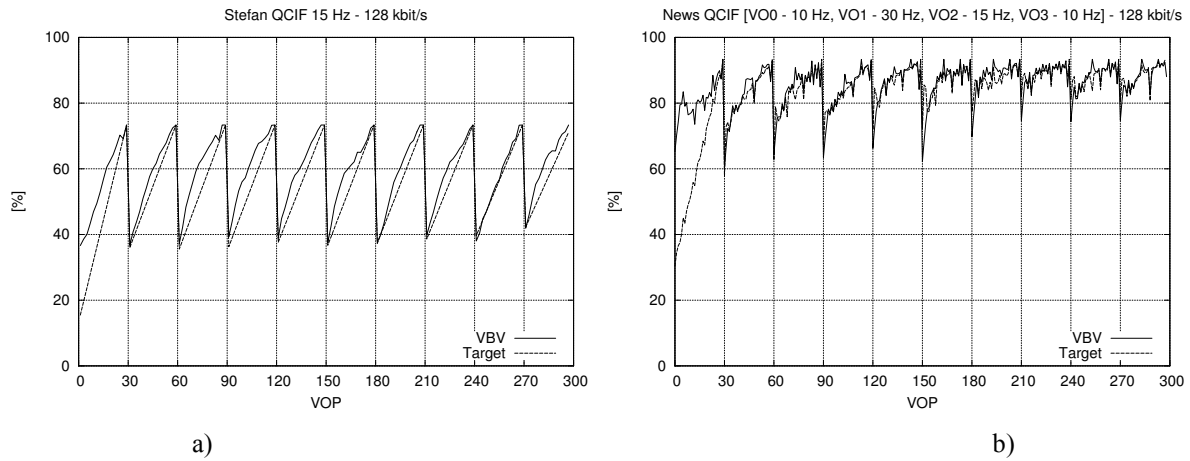


Figure 6.16 – Target and actual VBV buffer occupancy for each SP after VOP removing: a) SVO Stefan sequence; b) News sequence with 4 VOs encoded at different frames rates

Based on (6.102), the target number of bits used to encode the corresponding SP (6.88) is further adjusted by a multiplicative factor, K_{VBV} , given by the following expression

$$K_{VBV} = \begin{cases} 1 - \alpha_{VBV} \left(\frac{B_T - B}{B_T} \right) & \Leftarrow B \leq B_T \\ 1 + \alpha_{VBV} \left(\frac{B - B_T}{B_S - B_T} \right) & \Leftarrow B > B_T \end{cases} \quad (6.106)$$

where $\alpha_{VBV} = 0.25$ is a controller parameter set empirically.

The rationale for (6.106) is to decrease the SP bit allocation if the VBV buffer is approaching underflow (i.e., too many bits have been generated by the encoder in the past) and to increase the SP bit allocation if the VBV buffer is approaching overflow (i.e., too few bits have been generated by the encoder in the past).

Therefore, the bit allocation given by (6.88) is adjusted as follows

$$T_{SP}^{SBC}[p] = T_{SP}[p] \times K_{VBV} \quad (6.107)$$

II) Hard SP-level VBV Control

In some extreme cases, notably for small buffer sizes, the soft SP-level VBV control may lead to SP bit allocations near imminent violations of the VBV mechanism; therefore, whenever this situation occurs, a further adjustment is needed to correct it.

The first step in this adjustment is to check the VBV occupancy deviation relatively to the target VBV buffer occupancy, B_T (see Figure 6.17). In this case, if $B[p] < B_T[p] - B[p]$, it means that the VBV buffer occupancy is dangerously closer to VBV underflow than the desired target VBV buffer occupancy. In this case, the video buffering verifier control mechanism signals to the spatio-temporal resolution control module the imminent violation of the VBV indicating that the current encoding time instant should be skipped in order that the VBV buffer occupancy can reach a higher occupancy far from imminent VBV underflow.

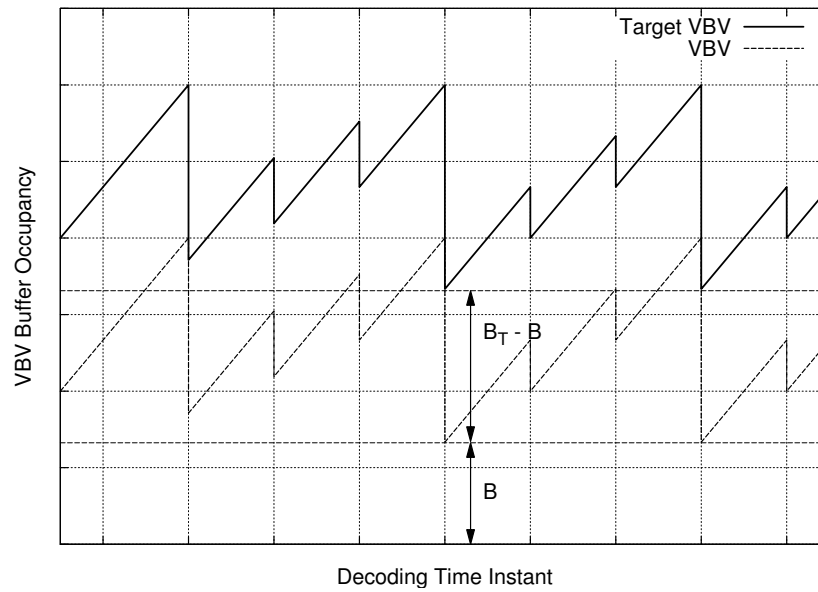


Figure 6.17 – VBV buffer occupancy deviation leading to imminent VBV underflow

For this purpose, a nominal VBV operation area is defined by the following VBV buffer occupancy conditions (see Figure 6.18)

$$B_L^{HBC} \leq B \leq B_U^{HBC} \quad (6.108)$$

where $B_L^{HBC} = \beta_L \times B_S$ and $B_U^{HBC} = \beta_U \times B_S$ (in this Thesis, $\beta_L = 0.05$ and $\beta_U = 1.0$).

Whenever the bit allocation given by (6.107) subtracted from the VBV buffer occupancy is below the underflow margin, B_L^{HBC} , the bit allocation is decreased by the amount of the excess; similarly, if the foreseen decoder occupancy is above the buffer size before the next decoding time instant, the bit allocation is increased by the corresponding amount, i.e.,

$$T_{SP}^{HBC}[p] = \begin{cases} B[p] - B_L^{HBC}[p] & \Leftarrow B[p] - T_{SP}^{SBC}[p] < B_L^{HBC} \\ B[p] + R_{SP}[p] - B_S & \Leftarrow B[p] - T_{SP}^{SBC}[p] > B_U^{HBC} - R_{SP}[p] \end{cases} \quad (6.109)$$

where

$$R_{SP}[p] = (t_{SP}[p+1] - t_{SP}[p]) \times R \quad (6.110)$$

is the number of bits drained from the buffer between two consecutive encoding time instants.

It is important to highlight, however, that even the hard VBV control mechanism expressed by (6.109) cannot guarantee full VBV compliance, since the sampling period for this mechanism is the SP encoding period. Therefore, reactions at the end of the SP encoding period may not be sufficient to avoid VBV violations, and for that reason a fine VBV control level is proposed in this Thesis. Nevertheless, it is important to state here that whenever a VBV buffer overflow is detected, the encoder mechanism adds stuffing data³¹ to the elementary streams.

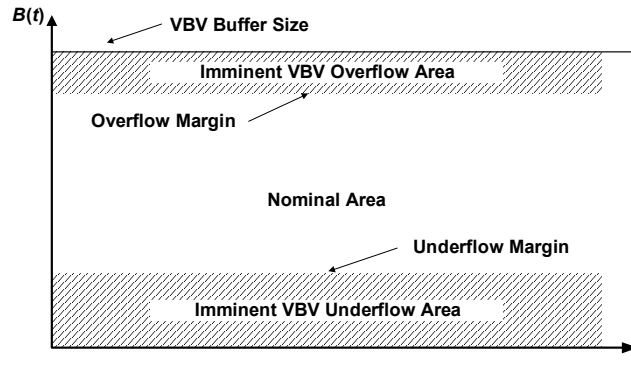


Figure 6.18 – Control limits to prevent violation of the VBV buffer

VOP-LEVEL VBV CONTROL

At the VOP-level, for MVO encoding, there is no explicit video buffering verifier control mechanism. This is handled with a fine granularity at the MB-level. Notice, however, that for SVO encoding, each SP contains a single VOP and, consequently, for each encoding time-instant, the methodology proposed above applies to each VOP.

³¹ In the case of rectangular VOs, this is achieved through the stuffing MB type, while for arbitrary shaped VOs, this is achieved through the stuffing start code.

MB-LEVEL VBV CONTROL

As referred above, to avoid violations of the VBV mechanism without dramatic spatial and temporal quality losses, a fine VBV control reaction is needed besides the SP-level. Consequently, during the encoding of each VOP, the rate control algorithm closely tracks and adjusts the VBV buffer occupancy tendency in order to avoid undesirable operational points as imminent VBV buffer underflows and, less persistently, imminent VBV buffer overflows.

I) Soft MB-level VBV Control

The purpose of the soft MB-level VBV control is essentially to regulate the allowable MB QP variation range, i.e., during normal operation the rate controller, through the coding mode control module, attempts to confine the MB QP variation inside each VOP in order to favor spatial quality smoothness; however, when there is an imminent VBV underflow, this restriction has to be cancelled in order to avoid possible VBV violations.

For this purpose, the algorithm defines a maximum target number of bits to encode a given MB through the following expression

$$T_{MB}^{\max}[i] = (B[i-1] - 10 \times \Delta_T[n]) \frac{X_{MB}[i]}{\sum_{k=1}^{N_{MB}} X_{MB}[k]}, \quad i = 1, \dots, N_{MB} \quad (6.111)$$

where $B[i-1]$ is the VBV buffer occupancy immediately before encoding MB i , and $\Delta_T[n]$ is the average number of target bits per MB assuming an uniform distribution along the VOP, i.e.,

$$\Delta_T[n] = \frac{T_{VOP}[n]}{N_{MB}[n]}, \quad n = 1, \dots, N_{VOP} \quad (6.112)$$

Based on (6.111) and the accumulated MB bit allocation error up to MB i , the algorithm defines a deviation measure, δ_{MB} , as follows

$$\delta_{MB}[i] = 1 + \frac{1}{T_{MB}^{\max}[i]} \sum_{k=1}^{i-1} \left(T_{VOP} \frac{X_{MB}[k]}{\sum_{k=1}^{N_{MB}} X_{MB}[k]} - S_{VOP}[k] \right), \quad i = 1, \dots, N_{MB} \quad (6.113)$$

If $\delta_{MB}[i] \leq 0.9$, the rate control algorithm signals an imminent VBV underflow condition (i.e., the encoder has been systematically spending more bits than the allocated), which will be used later to relax the QP variation limitation imposed to reduce the variation in spatial quality between consecutive MBs. This will be further explained below when the coding mode control module will be described.

II) Hard MB-level VBV Control

Hard VBV control at the MB-level is used in extreme cases, notably whenever the encoder detects that there will be no more space left in the encoder rate buffer to accommodate the encoded bits of a given MB. Therefore, some DCT coefficients will be skipped. In this case, a simple strategy is followed: for Intra coded MBs, only the DC coefficients are encoded; for Inter coded MBs, no DCT coefficients are encoded.

Notice that this is an extreme measure, typically only used when the target bit rate and the

buffer size are very small.

For a given MB, DCT coefficients will be skipped if the following condition holds

$$B[i-1] - (S_{MB}[i] + (N_{MB} - i) \times \hat{b}_{DCT}) < B_{MB}^{\min}, \quad i = 1, \dots, N_{MB} \quad (6.114)$$

where $B[i-1]$ is the VBV buffer occupancy immediately before encoding MB i , $S_{MB}[i]$ is the provisional number of encoded bits, $(N_{MB} - i)$ is the remaining number of MBs to encode in the VOP, \hat{b}_{DCT} is the estimated number of bits for encoding only the required DCT coefficients (for Intra VOPs, $\hat{b}_{DCT} = 46$; for Inter VOPs, $\hat{b}_{DCT} = 0$), and $B_{MB}^{\min} = 100$ is a VBV buffer occupancy threshold to guarantee some free space in the buffer to accommodate the VOP syntax information.

It is worthwhile mentioning here, that following the proposed strategy of performing VBV control at the scene-level and object-level allows to achieve an accurate and robust control of the VBV buffer occupancy and to efficiently balance the typically conflicting goals of proper VBV control and smooth spatio-temporal video quality variation. These two conflicting goals are not typically considered together in the reference methods (see Section 2.5) and consequently either the VBV is not so effective or the spatio-temporal quality is penalized (mainly due to excessive VBV underflows and consequent VOP skipping). Consequently, the VBV control performance is closely connected to the coding mode control module described in the following section.

6.4.7 Coding Mode Control

This module is conceptually responsible for deciding the appropriate coding parameters of each coding unit, i.e., MB texture and shape (if applicable) coding modes, MB motion vectors (if applicable), and MB quantization parameter.

In deciding the MB coding modes and motion vectors, the proposed rate control algorithm uses, essentially, the suggested methods of VM8 [108], described in Chapter 2. The only exception here is the MB skipping and stuffing decisions based on the video buffering verifier control module information, described in the previous section.

Therefore, in the proposed rate control algorithm, the coding mode control module aims essentially at controlling the quantization parameter variation for each VOP to be encoded.

VOP-LEVEL CODING MODE CONTROL

At the beginning of each VOP encoding, the rate controller computes a target quantization parameter for the current VOP, using the given bit allocation for the VOP and the corresponding rate-quantization model(s).

For I-VOPs, the target number of bits for the current VOP, T_{VOP} , given by (6.98) is used to compute a target quantization parameter, Q_T , through (6.59), i.e.,

$$Q_T[n] = \{Q : R(Q) = T_{VOP}[n]\} \quad (6.115)$$

For P-VOPs, the target quantization parameter computation requires three steps:

1. Using the stationary rate-quantization model (6.59) and the target number of bits for

the current VOP, T_{VOP} , given by (6.98), the coding mode control module computes a first approximation of Q_T , \tilde{Q}_T , i.e.,

$$\tilde{Q}_T[n] = \{Q : R(Q) = T_{VOP}[n]\} \quad (6.116)$$

2. Based on the delta rate-quantization model (6.61), the approximate target quantization parameter, \tilde{Q}_T , computed by (6.116), and the delta quantization parameter of the previous P-VOP, ΔQ^{prev} , computed by (6.62), the coding mode control module computes the estimated target deviation, ΔT_{VOP} .
3. Using the new target estimation, $\hat{T}_{VOP} = T_{VOP} + \Delta T_{VOP}$, the control module uses again (6.59) to compute the final estimate of the target quantization parameter, Q_T , i.e.,

$$Q_T[n] = \{Q : R(Q) = \hat{T}_{VOP}[n]\} \quad (6.117)$$

Notice that (6.117) intends to compensate the fact that when previous P-VOPs are encoded with higher or lower quantization parameters, the VO rate-quantization model tends to deviate from the stationary model as shown in Figure 5.18; therefore, the quantization parameter computed through the stationary model needs to be corrected, target that is achieved through the last two steps above.

Since one of the goals of the proposed rate control algorithm is to provide smooth quality variations along time, this Thesis proposes to restrict the quantization parameter variation between consecutive VOPs of the same VO similarly as it is performed in [29]. This way, the target quantization parameter, Q_T , at the beginning of each VOP is restricted to the following interval.

$$Q_T \in \left[\max\left(1, (1 - \gamma_Q) Q_T^{\text{prev}}\right), \min\left(31, (1 + \gamma_Q) Q_T^{\text{prev}}\right) \right] \quad (6.118)$$

where Q_T^{prev} is the quantization parameter used to encode the previous VOP of the same coding type, and γ_Q is a control parameter (as in [29], $\gamma_Q = 0.25$).

MB-LEVEL CODING MODE CONTROL

As referred in Section 6.4.4, the purpose of MB-level quantization selection is to provide a fine way to compensate deviations relatively to the nominal bit allocations. This task requires, however, dealing with conflicting goals under very restrictive degrees of freedom for MB quantization parameter variation.

In order to maintain the spatial quality inside each VOP approximately constant, the quantization parameter used to encode each MB of a given VOP should be kept approximately constant. In fact, recognizing this important goal, the MPEG-4 Visual standard limits the range of variation of the quantization parameter between consecutive MBs inside each VO to $\{-2, -1, 0, +1, +2\}$, and, in addition, the quantization parameter can only be changed for some MB coding types (see Table B-1 in [29]).

These restrictions, however, can lead to large deviations between bit allocations and actual encoded bits. Therefore, the rate control algorithm should closely track these deviations in order to take adequate and timely measures to avoid violations of the VBV mechanism and possibly some extreme measures that may have a high impact in terms of spatial and temporal

quality changes, such as skipping encoded data (e.g., some MBs or a complete VOP).

To deal with the described conflicting goals, this Thesis proposes the MB selection mechanism described below.

For each MB to be encoded, the algorithm defines two reference bit allocations: i) uniform bit allocation expressed by (6.119), that will be used as reference to measure bit allocation deviations; ii) rate-quantization bit allocation expressed by (6.120), defining the bit allocation for the current MB if the previous MB quantization parameter, $Q_{MB}[i-1]$, is used, i.e.,

$$\Delta_T = \frac{T_{VOP}}{N_{MB}} \quad (6.119)$$

$$T_{MB}^{RQ}[i] = R_{MB}(Q_{MB}[i-1]) \quad (6.120)$$

Subsequently, it computes the deviation between the actual MB bit allocation (6.100) and the rate-quantization bit-allocation (6.120), i.e.,

$$\varepsilon_{MB}[i] = T_{MB}[i] - T_{MB}^{RQ}[i] \quad (6.121)$$

If this deviation is less than a given threshold, the MB quantization parameter remains unchanged; otherwise, the MB quantization is computed from the MB rate-quantization function, i.e.,

$$Q_{MB}[i] = \begin{cases} Q_{MB}[i-1] & \Leftarrow |\varepsilon_{MB}[i]| < \frac{\Delta_T}{2} \\ R_{MB}^{-1}(T_{MB}[i]) & \Leftarrow |\varepsilon_{MB}[i]| \geq \frac{\Delta_T}{2} \end{cases} \quad (6.122)$$

Notice, however, that the MB quantization parameter must be restricted to verify the following condition, imposed by the MPEG-4 Visual syntax [29]

$$|Q_{MB}[i] - Q_{MB}[i-1]| \leq 2 \quad (6.123)$$

In order to favor smooth MB quantization parameter variations inside each VOP, and avoid large deviations between the target VOP quantization parameter, Q_T , and the average MB quantization parameter at the end of the VOP, the proposed rate control algorithm attempts to bound the MB quantization parameter around Q_T by defining the following lower and upper limits

$$\begin{aligned} Q_{MB}^{\min} &= \max[Q_L, (1 - \gamma_Q)Q_T] \\ Q_{MB}^{\max} &= \max[Q_L, (1 + \gamma_Q)Q_T] \end{aligned} \quad (6.124)$$

where $Q_L = 2$ for I-VOPs and $Q_L = 4$ for P-VOPs.

Therefore, Q_{MB} is further adjusted as follows

$$Q_{MB}[i] = \begin{cases} Q_{MB}^{\min} & \Leftarrow Q_{MB}[i] \leq Q_{MB}^{\min} \\ Q_{MB}[i] & \Leftarrow Q_{MB}^{\min} < Q_{MB}[i] < Q_{MB}^{\max} \\ Q_{MB}^{\max} & \Leftarrow Q_{MB}[i] \geq Q_{MB}^{\max} \end{cases} \quad (6.125)$$

It is important to notice, however, that the limits set by (6.124) can be too restrictive, notably

for low bit rate conditions. Therefore, following a gain scheduling control approach, the proposed rate control algorithm defines as a scheduling variable for each VOP the average number of bits per pixel allocated to the current VOP, R_{bpp} , such that if $R_{bpp} \leq \varepsilon_T$, the limits set by (6.124) are alleviated, i.e., $Q_{MB}^{\min} = 1$, and $Q_{MB}^{\max} = 31$ ³², where $\varepsilon_T = 0.13$ ³³.

Another circumstance where the limits set by (6.124) can also be too restrictive is when the algorithm signals an imminent VBV buffer underflow – see (6.113) above. In this case, the coding mode control module uses (6.113) as scheduling variable to change the allowable $Q_{MB}[i]$ range based on the VBV buffer occupancy tendency. In case of imminent VBV buffer underflow, the coding mode control does not allow the quantization parameter to decrease, i.e.,

$$0 \leq Q_{MB}[i] - Q_{MB}[i-1] \leq 2 \quad (6.126)$$

With this restriction, situations where a few low complexity MBs could lead the quantization parameter to decrease dangerously are avoided. Notice that the quantization parameter cannot change arbitrarily between consecutive MBs.

6.4.8 Summary of the Proposed Rate Control Algorithm

The main purpose of this section is to summarize the integration of the various modules proposed for the rate control algorithm presented in this Thesis in a way that it can be implemented with relative low degree of uncertainty. Therefore, the algorithm will be described step-by-step, referencing for each step the main equations presented in Section 6.4. In this description, it is assumed that the VOs are available to the scene encoder through the scene buffer (see Figure 6.5).

It is worthwhile mentioning here, that some rate controller modules interact iteratively with each other (e.g., the spatio-temporal resolution control and the scene analysis for resource allocation modules or the coding mode control and the video buffering verifier modules). Therefore, a step-by-step description of the proposed algorithm can hardly associate each step to a single rate controller module without compromising the simplicity and legibility of this description.

For a given scene composed by a set of VOs, the rate control algorithm requires from the user the following parameters:

- Profile and level for encoding the scene.
- Target temporal resolution for each VO.
- Common random access point period (I-VOP period).
- Target average channel bit rate, R .
- Video rate buffer size, B_s .

The proposed rate control algorithm can be implemented through the following six steps:

³² Notice that setting $Q_{MB}^{\min} = 1$ is virtually useless in low bit rate conditions; however, allowing Q_{MB} to take any possible value in the range of allowable quantization parameter values gives the coding mode control module a higher degree of freedom to change the MB quantization parameter in restrictive bit allocation conditions.

³³ An example of low bit rate conditions are: QCIF@15Hz [48 kbit/s], where $R_{bpp} = 0.126$.

STEP 1 – Initialize Rate Controller

Initialize video buffering verifier model parameters (see Section 4.2)
 Check profile and level limits (see Section 2.6)
 Compute scene base temporal resolution, SR (see Section 6.4.3)
 Set rate controller parameters {
 Rate-Distortion Modeling:
 $W_{RQ} = 10$, $\varepsilon_{FIT} = 10^{-3}$, $N_{ITER} = 10$
 Bit Allocation:
 $\alpha_S = 0.2$, $\alpha_A = 0.5$, $\alpha_C = 0.3$
 $\alpha_I = \min \left[\frac{B_S}{R_{GOV}/SR}, 2.7 \right]$, $\alpha_P = 1.0$, $\alpha_B = 0.5$, $W_I = 3$, $\gamma_T = 0.5$
 $\gamma_D = 0.2$
 Video Buffering Verifier Control:
 $\beta_{VBV} = 0.9$, $\beta_L = 0.05$, $\beta_U = 1.0$
 $\hat{b}_{DCT} = 46$ (I-VOPs), $\hat{b}_{DCT} = 0$ (P-VOPs), $B_{MB}^{\min} = 100$
 Coding Mode Control:
 $\gamma_Q = 0.25$, $\varepsilon_T = 0.13$, $Q_L = 2$ (I-VOPs), $Q_L = 4$ (P-VOPs)
 }

STEP 2 – Spatio-Temporal Resolution Control (Compute Next Encoding Time Instant)

Set target next encoding time instant += $1/SR$ (next SP)
 Set delta_time = target next encoding time instant – previous encoding time instant

Scene Analysis for Resource Allocation (for next SP) {

 For each VOP in SP {
 Read VO VOPs from Scene Buffer
 Count MBs {
 MB_transp = #{Transparent MBs in VO VOP}
 MB_opaque = #{Opaque MBs in VO VOP}
 MB_bound = #{Boundary MBs in VO VOP}
 MB_total = #{Total MBs in VO VOP}
 }
 Compute S_{VOP} (6.55), A_{VOP} (6.56), and C_{VOP} (6.57)
 }
 }
 Check video buffering verifier status for next encoding time instant {
 Estimate VMV buffer occupancy for target next encoding time instant {
 Estimate released reference memory, VMV_MB_released
 $E[VMV] = VMV - VMV_MB_released + MB_total$
 If $E[VMV] > VMV_SIZE \Rightarrow$ skip
 }
 If (!skip) Estimate VCV occupancy for target next encoding time instant {
 Estimate VCV_MB_released = VCV_rate \times delta_time
 $E[VCV] = VCV - VCV_MB_released + MB_total$
 If $E[VCV] > VCV_SIZE \Rightarrow$ skip
 }
 If (!skip) Estimate B-VCV occupancy for target next encoding time instant {
 Estimate B-VCV_MB_released = B-VCV_rate \times delta_time
 $E[B-VCV] = B-VCV - B-VCV_MB_released + MB_bound$
 If $E[B-VCV] > B-VCV_SIZE \Rightarrow$ skip
 }
 If (!skip) Estimate VBV occupancy for target next encoding time instant {
 Estimate bits_filled = $R \times$ delta_time
 Estimate bits_drained = R/SR

```

    E[VBV] = VBV + bits_filled – bits_drained
    If E[VBV] < 0 => skip (VBV underflow)
    Compute target VBV occupancy,  $B_T$ , (6.102)
    If  $B < B_T - B$  => skip (hard VBV buffer control)
  }
}
If (skip) GOTO STEP 2

```

STEP 3 – Update Rate Controller Parameters for current SP

```

For each VO in SP {
  Update VO Rate-Distortion Models (6.60) for I-VOPs and (6.60) + (6.64) for P-VOPs
}
Update video buffering verifier models for current SP (VMV, VCV, B-VCV, and VBV)

```

STEP 4 – Bit Allocation (Scene-Level)

```

If first SP of GOS compute target number of bits for current GOS,  $T_{GOS}$ , (6.78)
Compute  $\bar{S}_{VO}$  (6.81),  $\bar{A}_{VO}$  (6.82), and  $\bar{C}_{VO}$  (6.83)
For each VO {
  Compute the VO coding complexity weight,  $\omega$ , (6.80)
  If first SP of GOS estimate new coding type weight,  $\alpha_I$ , (6.92)
  Compute the VOP coding complexity,  $X_{VOP}$ , (6.84)
}
Compute the SP coding complexity,  $X_{SP}$ , (6.85)
Estimate the GOS coding complexity,  $X_{GOS}$ , (6.86)
Compute the target number of bits for the current SP,  $T_{SP}$ , (6.88)
Perform soft SP-level VBV control (6.107)
Perform hard SP-level VBV control if necessary (6.109)
Estimate the Header,  $S_{header}$ , Motion,  $S_{motion}$ , and, Shape,  $S_{shape}$ , bits for the current encoding time instant
Check  $T_{SP}$  regarding spatio-temporal resolution control {
  If  $T_{SP}[SP] \leq E[S_{header}[SP] + S_{motion}[SP] + S_{shape}[SP]]$  => skip
}
if (skip) GOTO STEP 2
For each VO in current SP {
  Compute VO feedback adjusted complexity,  $\eta_D$ , (6.97)
  Compute target number of bits for the current VO VOP,  $T_{VOP}$ , (6.98)
}

```

STEP 5 – Coding Mode Control (VOP Encoding)

```

For each VO VOP in current SP {
  Compute VOP target quantization parameter,  $Q_T$ , (6.115) for I-VOPs and (6.117) for P-VOPs
  Adjust target quantization parameter for temporal quality smoothness (6.118)
  Initialize MB-level coding mode control {
    Initialize MB-level rate-quantization parameter,  $\alpha$ , (6.75)
    Initialize MB-level quantization parameter bounds (6.124)
  }
  For each MB in current VOP {
    Compute MB target number of bits,  $T_{MB}$ , (6.100)
    Compute MB quantization parameter,  $Q_{MB}$ , (6.122)
    Adjust the MB quantization parameter according to (6.125)
    Compute soft MB-level VBV control variable,  $\delta_{MB}$ , (6.113)
    Perform soft MB-level VBV control adjusting of  $Q_{MB}$  (6.113) and (6.126)
    Encode MB
    Perform hard MB-level VBV control according to (6.114)
  }
}

```

```

        Adjust MB-level rate quantization model parameter,  $a$ , (6.73)
    }
    Update VBV occupancy
}

```

STEP 6 – GOTO STEP 2

Steps 1–6 describe the rate control algorithm proposed in this chapter in terms of the natural execution flow. However, in order to better understand its main features in terms of the object-based representation approach, the next section will present a complementary description of this algorithm following the rate control framework proposed in Chapter 3.

6.5 Scene-level and Object-level Rate Control Breakdown

This section analyzes the proposed rate control algorithm in terms of the scene-level/object-level rate control framework proposed in the Chapter 3. Its main purpose is the mapping of the various rate control proposed modules into this framework, presenting the same rate control solution using different structuring dimensions.

As referred in Chapter 3, the object-based video coding approach, such as the MPEG-4 Visual approach [29], where a video scene is composed by several video objects, requires that the rate control is performed by using two levels: the scene-level rate control and the object-level rate control. This Thesis tackles this problem by proposing a new rate control algorithm composed of new scene-level and object-level rate control algorithms for low-delay MPEG-4 video encoding. Previous MPEG-4 related video rate control solutions [14, 77, 78] assume synchronous VOs, this means all VOs are coded at the same VOP rate. However, this approach may reveal itself inefficient since the several VOs in the scene may exhibit very different needs in terms of temporal resolution, notably during object fast movements and stationary periods. In this context, this Thesis presents new scene-level and object-level rate control algorithms capable of performing bit allocation for the several VOs in the scene, encoded at different VOP rates and achieving an efficient trade-off among spatial and temporal quality for the overall scene. The proposed approach is supported by the architecture presented in Figure 6.5.

6.5.1 Scene-level Rate Control

The main goal of the proposed scene-level rate control is to efficiently allocate the available resources among the several VOs by performing the following tasks:

- Adapt the VOP coding rate of each object along time according to the Video Buffering Verifier status and the VO characteristics.
- Adaptively select the target number of bits for each encoding time instant based on the changing characteristics of each VO in the scene, minimizing quality fluctuations along time.
- Provide adequate scene-level VBV control to prevent violations of this mechanism.

To achieve these goals, the rate controller undertakes first an analysis step. Based on the information collected during this step and the video buffering verifier status, the temporal resolution of each VO is adapted accordingly. Afterwards, performs the bit allocation at the scene-level, which is, finally, adapted to meet the VBV requirements aiming at smooth quality variations along time and among the several VOs for each encoding time instant. The main components of the proposed rate control algorithm at scene-level are briefly described below.

SCENE ANALYSIS FOR RESOURCE ALLOCATION

This module is responsible for extracting relevant information from the input data based also on the previous encoding time instants history; therefore, this task has to be handled at scene-level since it provides information necessary to assist the other scene-level modules on deciding, which time instants, and, eventually, which VOs to encode.

A key feature of the proposed scene-level rate control algorithm is the scene analysis for resource allocation that is carried out prior to each encoding step for each encoding time instant. This task is performed before encoding any VOP, for all the VOPs to be encoded at the time instant under consideration.

This scene analysis module receives as input, from the scene buffer, the set of original VOPs to be encoded, for each particular time instant, and, from the symbol generator, the corresponding set of previously reconstructed VOPs stored in the prediction memory. After, based on the past encoding results of each VO and the VOs current time instant characteristics, computes a set of relevant information for the other modules, notably the video buffering verifier, the bit allocation, the spatio-temporal resolution control, and coding mode control.

Therefore, to achieve these goals, the algorithm undertakes first an analysis step for each possible encoding time instant (for 25Hz content, every 40 ms) extracting relevant characteristics of the several VOs in the scene, such as the size, the object activity, and the prediction error energy.

SPATIO-TEMPORAL RESOLUTION CONTROL

This module is conceptually responsible for deciding the appropriate spatial and temporal resolutions of the various VOs in the scene given the VOs characteristics and the available resources defined by the status of the different video buffering verifier mechanism. Since the outcome of this module implies the global analysis of all VOs in the scene (e.g., changing the spatial and temporal resolution of only some VOs of segmented scenes requires special consideration, due to the possible generation of holes in the decoded composed scene), this is a typical task that should be handled at scene-level, i.e., with knowledge of all the VOs to encode for a given time instant.

In the proposed rate control algorithm, temporal resolution control is closely related to the video buffering verifier control, i.e., whenever any of the video buffering verifier buffers signals an imminent violation of the corresponding model, the rate control mechanism immediately adapts the temporal resolution of the video objects composing the scene. This adaptation is typically accomplished by skipping one or more encoding time instants, if the violation is localized, or by decreasing each VO temporal resolution, if the scene is persistently too demanding.

BIT ALLOCATION – SCENE-LEVEL

Due to the complex nature of a scene encoder in a MVO encoding scenario, this Thesis proposes to partition the bit allocation task into several hierarchical levels, similarly to the syntactic organization of the encoded video data. In this context, the bit allocation should be handled, in a first stage, at scene-level, dealing with the joint allocation of bits for the different encoding time instants and for the different VOs to be encoded in each time instant, and, in a second stage, at object-level, dealing with the allocation of bits inside each VOP.

Therefore, the goal of the bit allocation module at scene-level is to control the allocation of

bits along the following hierarchical levels of the video sequence: GOS-, SP-, and VOP-level (see Section 6.4.5)

I) GOS-level Bit Allocation

The GOS-level is the higher hierarchical level of bit allocation associated to the scene random access period. At this level, in a CBR scenario, the main goal is to allocate a nominal number of bits to encode each GOS that should be proportional to the duration of the GOS. This is accomplished by (6.77) and deviations from the nominal targets are compensated by (6.78).

II) SP-level Bit Allocation

Jointly controlling the encoding of multiple video objects with different VOP rates poses some problems to the bit allocation since the number of VOPs to encode for each encoding time instant is not constant and, additionally, the VOs characteristics also change along time.

In order to reduce quality fluctuations, both along time and among the several video objects in the scene, the bit allocation module needs to change the bit allocation for each encoding time instant, i.e., for each SP, according to the number of VOPs to encode and to their complexities. This fact leads typically to a non-uniform bit allocation even when the overall scene is encoded at CBR. The nominal bit allocation for each SP is given by (6.87) and deviations from the expected results are compensated by (6.88).

Notice that the SP bit allocation may need to be further adjusted with input from the video buffering verifier control, using (6.107) for soft SP-level VBV control and (6.109) for hard SP-level VBV control, in order to prevent VBV violations.

III) VOP-level Bit Allocation

The next step of the bit allocation at scene-level is the distribution of the SP target number of bits among the several VOPs to encode for each time instant. The nominal bit allocation for each VO VOP is given by (6.94) and deviations from the expected results are compensated by (6.98), which incorporates two types of compensation: VO complexity weight adjustment to compensate spatial quality differences among the various VOs in the scene, and target bit rate feedback compensation due to the deviations for the previous encoding time instants.

VIDEO BUFFERING VERIFIER CONTROL – SCENE-LEVEL

As referred above, the main purpose of the video buffering verifier control module is to assist the other rate controller modules with respect to the constraints of the video buffering verifier mechanism, notably the spatio-temporal resolution control, the bit allocation, and the coding mode control modules.

From the three video buffering verifier mechanisms that have to be handled by this module, i.e., the VMV, VCV, and VBV, the first two should be handled at scene-level, since their main purpose is to verify scene-level restrictions, e.g., total amount of memory and total computational power required at the decoder.

With respect to the VBV control, its proper control requires that it should be handled, in a first stage, at the scene-level, to guide the spatio-temporal resolution and the bit allocation (at scene-level) modules. In a second stage, at the object-level, notably, at the MB-level inside each VOP, the rate controller provides a fine granularity VBV control that allows preventing violations of this mechanism without dramatic spatial and temporal quality losses.

Therefore, at the SP-level, the video buffering verifier control mechanism attempts to avoid

imminent situations of VBV violations. First, in a defensive manner, through soft SP-level VBV control by setting a target VBV buffer occupancy through (6.102) for each SP and compensating the SP bit allocations through (6.107); secondly, in a more aggressive way through hard SP-level VBV control using (6.109).

6.5.2 Object-level Rate Control

At the scene-level the proposed rate control algorithm computes the bit allocations for each VO to be encoded for the time instant under consideration and compensates these bit allocations according to the video buffering verifier control (scene-level) guidance. After the scene-level operations are completed, the rate control continues at the object-level. The object-level operations include the allocation of bits at the MB-level and the computation of the optimal coding parameters to achieve the target bit allocation given by (6.98) based on the rate-distortion characteristics of each VO. As referred above, these actions also include a fine VBV control at the object-level.

RATE-DISTORTION MODELING

The optimization of the encoding process is usually a very demanding task that can be simplified by using models of the coding process. The problem faced in video coding is, typically, the maximization of some quality measure given some restrictions, notably buffer and bit rate related. For this, a good prediction of the rate and distortion obtained while encoding with a given set of coding parameters is essential. Therefore, the proposed rate-control algorithm adopts two types of object-level rate-distortion models: VOP and MB rate-quantization models.

I) VOP-level Encoder Rate-Distortion Modeling

For each VO, at the VOP-level, the rate control algorithm uses a rate-quantization model for each VO and for each VOP coding type, in order to estimate the initial quantization parameter at the beginning of each VOP encoding; in this Thesis, it is proposed that this estimation is based on (6.59), for I- and P-VOPs. For P-VOPs, an additional delta rate-quantization model (6.61) is used to compensate deviations from the stationary rate-quantization model assumption. In this case, firstly, based on the target number of bits to encode the given VOP through (6.59), the rate control mechanism computes an initial estimate of the VOP quantization parameter; secondly, through (6.61), the rate control mechanism computes the rate deviation, $\Delta R(Q)$; finally, with the new target estimate $T_{VOP} + \Delta R(Q)$, the algorithm returns to (6.59) to re-estimate the VOP quantization parameter.

After each encoding time instant, based on the encoding results and parameters used, the rate control mechanism updates each model through (6.60) and (6.64), providing a way for the dynamic model adaptation.

II) MB-level Encoder Rate-Distortion Modeling

As referred previously, the purpose of this level of modeling is to provide a fine level of rate control, i.e., with a lower sampling period – MB period – and consequently provide the rate controller with a faster reaction to deviations regarding the nominal operation.

For each VO at the MB-level, the rate controller, through the coding mode control module, uses the MB rate-quantization model (6.67) to estimate the number of bits that will be generated by the given MB and based on the status of the VBV selects the appropriate MB

quantization parameter taking into account also spatial quality smoothness criteria.

BIT ALLOCATION – OBJECT-LEVEL

Bit allocation at the object-level is carried on inside each VO VOP at the MB-level in order to obtain approximately uniform quality among the several non-transparent MBs. Therefore, each MB should get a nominal target number of bits that is a fraction of the VOP target (6.98), proportional to the relative complexity of the MB to be encoded in that particular VOP.

In this case, the nominal bit allocation for each MB in a given VOP is given by (6.99) and deviations from the expected results are compensated using (6.100).

Additionally, the MB bit allocation may need to be further adjusted in order to prevent violations of the VBV with input from the video buffering verifier control using (6.111) for soft MB-level VBV control.

VIDEO BUFFERING VERIFIER CONTROL – OBJECT-LEVEL

Video buffering verifier control at the object-level is carried on inside each VO VOP at the MB-level aiming at avoiding possible violations of the VBV mechanism, notably, in bit allocation restrictive conditions.

Since the proposed rate-control algorithm aims at achieving smooth spatial quality inside each VOP, this algorithm attempts to reach this goal by favoring approximately constant quantization parameter inside each VOP, which may lead to imminent violations of the VBV mechanism.

Therefore, the purpose of the video buffering verifier control at the object-level is to signal these situations through (6.113) – soft MB-level VBV control; and, in extreme conditions, to signal the necessity of skipping some DCT coefficients through (6.114) – hard MB-level VBV control.

CODING MODE CONTROL

As referred previously, the coding mode control module aims at deciding the appropriate coding parameters of each MB, therefore, being a typical object-level rate control task.

In the proposed rate control algorithm, this module is responsible for computing the target quantization parameter for each VOP through (6.115) and (6.117), respectively for I- and P-VOPs, with a further temporal quality smoothness adjustment through (6.118). Additionally, it is responsible, for a fine adjustment of the MB quantization parameter taking into account both quality and VBV restrictions through (6.122) – (6.126).

6.6 Quality Control in the Proposed Rate Control Algorithm

Since this is a very important issue in an efficient rate control algorithm, this section intends to highlight the main aspects of the proposed rate control algorithm addressing specifically the spatial and temporal quality constraints, notably those aspects contributing to improve the spatial and temporal quality smoothness regarding already existing rate control solutions.

As referred in Chapter 3, one of the main goals of a rate control mechanism is to keep the quality of a given visual object sequence³⁴ [29] approximately constant along the temporal and

³⁴ Generically, a visual object sequence is composed by multiple video objects with different sizes and temporal

spatial axes, i.e., among consecutive encoding time instants – Inter SP – and, both among the different VOPs of a given SP – Intra SP – and inside each VOP – Intra VOP.

6.6.1 Temporal Inter SP Quality Control

Temporal quality control is mainly related to maintain smoothly varying spatial quality along the different SPs, i.e., Inter SP. The strategy proposed in this Thesis to deal with temporal quality control can be divided in two steps:

1. Allocation of bits at scene-level for each SP taking into account the number, coding complexity, and coding type weight of the VOs to be encoded in that SP, using (6.88). This approach leads typically to non-uniform bit allocations along time, notably for MVOs encoded at different temporal resolutions.
2. Limitation at the object-level of the quantization parameter range variation between consecutive VOPs of a given VO of the same coding type through (6.118) aiming at achieving approximately constant quantization parameter between the VOPs of the same coding type.

While step 1 uses the VOP coding type weights, α_T , of each VO, trying to achieve an approximate constant distortion between the various VOP coding types, step 2 aims at keeping the average quantization parameter of each VOP coding type approximately constant, by this way maintaining the quality between the VOPs of the same coding type also constant.

6.6.2 Spatial Intra SP Quality Control

Spatial quality control for a given SP is mainly related to maintaining smooth spatial quality differences between the different VOs composing the scene. For MVO encoding, an important goal of the rate control algorithm is to achieve approximately constant quality between the several VOs composing the scene, provided that enough bit rate is available for encoding the visual object sequence; otherwise, some prioritization of VOs can be made [14].

The mechanism proposed for achieving this goal, in the context of this Thesis, consists in the feedback adaptation at the scene-level of the different VO coding complexity weights through (6.97), as described in Section 6.4.5.

6.6.3 Spatial Intra VOP Quality control

Spatial intra VOP quality control aims at maintaining the spatial quality inside each VOP approximately constant. For this, the quantization parameter used to encode each MB of a given VOP should be kept approximately constant.

Therefore, the proposed rate control algorithm attempts to accomplish this goal by favoring the steadiness of the MB quantization parameter, as expressed by (6.122), and by changing it only when it is recognizably necessary, notably when imminent VBV violations are signaled, as expressed by (6.113).

The combination of the spatial intra VOP quality control with the MB-level VBV control mechanisms allows smooth variations of the VOP quality while still producing MPEG-4 Visual compliant elementary streams.

6.7 Test Conditions and Performance Analysis

In this section, the performance of the rate control algorithm proposed above is exhaustively evaluated by using a significant amount of results obtained under relevant conditions. First, the algorithm will be analyzed for single video object (SVO) encoding scenarios; afterwards, for multiple video objects (MVO) encoding scenarios.

6.7.1 Single Video Object Performance Analysis

For the SVO tests, six representative test sequences at 30Hz have been selected (see Figure 6.19): the *Football* (260 frames), *Kayak* (220 frames), and *Stefan* (300 frames) sequences represent the high-motion video sequences, typically more difficult to encode, while the *Foreman* (300 frames), *Mother & Daughter* (300 frames), and *News* (300 frames) sequences represent the low-motion video sequences, typically more easy to encode.



Figure 6.19 – Sample frames for the test sequences used for SVO encoding: a) *Football*; b) *Kayak*; c) *Stefan*; d) *Foreman*; e) *Mother & Daughter*; f) *News*

For comparison purposes, the SVO rate control algorithm proposed in this Thesis will be compared with the VM8 rate control algorithm implemented in the MPEG-4 Visual MoMuSys reference software [32]. This reference selection is motivated by the fact that the algorithm described in the informative Annex on rate control of the MPEG-4 Visual standard [29] is generally the one used for low-delay SVO rate control algorithm comparisons. For the SVO encoding tests, the rate control algorithm proposed in this Thesis is labeled IST, while the MPEG-4 Visual reference software solution [32] is labeled VM8 [108].

The VBV buffer size, B_s , has been set to 0.5 s, i.e., numerically $B_s = R/2$ bits (being R the average target bit rate). For each encoding condition specified in Table 6.2 and Table 6.3, two different conditions in terms of random access points have been tested: 1) one random access

point (I-VOP) every second³⁵, which corresponds to the set of tests with label IP = 1s; 2) one single random access point at the beginning of the sequence (IPPPP...), which corresponds to the set of tests with label IP = 10s (the length of the test sequences used). The bit rate ranges selected attempt to cover a large range of bit rate conditions in terms of encoded bits per pixel for each spatio-temporal resolution. These ranges cover the bit rate test conditions defined in [176] in the context of the MPEG call for proposals on scalable video coding technology. For each condition, the corresponding profile@level used is also presented in the tables.

Table 6.2 – SVO spatio-temporal resolutions and target bit rates for the high-motion test sequences: Football, Kayak, and Stefan

Profile@Level	Luminance Spatial Resolution (Width × Height) ³⁶	Encoded Temporal Resolution ³⁷ [Hz]	Target Encoded Bit Rate Range (Step) [kbit/s]
Simple@L2	176 × 144	7,5	48 – 128 (16)
Simple@L3	176 × 144	15	96 – 256 (32)
Core@L2	352 × 288	15	192 – 512 (64)
Core@L2	352 × 288	30	512 – 1152 (128)

Table 6.3 – SVO spatio-temporal resolutions and target bit rates for the low-motion test sequences: Foreman, Mother & Daughter, and News

Profile@Level	Luminance Spatial Resolution (Width × Height)	Encoded Temporal Resolution [Hz]	Target Encoded Bit Rate Range (Step) [kbit/s]
Simple@L2	176 × 144	7,5	32 – 112 (16)
Simple@L3	176 × 144	15	64 – 224 (32)
Core@L2	352 × 288	15	128 – 448 (64)
Core@L2	352 × 288	30	384 – 1024 (128)

The two rate control algorithms mentioned above will be compared, in this section, based on their relative merits, in meeting typical rate control quality constraints, i.e., in terms of average spatial quality achieved, measured as the average in time of the peak signal to noise ratio (PSNR) for the luminance³⁸ component between the original and the reconstructed VOPs at

³⁵ For the 7,5Hz temporal resolutions, the random access points have been set to 1.2 s.

³⁶ QCIF – 176 × 144; CIF – 352 × 288

³⁷ All original sequences have a temporal resolution of 30Hz.

³⁸ The luminance component has been used since typically the HVS is more sensitive to this component.

the decoder³⁹, and in terms of providing approximately stable quality, measured as the PSNR variation, defined as the ratio between the PSNR standard deviation and the average PSNR (see definition of variation coefficient in [177]). The PSNR variation is an adequate measure to assess the quality variation – average quality trade-off – since it allows comparing two solutions with different average PSNR values in terms of their dispersion relatively to the mean values.

Figure 6.20 to Figure 6.37 present the rate-distortion curves, in the form of average luminance PSNR and PSNR variation curves as a function of the average bit rate, for the sequences in Figure 6.19 encoded in SVO mode with the proposed (IST) and VM8 rate control algorithms. In order to be able to compactly compare the PSNR curves⁴⁰ obtained with the two algorithms, the tool developed by the ITU Video Coding Experts Group (VCEG) in [178] for this purpose is here used. This tool computes the average PSNR difference between two PSNR curves or, alternatively, the bit rate savings between the two curves as follows:

1. Compute a polynomial function of third order that best fits the rate-distortion curve, using four rate-distortion points⁴¹.
2. Compute the integral expression of the two curves in comparison based on the analytical function.
3. Compute the average difference between the two curves as the difference between the two integrals divided by the integration interval.

Notice that this method constitutes a compact way of comparing two rate-distortion curves; however, it does not pretend to substitute the analysis of the actual curves.

HIGH-MOTION VIDEO SEQUENCES

Table 6.4 summarizes the SVO encoding results illustrated in Figure 6.20 – Figure 6.22 for the high-motion *Football* sequence. From these results the following conclusions may be extracted:

- As can be seen in Table 6.4, the IST rate control algorithm achieves an overall PSNR gain of approximately 0.9 dB over all encoding conditions, or equivalently an average decrease of approximately 17% of the bit rate, for the same average quality.
- Figure 6.20a and Figure 6.21a show that the IST algorithm outperforms the VM8 algorithm in terms of achieving a higher average PSNR for all encoding points, for both random access conditions.
- In terms of PSNR variation, the IST algorithm also outperforms VM8, except for some points at QCIF@15Hz, as can be seen in Figure 6.20b and Figure 6.21b.
- For these particular cases, the PSNR variations are very similar for both algorithms, though the IST algorithm achieves consistently an average higher PSNR than VM8 of approximately 0.6 dB, as illustrated in Figure 6.22.

³⁹ Skipped VOPs are replaced by the previous encoded VO VOP at the decoder side.

⁴⁰ Rate-distortion curves in the form of average frame PSNR as a function of the average bit rate.

⁴¹ In this Thesis, from the six rate-distortion points of each curve only the four inner points are used, i.e., the two outermost points are excluded from the calculations.

Table 6.4 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Football sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.80	0.90	-16.90	-18.77
QCIF@15Hz	0.55	0.62	-12.07	-13.47
CIF@15Hz	1.17	1.18	-19.30	-21.62
CIF@30Hz	0.89	0.80	-17.98	-16.28
	0.85	0.88	-16.56	-17.05

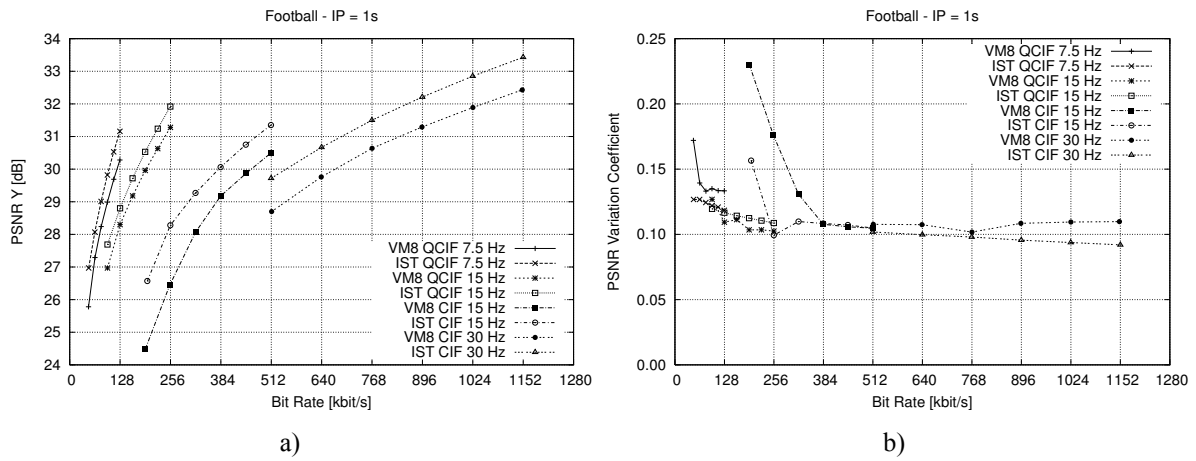


Figure 6.20 – Football SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

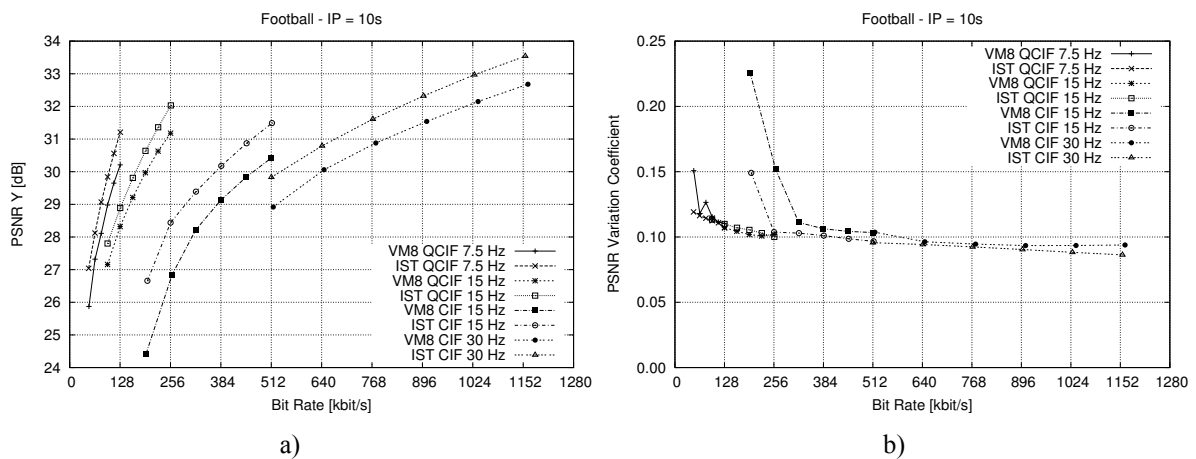


Figure 6.21 – Football SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

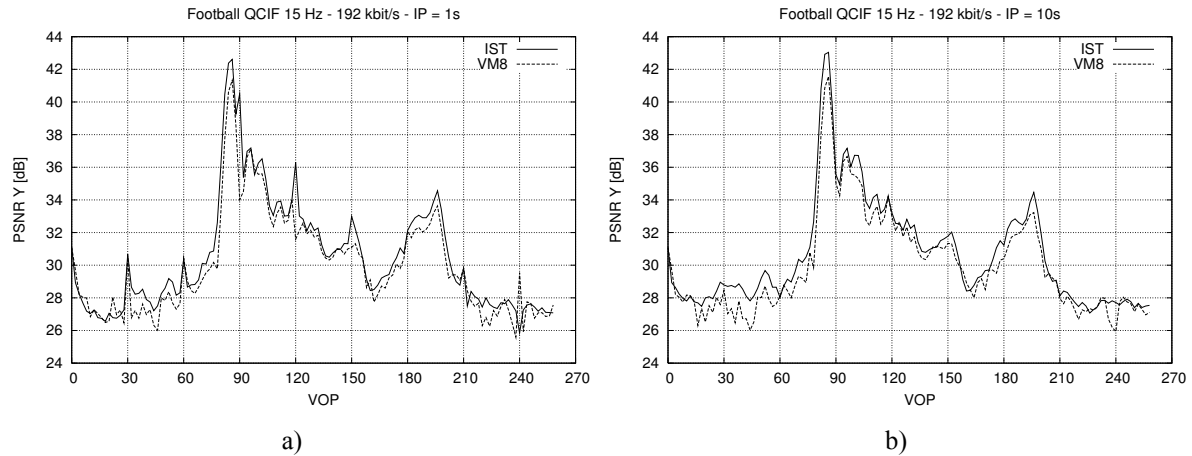


Figure 6.22 – VOP luminance PSNR for the Football sequence encoded at 192 kbit/s:
a) Intra period 1s; b) Intra period 10s

Table 6.5 summarizes the SVO encoding results illustrated in Figure 6.23 – Figure 6.25 for the high-motion *Kayak* sequence. Similarly to the *Football* sequence, the following conclusions may be extracted from the results obtained for the *Kayak* sequence:

- The IST algorithm also consistently outperforms the VM8 solution, both in terms of average PSNR and PSNR variation, as illustrated in Figure 6.23 and Figure 6.24.
- In terms of overall gain, the IST algorithm leads to an average PSNR gain of approximately 1 dB or, equivalently, to an average decrease of approximately 17% of the bit rate, as can be seen in Table 6.5.
- The rate-distortion points corresponding to high PSNR variations (see Figure 6.23b and Figure 6.24b) correspond to situations where the bit rate is scarce and thus the rate control algorithm needs to perform a stringent bit allocation or eventually skip some VOPs. In the case of the VM8 algorithm, skipping VOs does not occur only in scarce bit rate situations but also when the bit rate is high because the algorithm does not control very tightly the bit allocations leading to possible violations of the VBV. Figure 6.25 illustrates this situation for the *Kayak* sequence encoded at 1024 kbit/s.

Table 6.5 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the *Kayak* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.72	0.81	-13.82	-16.06
QCIF@15Hz	0.43	0.42	-8.47	-8.45
CIF@15Hz	1.58	1.53	-25.96	-25.44
CIF@30Hz	1.05	0.78	-22.81	-17.26
	0.95	0.89	-17.77	-17.28

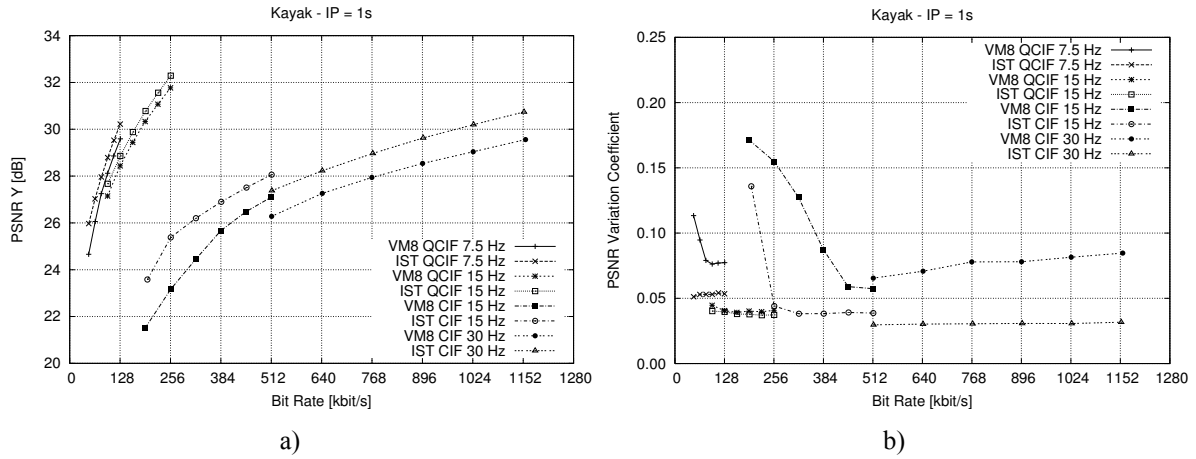


Figure 6.23 – Kayak SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

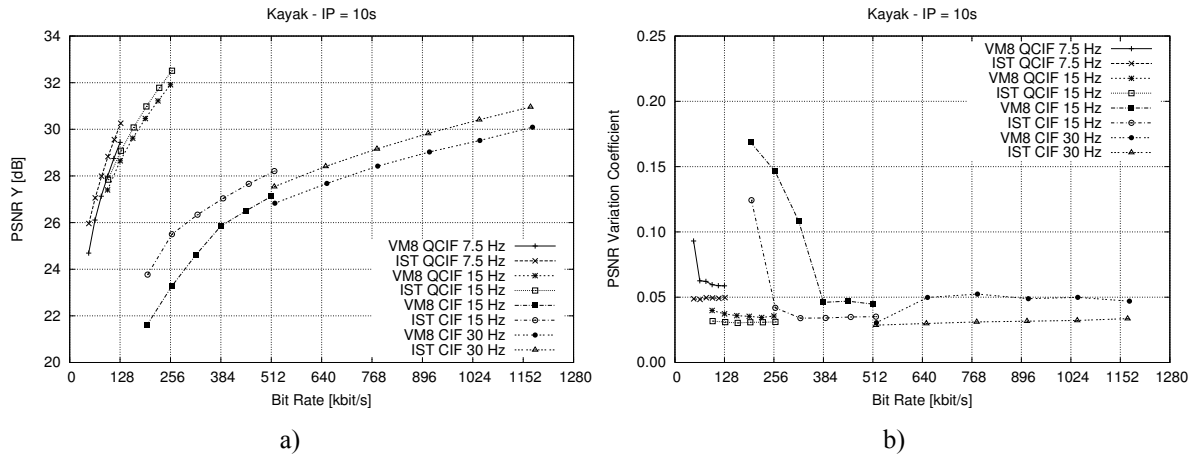


Figure 6.24 – Kayak SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

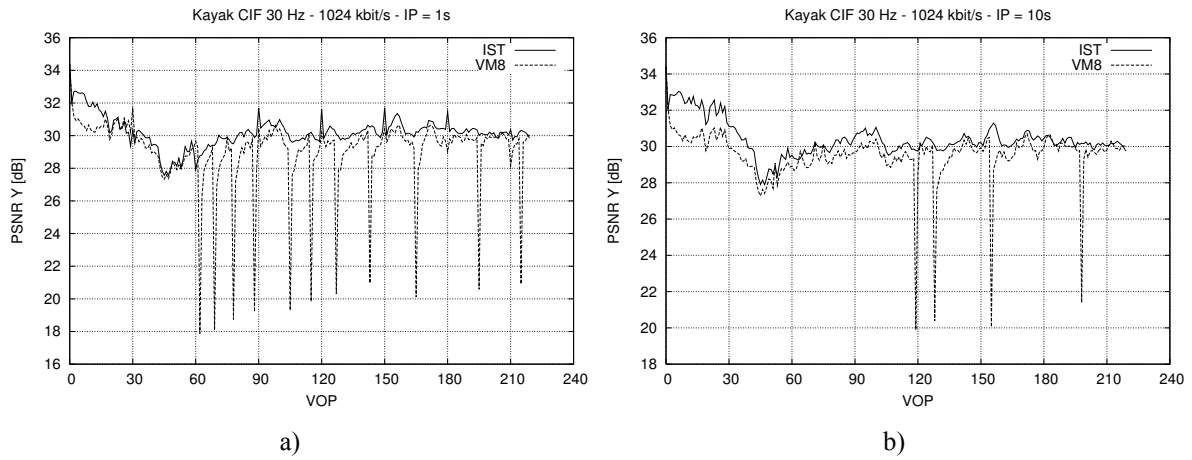


Figure 6.25 – VOP luminance PSNR for the Kayak sequence encoded at 1024 kbit/s:
a) Intra period 1s; b) Intra period 10s

Table 6.6 summarizes the SVO encoding results illustrated in Figure 6.26 – Figure 6.28 for the high-motion *Stefan* sequence. From these results the following conclusions may be extracted:

- The IST algorithm consistently outperforms the VM8 algorithm showing higher average PSNR and lower PSNR variation, for all encoding conditions.
- As shown in Table 6.6, the IST algorithm exhibits an average PSNR gain of approximately 0,5 dB for both random access conditions, which correspond in terms of bit rate savings to approximately 8%.
- The IST algorithm can efficiently allocate the available bit rate in order to produce smoother quality variations immediately after I-VOPs, provided that the bit rate is not too scarce, as illustrated in Figure 6.28a and Figure 6.28b.

Table 6.6 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the *Stefan* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.28	0.40	-5.50	-8.37
QCIF@15Hz	0.14	0.17	-2.72	-3.40
CIF@15Hz	0.89	0.81	-13.49	-12.26
CIF@30Hz	0.60	0.31	-10.60	-6.00
	0.48	0.42	-8.08	-7.51

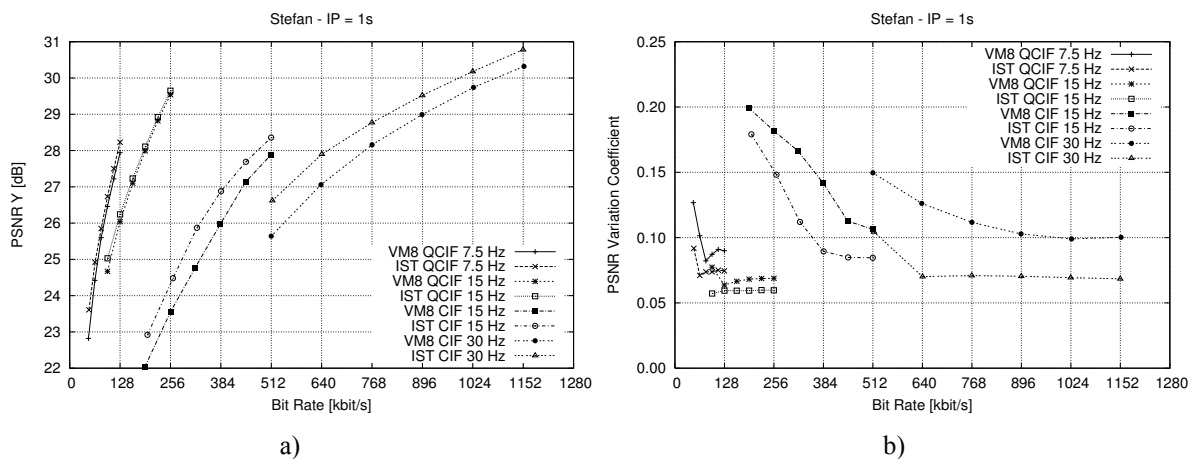


Figure 6.26 – Stefan SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

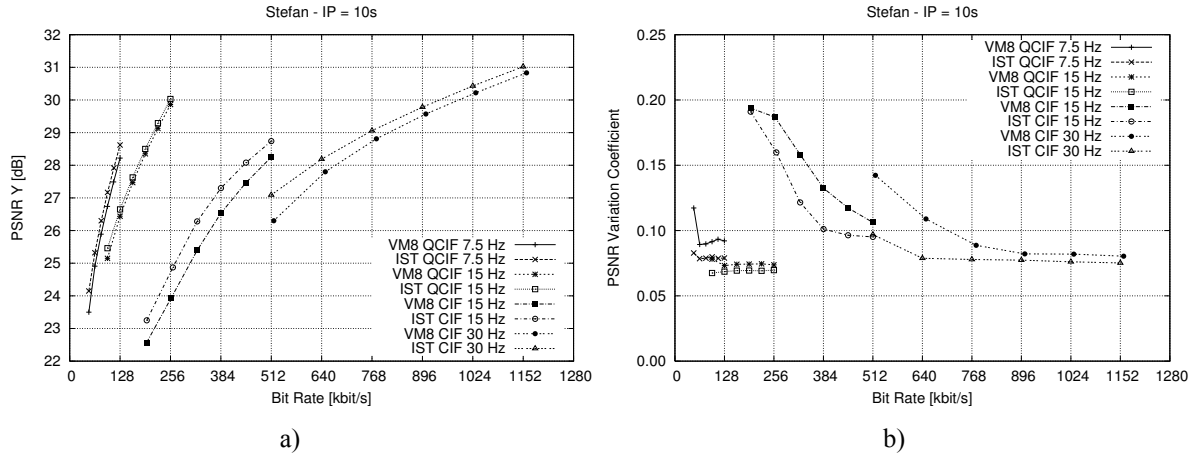


Figure 6.27 – Stefan SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

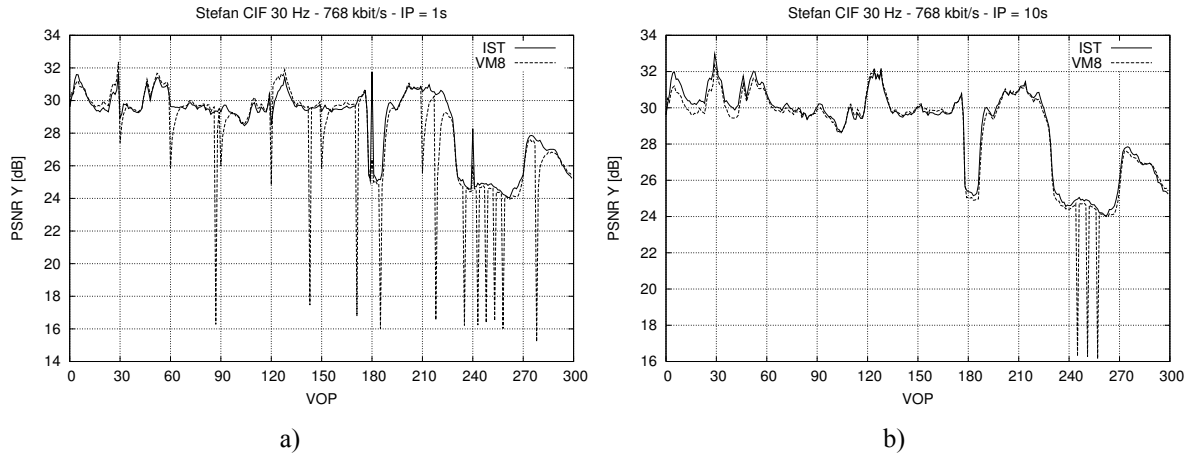


Figure 6.28 – VOP luminance PSNR for the Stefan sequence encoded at 768 kbit/s: a) Intra period 1s; b) Intra period 10s

LOW-MOTION VIDEO SEQUENCES

Table 6.7 summarizes the SVO encoding results illustrated in Figure 6.29 – Figure 6.31 for the low-motion *Foreman* sequence. From these results the following conclusions may be extracted:

- As can be seen in Table 6.7, the IST rate control algorithm achieves an overall PSNR gain of approximately 0.4 dB over all encoding conditions or, equivalently, an average decrease of approximately 8% of the bit rate, for the same average quality.
- The highest gain is 0,6 dB for CIF@30Hz with an Intra period of 1s and the lowest gain is 0,1 dB for QCIF@15Hz with an Intra period of 10 s.
- In terms of PSNR variation, the IST algorithm also outperforms VM8, as can be seen in Figure 6.29b and Figure 6.30b. Notice that in these figures, the high PSNR variation points correspond to situations where the bit rate is scarce; there, the VM8 algorithm skips some VOPs at the scene change (approximately around VOP 200), as illustrated in Figure 6.31, while the IST algorithm can encode these VOPs without violating the VBV and without penalizing dramatically the spatial quality of the encoded video.

This is due to the usage of an accurate VBV control and the definition of precise target VBV buffer occupancies for each encoding time instant.

Table 6.7 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the Foreman sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.35	0.37	-6.36	-7.31
QCIF@15Hz	0.18	0.09	-3.42	-1.83
CIF@15Hz	0.54	0.53	-10.72	-11.17
CIF@30Hz	0.57	0.28	-12.95	-6.88
	0.41	0.32	-8.36	-6.80

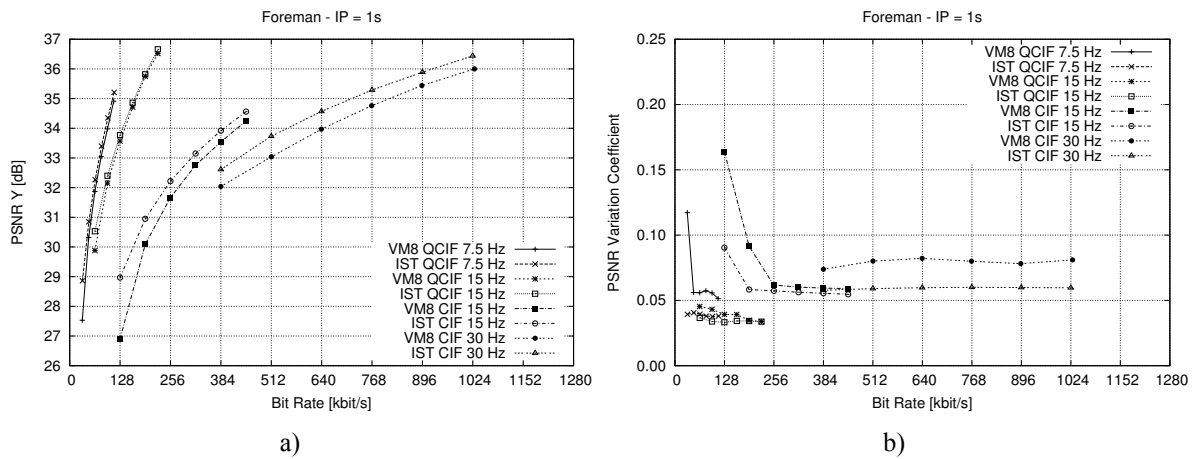


Figure 6.29 – Foreman SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

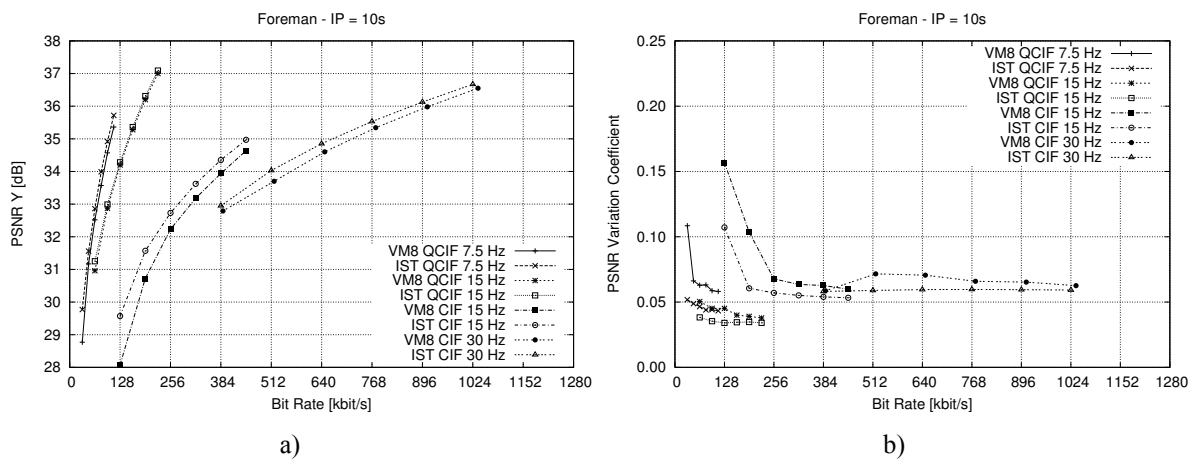


Figure 6.30 – Foreman SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

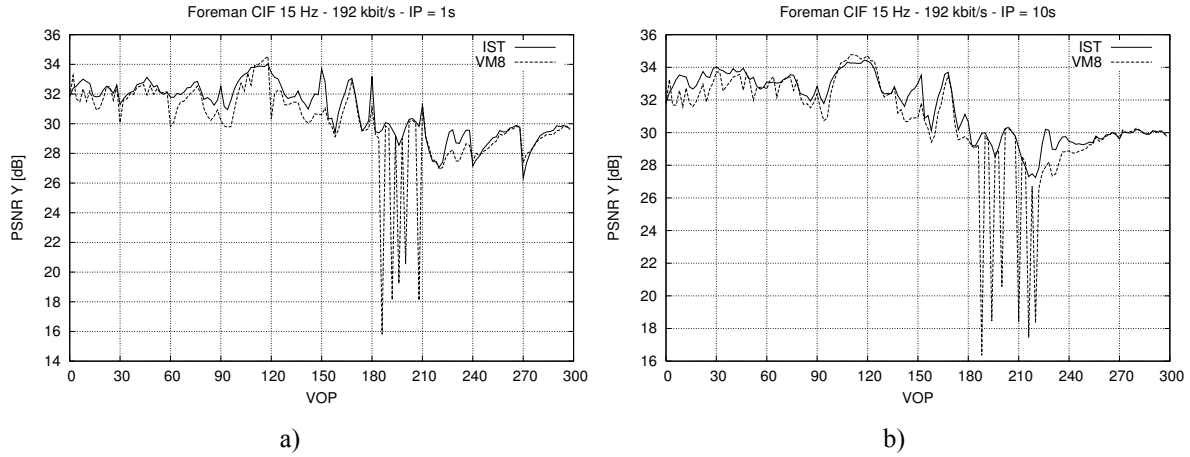


Figure 6.31 – VOP luminance PSNR for the Foreman sequence encoded at 192 kbit/s:
a) Intra period 1s; b) Intra period 10s

Table 6.8 summarizes the SVO encoding results illustrated in Figure 6.32 – Figure 6.34 for the low-motion *Mother & Daughter* sequence. From these results the following conclusions may be extracted:

- The IST algorithm achieves consistently a higher PSNR for all encoding conditions, notably for the higher bit rates of each spatio-temporal resolution, as can be seen in Figure 6.32a and Figure 6.33a.
- Regarding the PSNR variation, as can be seen in Figure 6.32b and Figure 6.33b, for an Intra period of 1s, the IST algorithm also outperforms VM8, except for some points of QCIF@15Hz, while for an Intra period of 10 s, the results of both algorithms are very similar. In this case, the PSNR variation coefficient is also very low due to the high PSNR achieved for most encoding points.
- For an Intra period of 1s, the VM8 algorithm leads typically to significant drops in PSNR after each I-VOP (see Figure 6.34), which is reflected as an overall average PSNR drop of approximately 0,7 dB relatively to the IST algorithm (see Table 6.8).

Table 6.8 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the *Mother & Daughter* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.65	0.51	-11.75	-10.29
QCIF@15Hz	0.76	0.57	-13.67	-12.14
CIF@15Hz	0.66	0.40	-15.27	-10.97
CIF@30Hz	0.86	0.58	-20.32	-17.51
	0.73	0.52	-15.25	-12.73

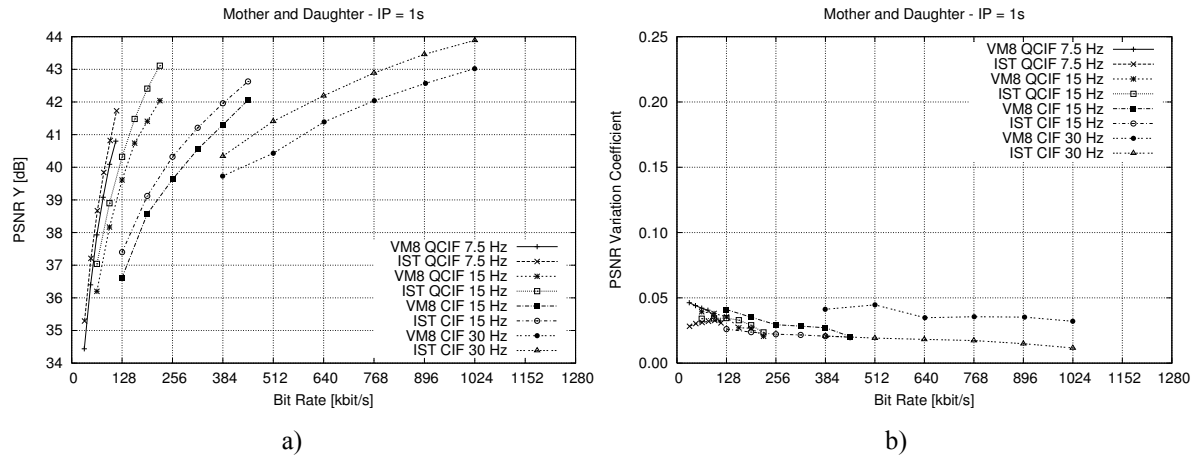


Figure 6.32 – Mother & Daughter SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

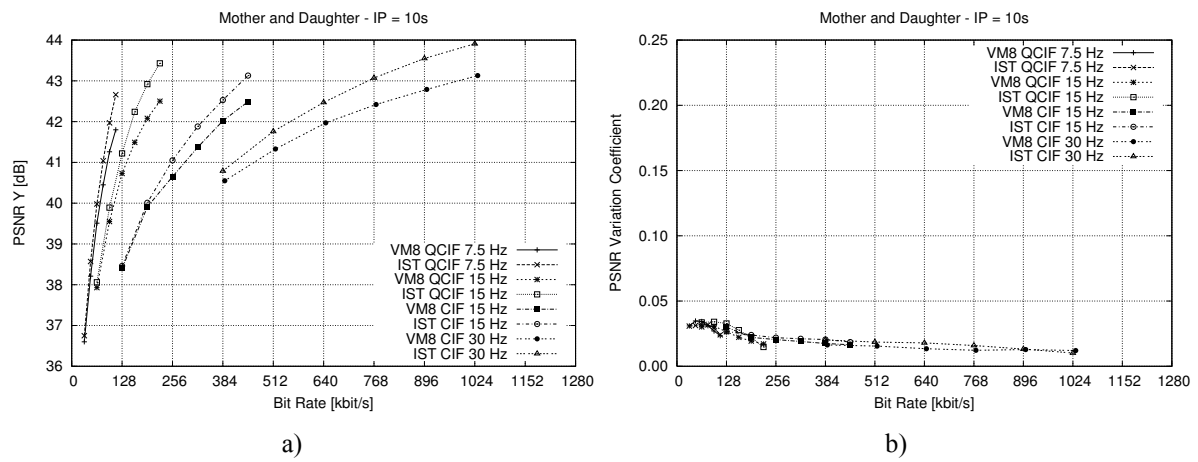


Figure 6.33 – Mother & Daughter SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

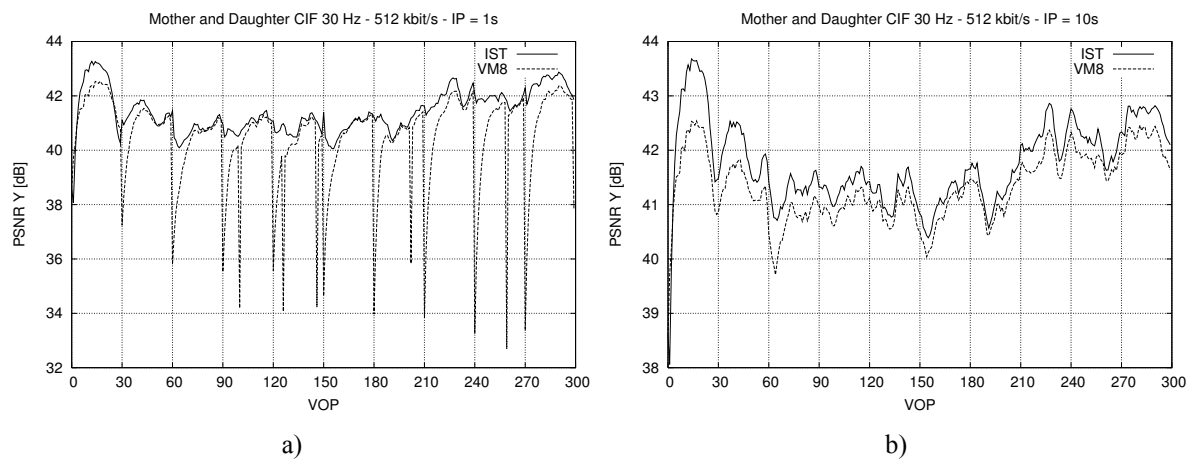


Figure 6.34 – VOP luminance PSNR for the Mother & Daughter sequence encoded at 512 kbit/s: a) Intra period 1s; b) Intra period 10s

Table 6.9 summarizes the SVO encoding results illustrated in Figure 6.35 – Figure 6.37 for the low-motion *News* sequence. From these results the following conclusions may be extracted:

- The IST algorithm achieves a higher PSNR for most encoding conditions, being the exceptions the QCIF@7.5Hz 32 kbit/s, QCIF@15Hz 64 kbit/s, and CIF@15Hz 128 kbit/s cases, for an Intra period of 10 s (see Figure 6.35a and Figure 6.36a).
- The exception points correspond to encoding conditions where the bit rate is scarce and consequently the IST algorithm attempts to achieve a better trade-off by penalizing the easy to code MBs, e.g., those in the background, to favor the more difficult to code MBs, e.g., the dancers and the speakers (see Figure 6.19).
- The VM8 makes a less restrictive encoding, allocating more bits to the easy to code background and few bits to the foreground (dancers and speakers MBs), achieving a slight higher PSNR at the expense of a higher MB quality variation.
- Nevertheless, on the overall encoding conditions, the IST algorithm has an average gain of 1,2 dB for an Intra period of 1s and an average gain of 0,4 dB for an Intra period of 10 s or, equivalently, bit rate savings of approximately 17% and 8%, respectively (see Table 6.9)
- As for the *Mother & Daughter* sequence, for the *News* sequence, the VM8 also typically leads to significant drops in PSNR after each I-VOP (see Figure 6.37a).
- As can be seen in Figure 6.37b, for VOPs 90, 150, and 240, for an Intra period of 10 s, there are also significant drops in terms of VOP PSNR, which are due to the sudden changes in the foreground (dancers MBs). Nevertheless, the IST algorithm exhibits a much smaller decay in the PSNR when compared to the VM8 algorithm.

Table 6.9 – SVO average PSNR and bit rate gains of the proposed rate control algorithm for the *News* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	0.95	0.29	-12.32	-4.40
QCIF@15Hz	1.12	0.20	-13.66	-3.06
CIF@15Hz	1.39	0.77	-21.05	-14.18
CIF@30Hz	1.48	0.42	-22.66	-8.44
	1.24	0.42	-17.42	-7.52

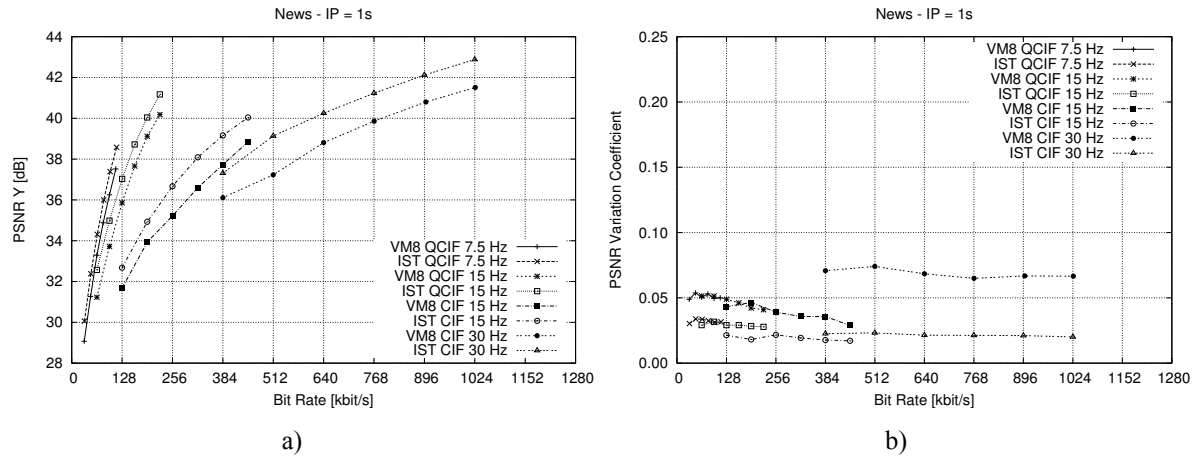


Figure 6.35 – News SVO (Intra period 1s): a) Average PSNR; b) PSNR Variation

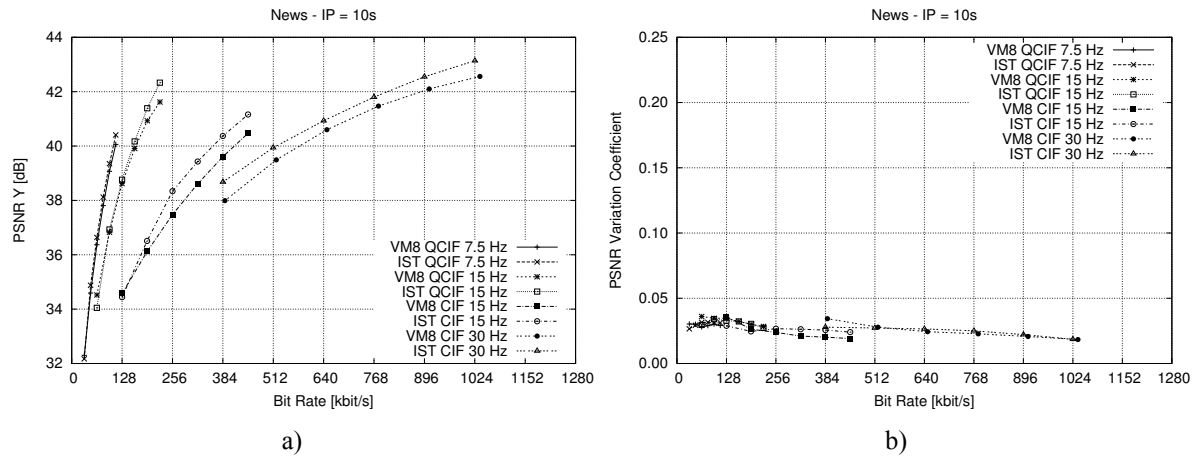


Figure 6.36 – News SVO (Intra period 10s): a) Average PSNR; b) PSNR Variation

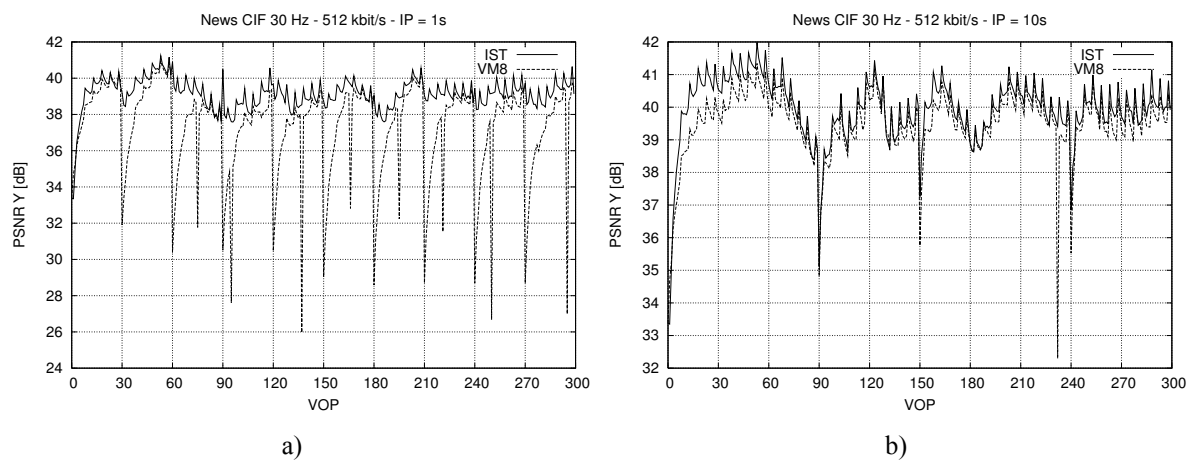


Figure 6.37 – VOP luminance PSNR for the News sequence encoded at 512 kbit/s:
a) Intra period 1s; b) Intra period 10s

SUMMARY OF SVO PERFORMANCE ANALYSIS

In order to better illustrate the relative behavior of the two rate control algorithms for the various encoding conditions, Table 6.10 to Table 6.13 summarize the PSNR and bit rate gains of the proposed SVO algorithm for the different spatio-temporal encoding conditions, notably: QCIF@7.5Hz, QCIF@15Hz, CIF@15Hz, and CIF@30Hz. From these tables, the following conclusions may be derived:

- The IST algorithm shows the biggest gains for CIF@15Hz for a random access point at each second with approximately 1 dB more in terms of PSNR and 17% less in terms of bit rate
- The smallest gain of the IST algorithm is achieved for QCIF@15Hz for a single random access point at the beginning of the sequence with approximately 0,4 dB more in terms of PSNR and 8% less in terms of bit rate.
- Typically, the IST algorithm exhibits higher gains for the high-motion sequences and for the shorter random access conditions, i.e., for the more difficult encoding conditions.

Table 6.10 – SVO average PSNR and bit rate gains for QCIF@7.5Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Football	0.80	0.90	-16.90	-18.77
Kayak	0.72	0.81	-13.82	-16.06
Stefan	0.28	0.40	-5.50	-8.37
Foreman	0.35	0.37	-6.36	-7.31
Mother & Daughter	0.65	0.51	-11.75	-10.29
News	0.95	0.29	-12.32	-4.40
	0.63	0.55	-11.11	-10.87

Table 6.11 – SVO average PSNR and bit rate gains for QCIF@15Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Football	0.55	0.62	-12.07	-13.47
Kayak	0.43	0.42	-8.47	-8.45
Stefan	0.14	0.17	-2.72	-3.40
Foreman	0.18	0.09	-3.42	-1.83
Mother & Daughter	0.76	0.57	-13.67	-12.14
News	1.12	0.20	-13.66	-3.06
	0.53	0.35	-9.00	-7.06

Table 6.12 – SVO average PSNR and bit rate gains for CIF@15Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Football	1.17	1.18	-19.30	-21.62
Kayak	1.58	1.53	-25.96	-25.44
Stefan	0.89	0.81	-13.49	-12.26
Foreman	0.54	0.53	-10.72	-11.17
Mother & Daughter	0.66	0.40	-15.27	-10.97
News	1.39	0.77	-21.05	-14.18
	1.04	0.87	-17.63	-15.94

Table 6.13 – SVO average PSNR and bit rate gains for CIF@30Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Football	0.89	0.80	-17.98	-16.28
Kayak	1.05	0.78	-22.81	-17.26
Stefan	0.60	0.31	-10.60	-6.00
Foreman	0.57	0.28	-12.95	-6.88
Mother & Daughter	0.86	0.58	-20.32	-17.51
News	1.48	0.42	-22.66	-8.44
	0.91	0.53	-17.89	-12.06

6.7.2 Multiple Video Objects Performance Analysis

For the MVO tests, four representative test sequences at 30Hz and with 300 frames have been selected: *Stefan* and *Bream* with 2 VOs; *Coastguard* and *News* with 4 VOs. Similarly to the SVO tests, these sequences can be grouped according to their motion activity into: 1) high-motion video sequences (*Stefan* and *Coastguard*) and 2) low-motion video sequences (*Bream* and *News*) – see Figure 6.38 to Figure 6.41.

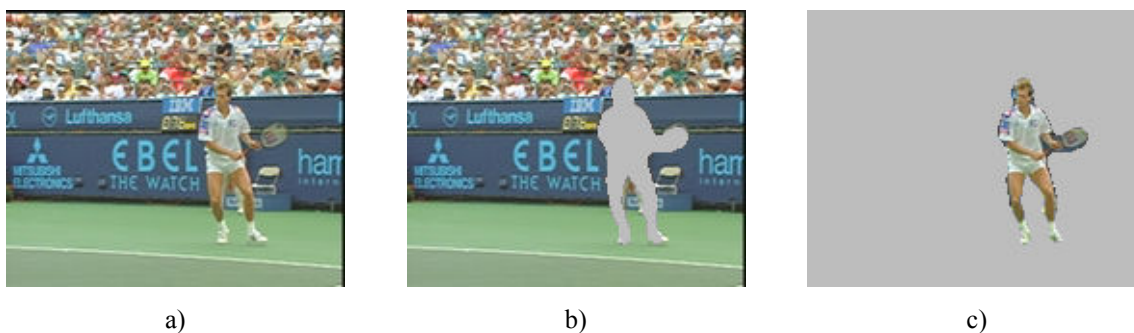


Figure 6.38 – MVO Stefan sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Player)

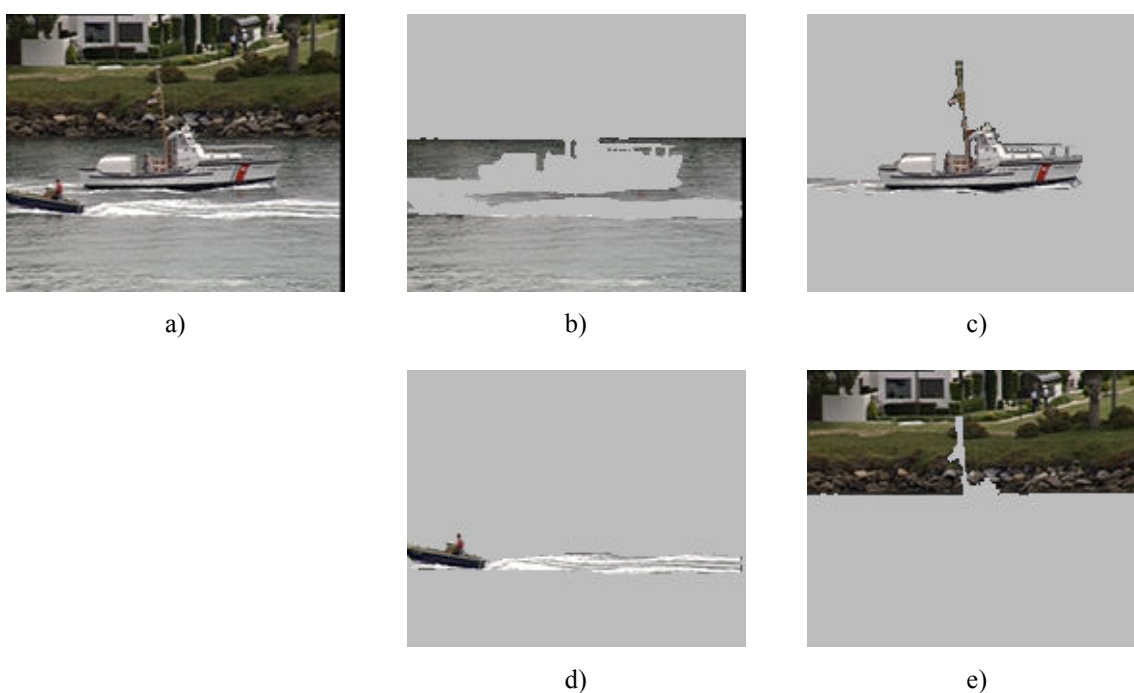


Figure 6.39 – MVO Coastguard sequence (frame 100): a) Composed Scene; b) VO 0 (Water); c) VO 1 (Large Boat); d) VO 2 (Small Boat); e) VO 3 (Shore)

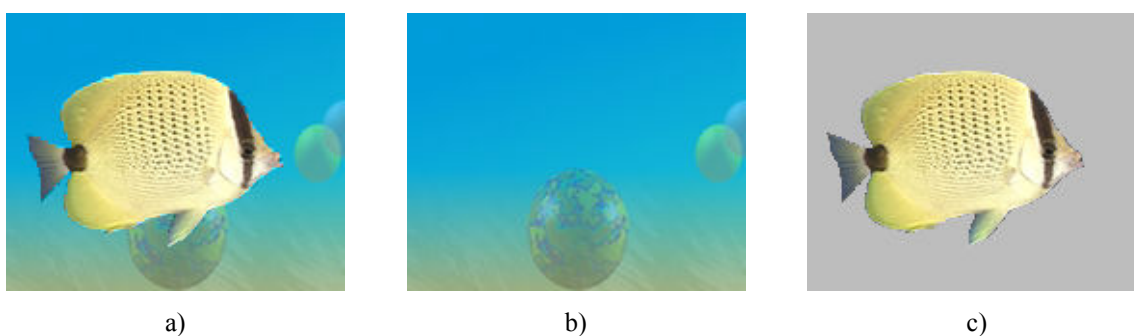


Figure 6.40 – MVO Bream sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Fish)

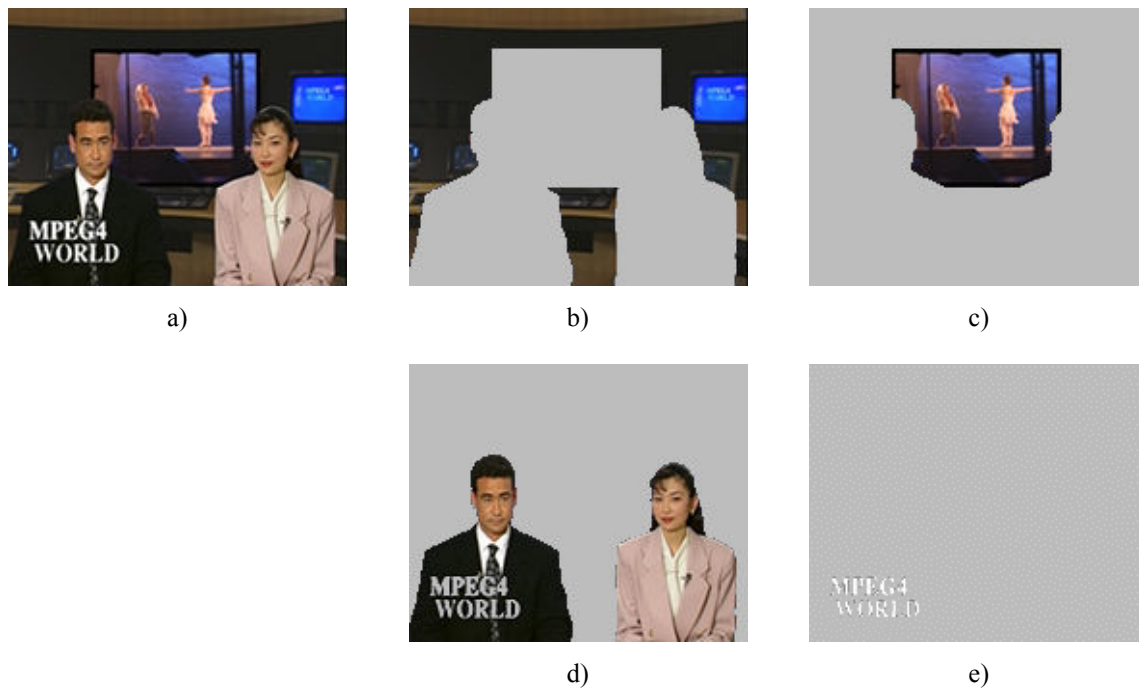


Figure 6.41 – MVO News sequence (frame 0): a) Composed Scene; b) VO 0 (Background); c) VO 1 (Dancers); d) VO 2 (Speakers); e) VO 3 (Logo)

For comparison purposes, the MVO rate control algorithm proposed in this Thesis will be compared with the VM5 rate control algorithm implemented in the MPEG-4 Visual MoMuSys reference software[32]. This reference selection is motivated by the fact that the rate algorithm described in the informative Annex L of the MPEG-4 Visual standard [29], is generally used for low-delay MVO rate control algorithm comparisons. For the MVO encoding tests, the algorithm proposed in this Thesis is labeled IST, while the MPEG-4 Visual reference software solution [32] is labeled VM5⁴² [102].

As with the SVO tests, the VBV buffer size as been numerically set to $B_s = R/2$. Table 6.14 and Table 6.15 specify the spatio-temporal encoding resolutions and target bit rates for the different test sequences. As with the SVO tests, for each encoding condition specified in Table 6.14 and Table 6.15, two different conditions in terms of random access have been tested: 1) one random access point (I-VOP) every second⁴³, which corresponds to the set of tests with label IP = 1s; 2) one single random access point at the beginning of the sequence (IPPPP...), which corresponds to the set of tests with label IP = 10s. The bit rate conditions remain the same as for the SVO tests for each class of sequences. For each condition, the selected profile@level used is also presented in the tables.

⁴² Notice that the VM8 algorithm supports neither arbitrary shaped VOs nor MVO encoding.

⁴³ For the 7,5Hz temporal resolutions, the random access points have been set to 1.2 s.

Table 6.14 – MVO spatio-temporal resolutions and target bit rates for the high-motion test sequences: Stefan and Coastguard

Profile@Level ⁴⁴	Luminance Spatial Resolution (Width × Height)	Encoded Temporal Resolution [Hz]	Target Encoded Bit Rate Range (Step) [kbit/s]
Main@L2	176 × 144	7,5	48 – 128 (16)
Main@L2	176 × 144	15	96 – 256 (32)
Main@L2	352 × 288	15	192 – 512 (64)
Main@L2	352 × 288	30	512 – 1152 (128)

Table 6.15 – MVO spatio-temporal resolutions and target bit rates for the low-motion test sequences: Bream and News

Profile@Level	Luminance Spatial Resolution (Width × Height)	Encoded Temporal Resolution [Hz]	Target Encoded Bit Rate Range (Step) [kbit/s]
Main@L2	176 × 144	7,5	32 – 112 (16)
Main@L2	176 × 144	15	64 – 224 (32)
Main@L2	352 × 288	15	128 – 448 (64)
Main@L2	352 × 288	30	384 – 1024 (128)

The two rate control algorithms will be compared, in this section, based on their relative merits, in meeting typical rate control quality constraints in MVO encoding scenarios, i.e., in terms of average scene spatial quality achieved, measured as the Average Scene PSNR for the luminance component between the original and the reconstructed video frames at the decoder⁴⁵, and in terms of providing approximately stable quality between the various VOs in the scene, measured as the PSNR variation, defined as follows.

For each encoding time instant, the Mean Square Error for the Scene Plane, MSE_{SP} , for the luminance component is computed as

$$MSE_{SP} = \frac{1}{SIZE_{SP}} \sum_{n=1}^{N_{VO}} MSE_{VO}[n] \times SIZE_{VO}[n] \quad (6.127)$$

where

$$SIZE_{SP} = \sum_{n=1}^{N_{VO}} SIZE_{VO}[n] \quad (6.128)$$

⁴⁴ Since the VM5 algorithm does not implement VMV and VCV verifications and due to the problems referred in section 4.6.2, in order to compare the two competing algorithms under fair conditions, a more powerful profile@level has been selected than what would be natural for this type of content.

⁴⁵ Skipped VOPs are replaced by the previous encoded VO VOP at the decoder side.

From (6.127), the PSNR for the Scene Plane, $PSNR_{SP}$, is computed as

$$PSNR_{SP} = 10 \log_{10} \left(\frac{255^2}{MSE_{SP}} \right) \quad (6.129)$$

Based on the Scene Plane PSNR (6.129) and each VO PSNR, (6.131) the Scene PSNR Difference, $PSNRD_{SP}$, is computed as

$$PSNRD_{SP} = \frac{1}{SIZE_{SP}} \sum_{n=1}^{N_{VO}} |PSNR_{SP} - PSNR_{VO}[n]| \times SIZE_{VO}[n] \quad (6.130)$$

where

$$PSNR_{VO} = 10 \log_{10} \left(\frac{255^2}{MSE_{VO}} \right) \quad (6.131)$$

Notice that the Scene PSNR Difference expresses the average absolute deviation between the weighted Average Scene PSNR and the PSNR of each VO, i.e., it is a measure related to the spatial quality smoothness inside each SP.

Averaging (6.130) over all GOS of the sequence leads to the Average Scene PSNR Difference, \overline{PSNRD}_S , computed as

$$\overline{PSNRD}_S = \sum_{m=1}^{N_{GOS}} \sum_{p=1}^{N_{SP}[m]} PSNRD_{SP}[m][p] \quad (6.132)$$

Similarly, the Average Scene PSNR, \overline{PSNR}_S , is obtained through

$$\overline{PSNR}_S = \sum_{m=1}^{N_{GOS}} \sum_{p=1}^{N_{SP}[m]} PSNR_S[m][p] \quad (6.133)$$

Finally, the Scene PSNR Variation, \overline{PSNRV}_S , is the ratio between (6.132) and (6.133), i.e.,

$$\overline{PSNRV}_S = \frac{\overline{PSNRD}_S}{\overline{PSNR}_S} \quad (6.134)$$

Figure 6.38 to Figure 6.57 present the rate-distortion curves, in the form of average luminance PSNR and PSNR variation curves as a function of average bit rate, for the sequences in Figure 6.40 to Figure 6.41, encoded in MVO mode with the IST and VM5 algorithms.

HIGH-MOTION VIDEO SEQUENCES

Table 6.16 summarizes the MVO encoding results illustrated in Figure 6.42 – Figure 6.45 for the high-motion *Stefan* sequence (2 VOs). From these results the following conclusions may be extracted:

- The IST MVO algorithm outperforms the VM5 MVO algorithm for all encoding conditions (see Figure 6.42a and Figure 6.43a) showing an average PSNR gain of around 2 dB for both random access period conditions or, equivalently, a gain in terms of bit rate between 31% and 47% (see Table 6.16).
- Both algorithms exhibit a very similar Scene PSNR Variation indicating also that the quality of both VOs is very similar (see Figure 6.42b and Figure 6.43b). For all encoding conditions the Scene PSNR Variation, for both the IST and VM5 algorithms,

is below 5%.

- The IST algorithm, however, besides achieving a higher Average Scene PSNR, also achieves a lower Scene PSNR Variation, indicating a superior performance in terms of bit allocation and rate control.

Figure 6.44 and Figure 6.45 illustrate for each spatio-temporal and random access condition the behavior of both algorithms in terms of Scene PSNR. The following conclusions can be derived from these figures:

- For one random access point at every second, the VM5 algorithm exhibits a poorer bit allocation between I- and P-VOPs, resulting in I-VOPs with worthless high quality, followed by a set of skipped encoding time instants resulting in severe drops of PSNR, which can be seen by the low PSNR peaks in Figure 6.44a – Figure 6.44d.
- Even for higher bit rates, such as in Figure 6.44d, VM5 tends to skip several encoding time instants during some parts of the sequence, as between VOPs 240 and 270. On the contrary, the IST MVO algorithm not only can keep the scene quality higher during most of time but it can also avoid skipping VOPs during more demanding time instants, due to a more efficient bit allocation and VBV control.
- For the single access point at the beginning of each sequence condition, the IST MVO algorithm also exhibits a higher Average Scene PSNR and less skipped time instants. As can be seen in Figure 6.45a to Figure 6.45d, the IST MVO algorithm achieves a higher Scene PSNR most of the time, avoids severe drops in Scene PSNR during the more demanding parts of the sequence, and finally, skips less encoding time instants.

Table 6.16 – MVO average PSNR and bit rate gains of the proposed rate control algorithm for the Stefan sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	2.56	1.85	-46.68	-36.79
QCIF@15Hz	1.94	1.89	-37.38	-35.83
CIF@15Hz	2.57	2.15	-38.77	-37.66
CIF@30Hz	1.84	2.09	-30.95	-35.46
	2.23	2.00	-38.45	-36.44

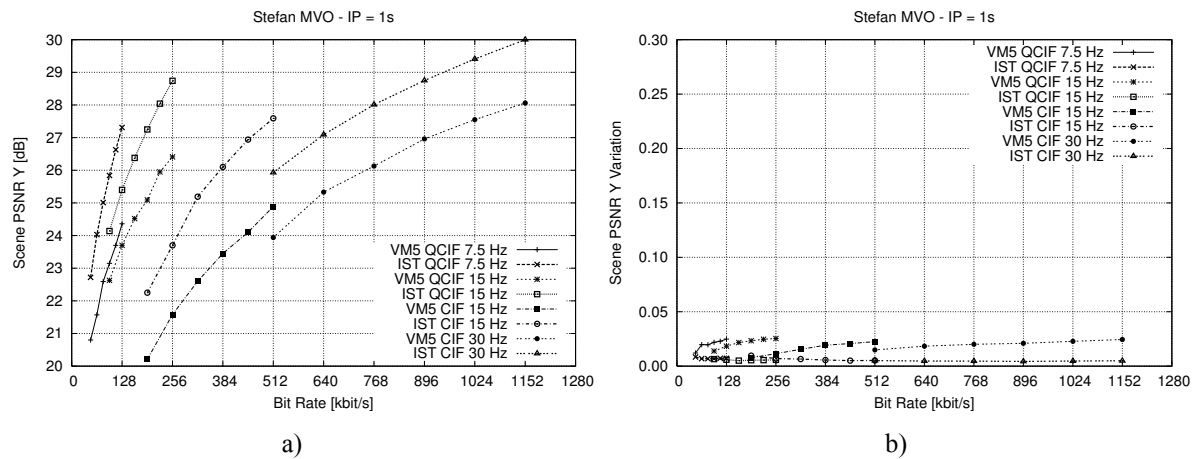


Figure 6.42 – Stefan MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation

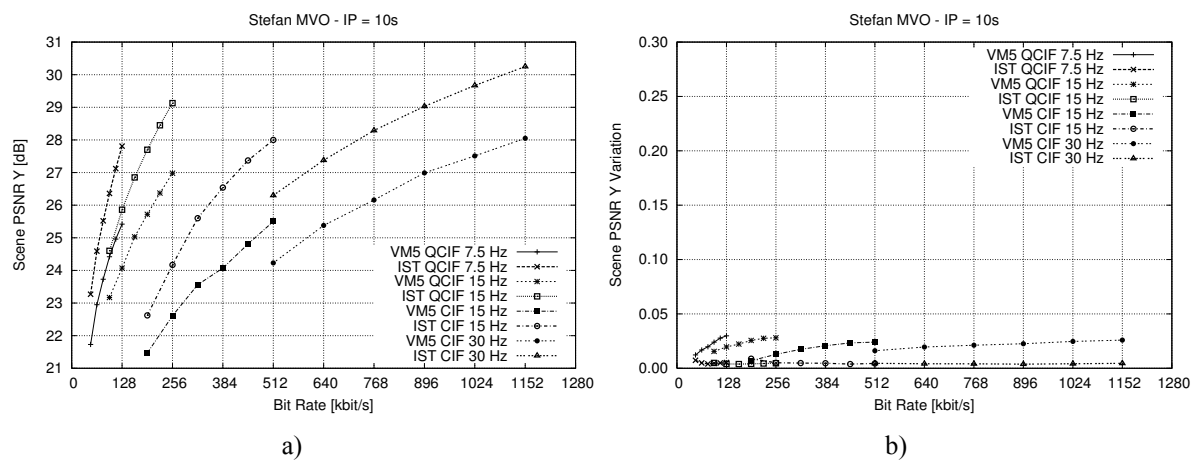
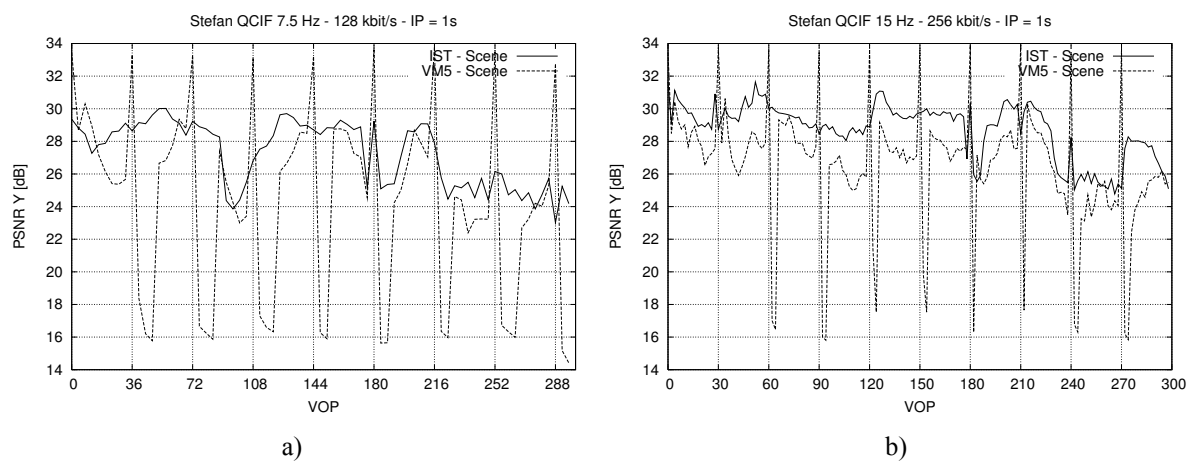


Figure 6.43 – Stefan MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation



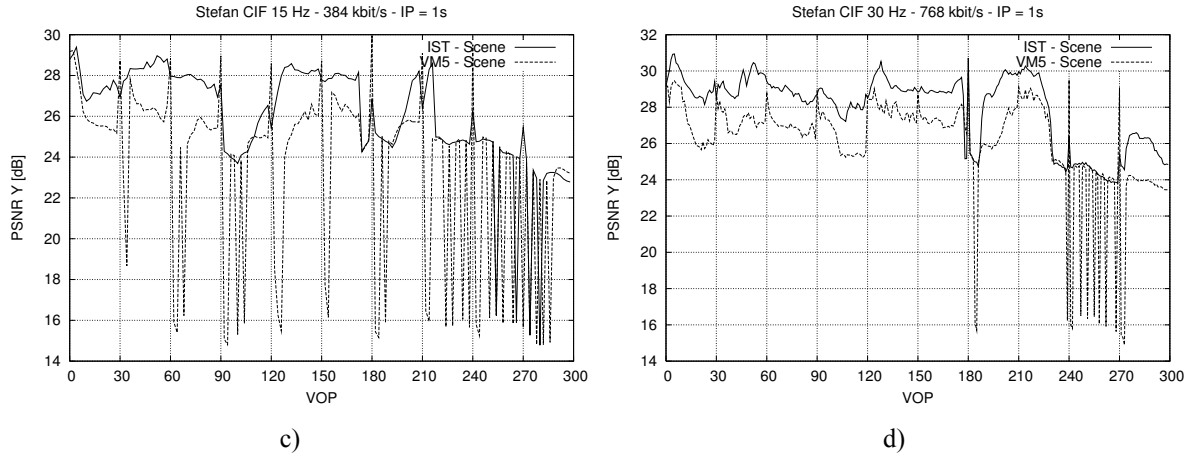


Figure 6.44 – Stefan MVO Scene PSNR (Intra period 1s): a) QCIF@7.5Hz 128 kbit/s; b) QCIF@15Hz 256 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@30Hz 768 kbit/s

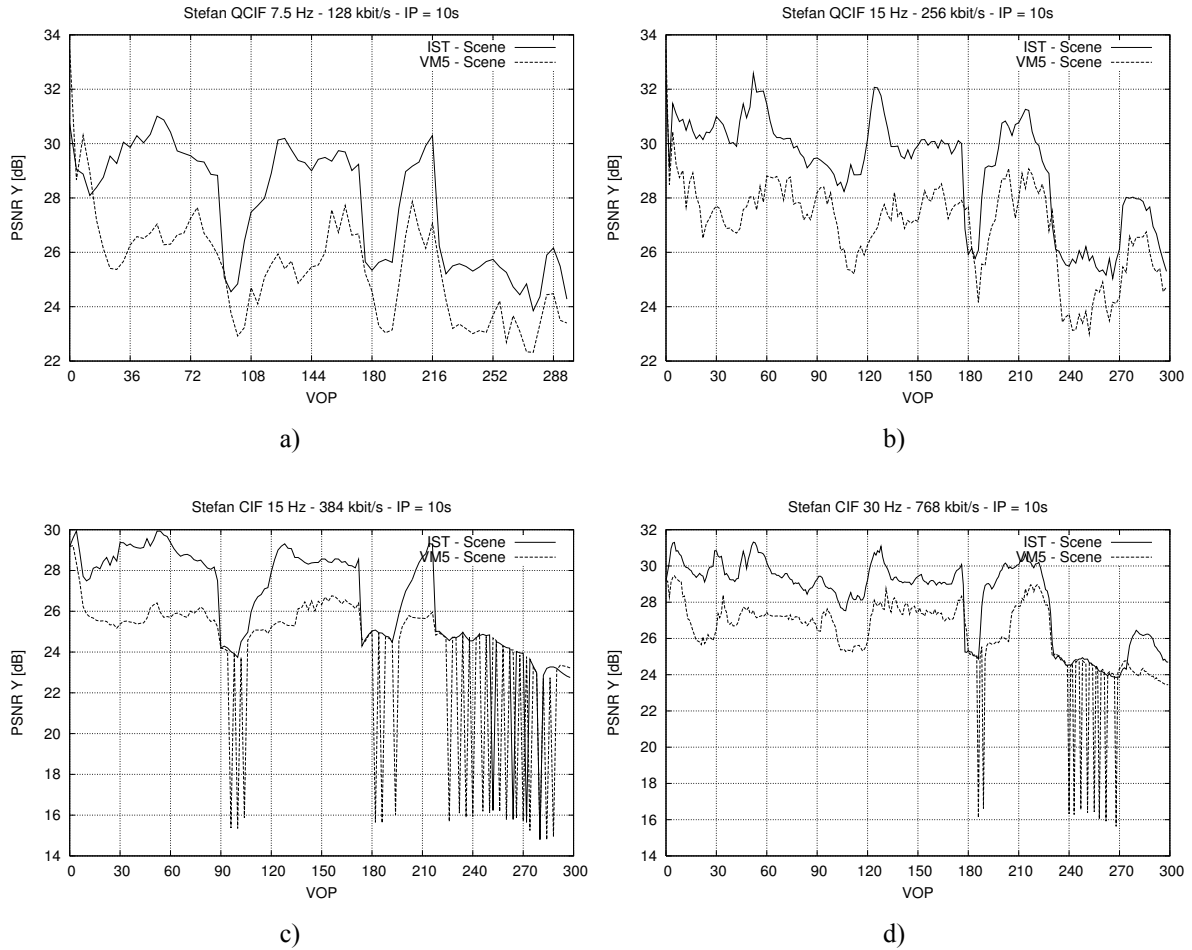


Figure 6.45 – Stefan MVO Scene PSNR (Intra period 10s): a) QCIF@7.5Hz 128 kbit/s; b) QCIF@15Hz 256 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@30Hz 768 kbit/s

Table 6.17 summarizes the MVO encoding results illustrated in Figure 6.46 – Figure 6.49 for the high-motion *Coastguard* sequence (4 VOs). From these results the following conclusions may be extracted:

- The IST MVO algorithm achieves an average PSNR gain of over 1 dB, for a random access point every 1s, except for CIF@30Hz where the gain is marginal.
- The Scene PSNR Variation achieved by the IST MVO algorithm is very similar to the VM5 MVO algorithm, being lower for most the encoding points (see Figure 6.46b).
- The IST MVO algorithm can typically achieve a higher quality for I-VOPs while maintaining the temporal resolution, i.e., without skipping VOPs, while the VM5 MVO algorithm tends to skip a few VOPs after Intra coded VOPs. This situation results, typically, in severe drops in the Scene PSNR (Figure 6.48 illustrates this situation for QCIF@7.5Hz and CIF@15Hz).
- One of the main reasons for such behavior is the accurate VBV control of the IST MVO algorithm proposed in Section 6.4.6, which allows the encoder to use more efficiently the available space in the buffer by predicting in advance its occupancy along the GOS and allowing the encoder rate buffer to be almost full after each I-VOP encoding. On the other hand, the VM5 MVO algorithm, by targeting always a buffer occupancy of $B_s/2$, needs to perform skipping after the I-VOPs, for the situations where the bit rate is scarce.
- Even for larger random access periods, the IST MVO algorithm still behaves generically better than VM5, as can be seen in Table 6.17 and is illustrated in Figure 6.49. For this case, the IST MVO algorithm has an average gain of over 0.5 dB or, equivalently, a saving of approximately 11% in terms of bit rate. Notice that these gains are obtained with a marginal higher Scene PSNR Variation.

Table 6.17 – MVO average PSNR and bit rate gains of the proposed rate control algorithm for the *Coastguard* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	2.84	0.25	-61.34	-5.20
QCIF@15Hz	1.06	0.12	-19.92	-2.47
CIF@15Hz	1.13	0.55	-21.78	-10.94
CIF@30Hz	0.05	0.16	-1.11	-3.41
	1.27	0.27	-26.04	-5.51

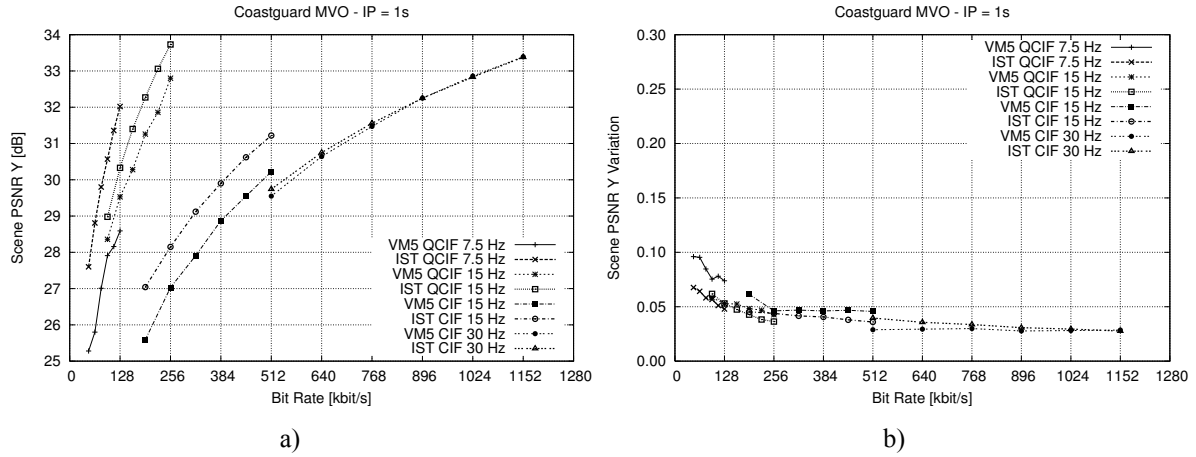


Figure 6.46 – Coastguard MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation

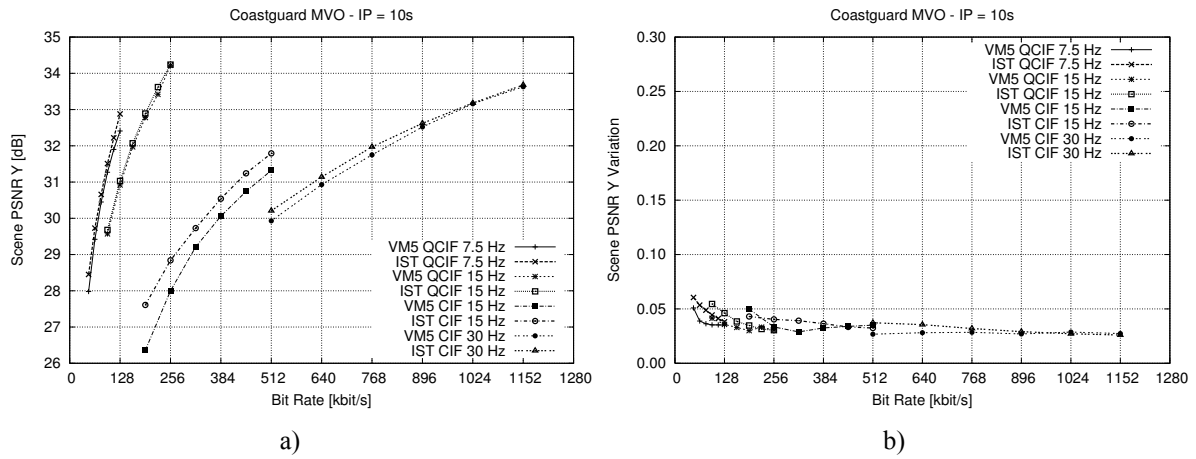


Figure 6.47 – Coastguard MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation

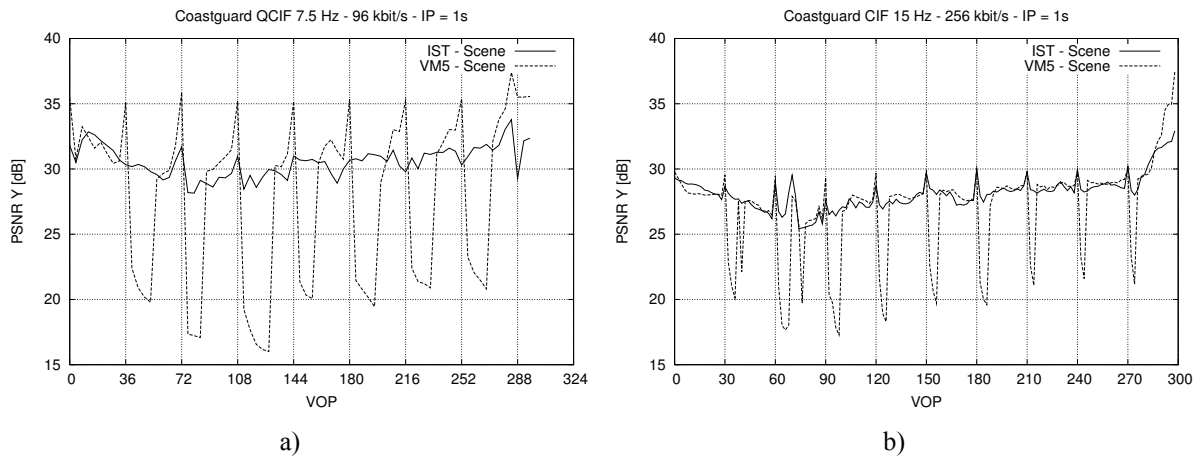


Figure 6.48 – Coastguard MVO Scene PSNR (Intra period 1s): a) QCIF@7.5Hz 96 kbits/s; b) QCIF@15Hz 256 kbits/s

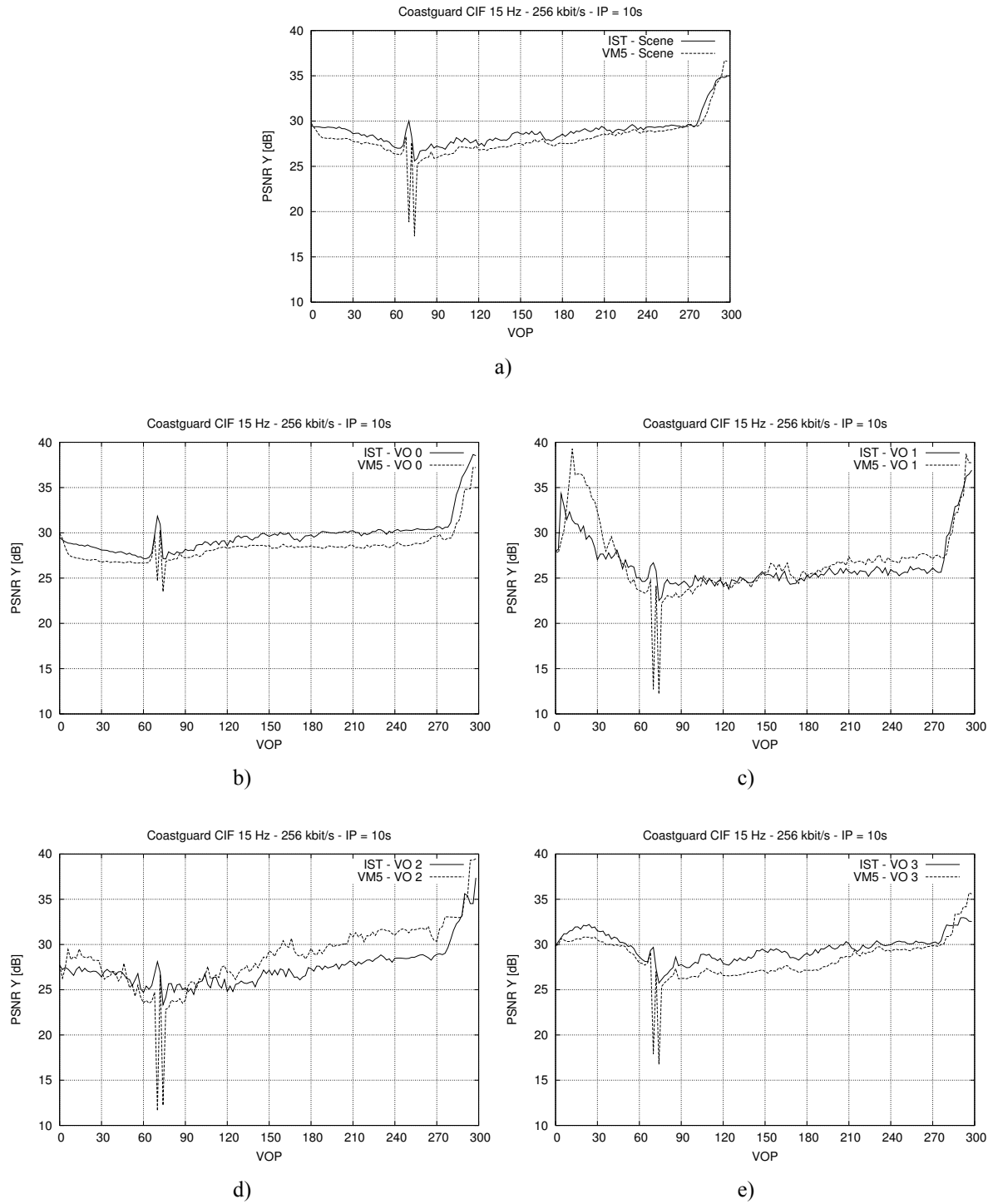


Figure 6.49 – Coastguard MVO Scene and VOs PSNR (Intra period 1s): a) Scene PSNR; b) VO 0 PSNR; c) VO 1 PSNR; d) VO 2 PSNR; e) VO 3 PSNR

LOW-MOTION VIDEO SEQUENCES

Table 6.18 summarizes the MVO encoding results illustrated in Figure 6.50 – Figure 6.53 for the low-motion *Bream* sequence (2 VOs). From these results the following conclusions may be extracted:

- For the lowest spatio-temporal resolution (i.e., for QCIF@7.5Hz), where the bit rate is typically scarce, the IST algorithm achieves a better Average Scene PSNR, for both random access points conditions (see Table 6.18).
- For the points where the VM5 MVO algorithm outperforms the IST MVO algorithm in terms of Average Scene PSNR, the gains are typically marginal and this is accomplished at the expense of higher Scene PSNR variations (see Figure 6.50 and Figure 6.51), therefore with larger differences in spatial quality between the two VOs in the scene.
- The gains in terms of average PSNR of the VM5 MVO algorithm, in this case, are obtained at the expense of encoding VO 0 (*Background*) with extremely high qualities, which will have a high impact in terms of Scene PSNR computed through (6.133), although this is visually worthless.
- On the other hand, the IST MVO algorithm attempts to better allocate the available bit rate in order to favor the harder to encode VO 1 (*Fish*), at the expense of decreasing the spatial quality of the easier to code VO 0 where quality investments are subjectively irrelevant above a certain quality.

The parameter that controls this trade-off is γ_D in equation (6.96). As it is illustrated in Figure 6.52 for a random access period of 10 s, when γ_D is increased from 0.2 to 0.5, the Scene PSNR Variation tends to decrease significantly (see Figure 6.51b and Figure 6.52b). As can be seen in Figure 6.53a, where the IST and the VM5 MVO algorithms achieve the same Average Scene PSNR, the IST MVO algorithm reduces the quality of VO 0 along time in order to better encode VO 1. This is even more evident when $\gamma_D = 0.5$ (see Figure 6.53b). Notice, however, that since VO 0 is relatively easy to encode, the decrease in PSNR of VO 0 of approximately 3.7 dB leads to an increase in the PSNR of VO 1 of only 0.5 dB. Nevertheless, it is important to stress that this is a better trade-off than encoding the *Background* with such very high (and useless) quality as in VM5.

Table 6.18 – MVO average PSNR and bit rate gains of the proposed rate control algorithm for the *Bream* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	3.14	0.23	-40.39	-4.86
QCIF@15Hz	0.01	-0.16	0.84	3.05
CIF@15Hz	0.80	-0.06	-15.47	1.12
CIF@30Hz	-0.25	-0.10	4.89	2.03
	0.93	-0.02	-12.53	0.34

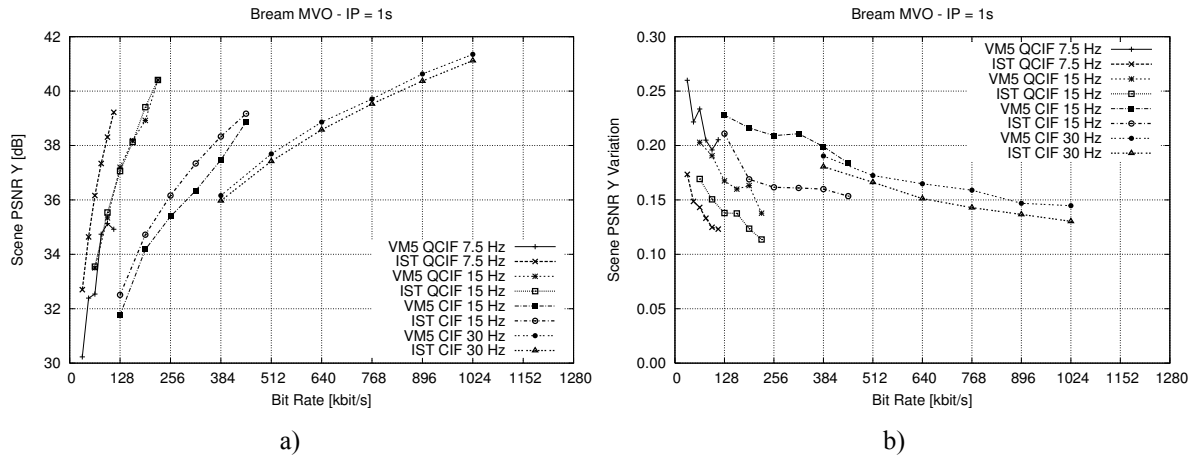


Figure 6.50 – Bream MVO (Intra period 1s) ($\gamma_D = 0.2$): a) Average Scene PSNR; b) Scene PSNR Variation

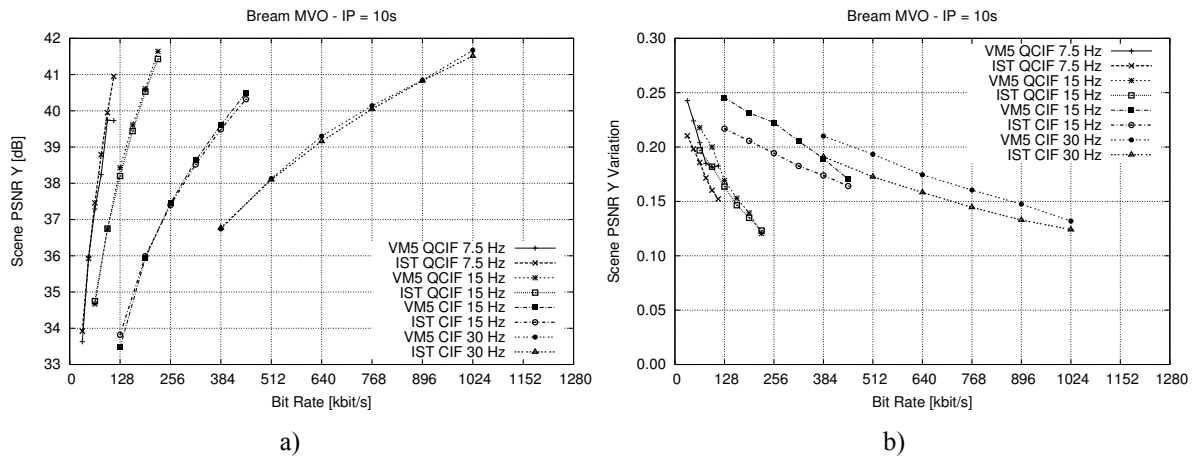


Figure 6.51 – Bream MVO (Intra period 10s) ($\gamma_D = 0.2$): a) Average Scene PSNR; b) Scene PSNR Variation

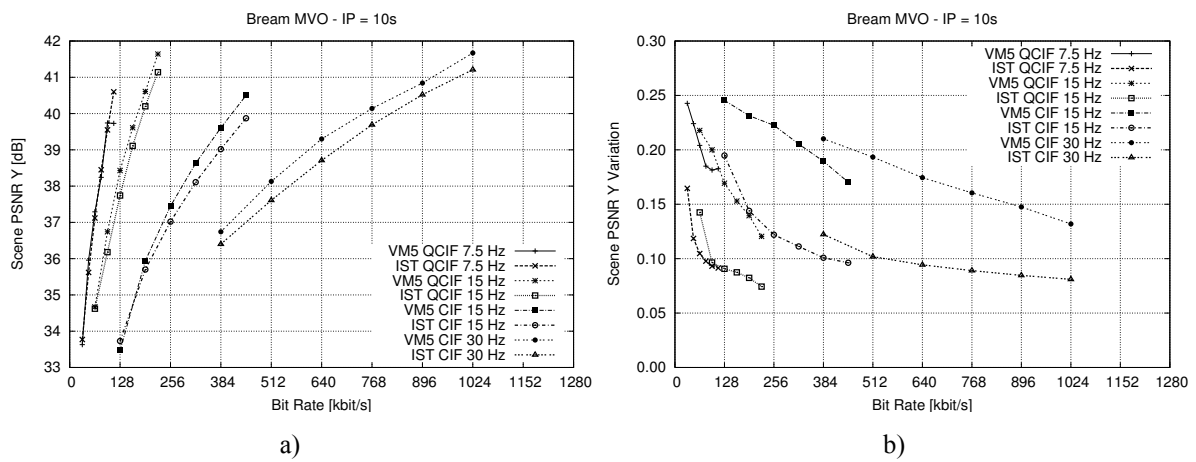


Figure 6.52 – Bream MVO (Intra period 10s) ($\gamma_T = 0.5$): a) Scene Average PSNR; b) Scene PSNR Variation

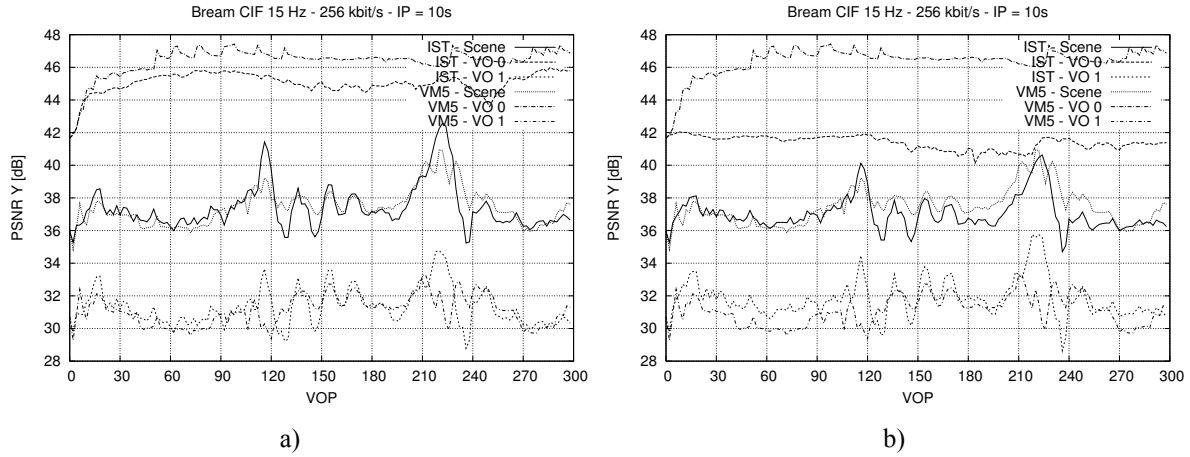


Figure 6.53 – Scene and VOs PSNR for the Bream sequence: a) $\gamma_D = 0.2$; b) $\gamma_D = 0.5$

Table 6.19 summarizes the MVO encoding results illustrated in Figure 6.54 – Figure 6.57 for the low-motion *News* sequence (4 VOs). Before analyzing these results it is worthless to referee some idiosyncrasies of this sequence that make it especially annoying for rate control purposes, namely:

- The texture of VO 0 (*Background*) is static (only the shape changes along time), which means that it can easily achieve very high qualities biasing the Scene PSNR and causing large deviations in the bit allocation at scene- and object-level⁴⁶.
- VO 1 (*Dancers*) has been converted from a temporal resolution of 25Hz to 30Hz by repeating one frame for every set of 5 original frames (this means that periodically two consecutive VOPs are exactly equal⁴⁷), which means that periodically the VO characteristics change abruptly, notably the prediction error and motion activity
- VO 3 (*Logo*) is a static synthetic object, which tends also to cause large deviations at the scene- and object-level bit allocations.

In the case of the VM5 MVO algorithm, the eccentricities of this sequence produce, for some encoding conditions, non-monotonic rate-distortion curves as can be seen in Figure 6.54 for CIF@15Hz. This situation is illustrated in Figure 6.56 showing the Scene PSNR for four different target bit rates. From Figure 6.56 the following observations should be highlighted:

- The IST MVO algorithm achieves a more stable Scene PSNR without skipping VOs, while the VM5 MVO algorithm tends to skip the VOs immediately after I-VOPs reflected in the abrupt Scene PSNR drops
- This phenomenon is more critical for the higher target bit rates since the VM5 tends to use lower quantization parameters. In this situations, if the rate control tries to further decrease the quantization parameter the result is an extremely high bit production⁴⁸. This behavior reveals some instability that results from incorrect bit allocations and

⁴⁶ The coding mode control module tends to decrease the quantization parameter for this type of objects, leading sometimes to an excessive and worthless production of bits.

⁴⁷ This situation only has impact for the 30Hz test conditions.

⁴⁸ From the VM5 algorithm output statistics, it can be observed that this behaviour occurs when the quantization parameter is typically between 1 and 2.

poor control of bit production and buffer occupancy. It is important to refer also that in these cases the VM5 rate control algorithm usually violates the VBV.

Figure 6.57 illustrates the behavior of the two rate control algorithms, for one target bit rate of each spatio-temporal condition. From Figure 6.57 the following observations should be highlighted:

- Although the IST MVO algorithm produces an Average Scene PSNR lower than VM5 by approximately 1 dB for the three first cases, the evolution along time reveals a more stable behavior, notably without skipping encoding time instants and producing more steady qualities.
- Figure 6.57d illustrates the problems caused by VO1 when encoding the *News* sequence at 30Hz; in this case, the periodically repeated VOPs originate the “saw tooth” appearance of the Scene PSNR curve.

Although this sequence is not the typical sequence one may expect in a real scenario, the following conclusions can still be extracted from the results obtained for the *News* sequence:

- The IST MVO algorithm can still perform adequately for both random access conditions.
- In fact, for an Intra period of 1s, the IST MVO algorithm outperforms the VM5 MVO algorithm by approximately 2 dB on average while maintaining a smaller Scene PSNR Variation.
- For an intra period of 10 s, although the VM5 MVO algorithm achieves generally a higher Average Scene PSNR (between approximately 1.1 dB and 0 dB (see Table 6.19), this is accomplished at the expense of very high Scene PSNR Variations and a large number of skipped encoding time instants.

Table 6.19 – MVO average PSNR and bit rate gains of the proposed rate control algorithm for the *News* sequence

Spatio-Temporal Resolution	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
QCIF@7.5Hz	1.86	-1.12	-47.70	30.80
QCIF@15Hz	1.23	-1.16	-17.79	18.96
CIF@15Hz	2.50	-0.49	-33.92	8.87
CIF@30Hz	1.91	0.03	-22.72	-0.71
	1.91	-0.69	-30.53	14.48

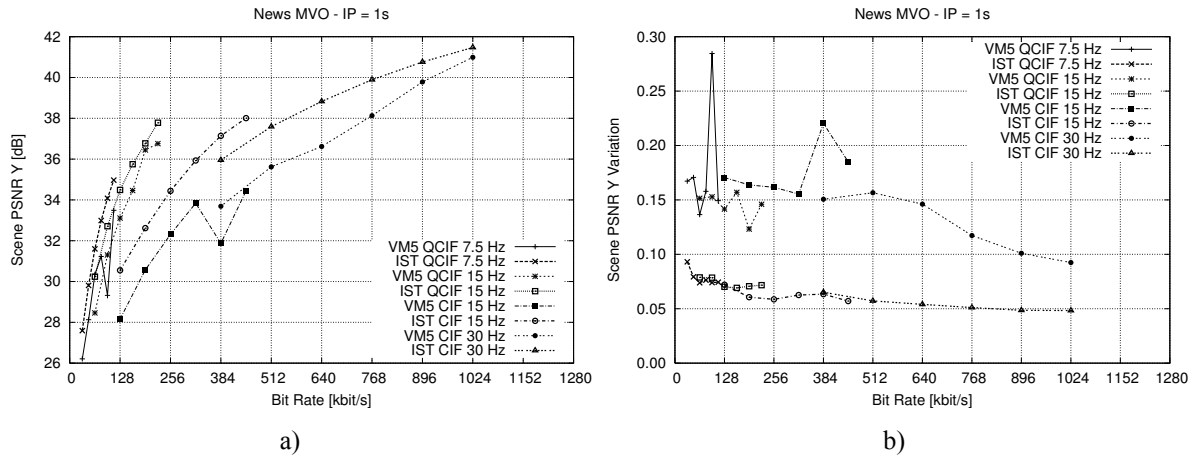


Figure 6.54 – News MVO (Intra period 1s): a) Average Scene PSNR; b) Scene PSNR Variation

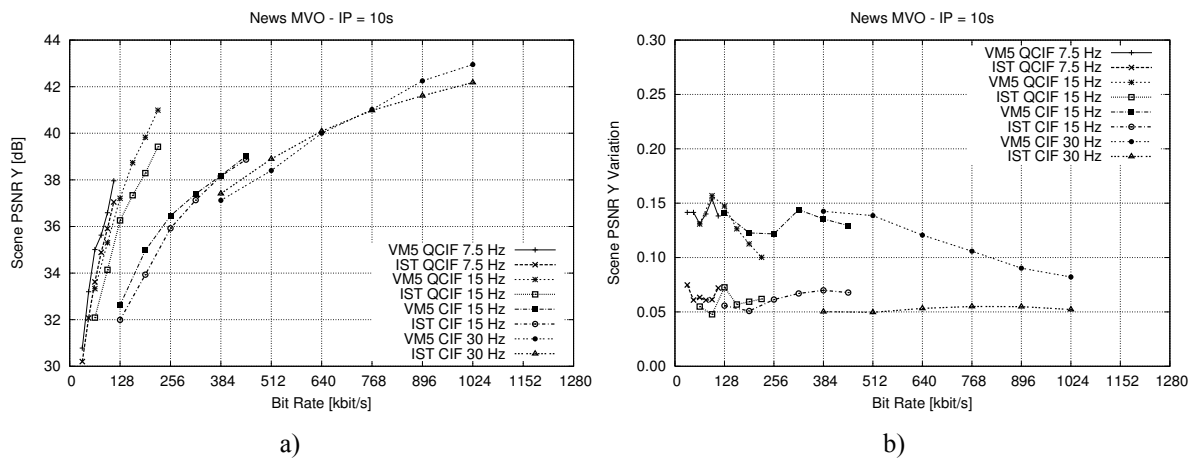
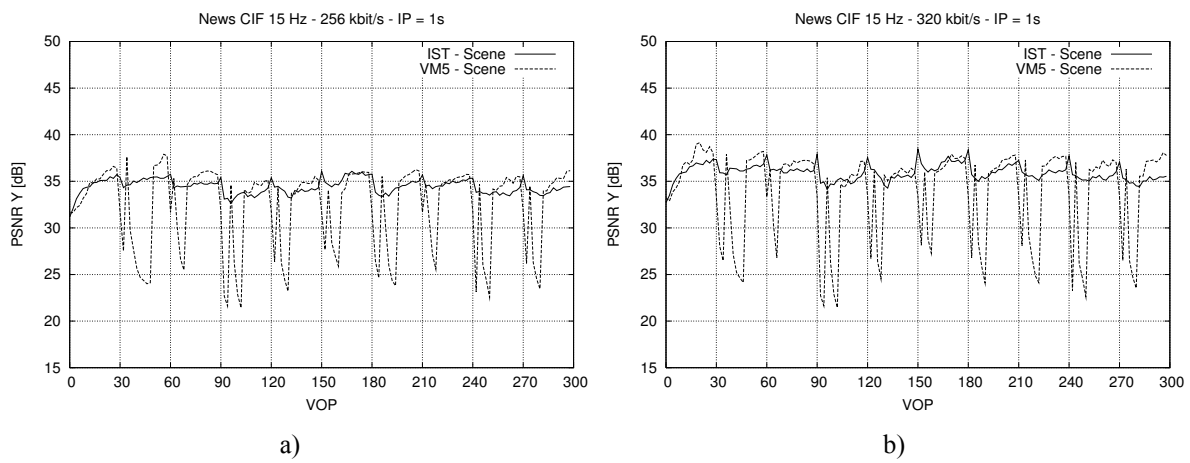


Figure 6.55 – News MVO (Intra period 10s): a) Average Scene PSNR; b) Scene PSNR Variation



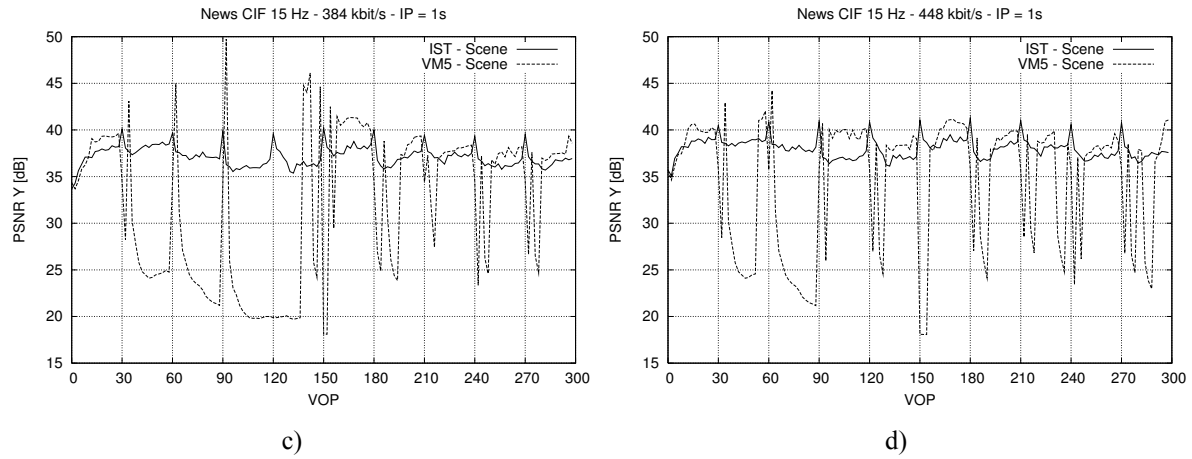


Figure 6.56 – News MVO Scene PSNR (Intra period 1s): a) CIF@15Hz 256 kbit/s; b) CIF@15Hz 320 kbit/s; c) CIF@15Hz 384 kbit/s; d) CIF@15Hz 448 kbit/s

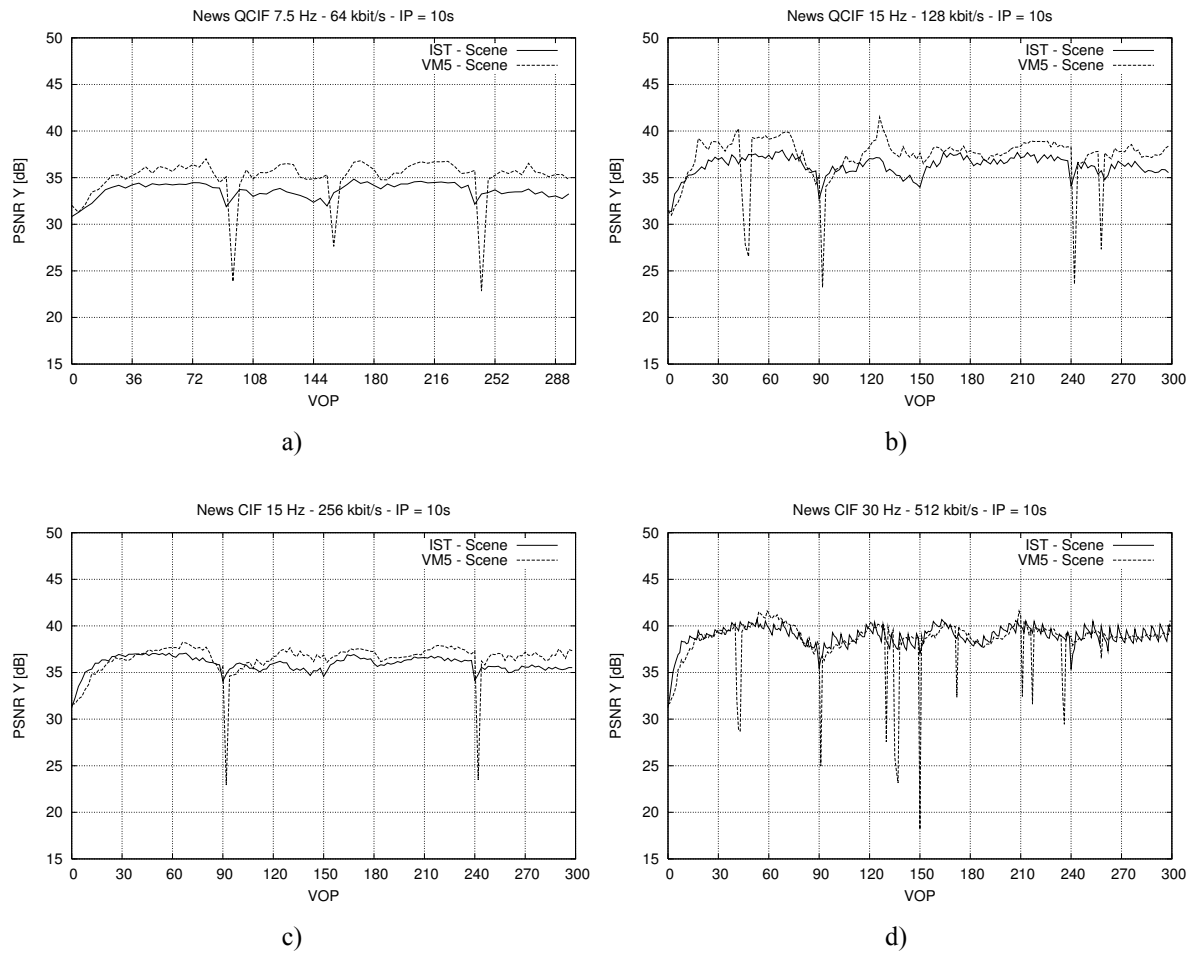


Figure 6.57 – News MVO Scene PSNR (Intra period 10s): a) QCIF@7.5Hz 64 kbit/s; b) QCIF@15Hz 128 kbit/s; c) CIF@15Hz 256 kbit/s; d) CIF@30Hz 640 kbit/s

SUMMARY OF MVO PERFORMANCE ANALYSIS

In order to better illustrate the relative behavior of the two MVO rate control algorithms for the different encoding conditions, Table 6.20 to Table 6.23 summarize the Scene PSNR and bit rate gains of the proposed MVO algorithm for the different spatio-temporal encoding conditions, notably: QCIF@7.5Hz, QCIF@15Hz, CIF@15Hz, and CIF@30Hz. From these tables the following conclusions may be derived:

- Clearly, the IST MVO algorithm has higher gains for a random access point every second, ranging from an average gain of 0.9 dB for CIF@30Hz to 2.6 dB for QCIF@7.5Hz, in terms of Scene PSNR, and from 12.5% to 49 %, in terms of bit rate savings.
- For a single random access point only at the beginning of the sequence, the gains are lower, ranging from 0.2 dB for QCIF@15Hz to 0.6 dB for CIF@30Hz, in terms of Scene PSNR, and from 4% to 9.7%, in terms of bit rate savings.
- The major gains of the IST MVO algorithm are obtained for the high-motion sequences, and for the shorter random access conditions.

Table 6.20 – MVO average PSNR and bit rate gains for QCIF@7.5Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Stefan	2.56	1.85	-46.68	-36.79
Coastguard	2.84	0.25	-61.34	-5.20
Bream	3.14	0.23	-40.39	-4.86
News	1.86	-1.12	-47.70	30.80
	2.60	0.30	-49.03	-4.01

Table 6.21 – MVO average PSNR and bit rate gains for QCIF@15Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Stefan	1.94	1.89	-37.38	-35.83
Coastguard	1.06	0.12	-19.92	-2.47
Bream	0.01	-0.16	0.84	3.05
News	1.23	-1.16	-17.79	18.96
	1.06	0.17	-18.56	-4.07

Table 6.22 – MVO average PSNR and bit rate gains for CIF@15Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Stefan	2.57	2.15	-38.77	-37.66
Coastguard	1.13	0.55	-21.78	-10.94
Bream	0.80	-0.06	-15.47	1.12
News	2.50	-0.49	-33.92	8.87
	1.75	0.54	-27.49	-9.65

Table 6.23 – MVO average PSNR and bit rate gains for CIF@30Hz

Sequence	PSNR [dB]		Bit Rate [%]	
	IP = 1s	IP = 10s	IP = 1s	IP = 10s
Stefan	1.84	2.09	-30.95	-35.46
Coastguard	0.05	0.16	-1.11	-3.41
Bream	-0.25	-0.10	4.89	2.03
News	1.91	0.03	-22.72	-0.71
	0.89	0.55	-12.47	-9.39

6.8 Final Remarks

This chapter proposed a new rate control algorithm for low-delay and constant bit rate application scenarios capable of efficiently encode single and multiple arbitrarily shaped video objects under a wide range of bit rates and spatio-temporal resolutions. This algorithm can efficiently deal with deviations between the ideal and the actual behavior of the scene encoder.

To deal with these deviations between the theoretical models and the actual coding results, it was necessary to develop adequate adaptation and compensation mechanisms that were able to track these deviations and compensate them in order to allow a stable and efficient operation of the encoder. These two problems (adaptation and compensation) became the main focus of this chapter and were tackled along the different modules composing the architecture of the proposed rate control algorithm. In this context, the following two rate control approaches have been combined in the proposed rate control algorithm:

- **Feedback rate control** – This rate control approach compares the actual coding results (e.g., VBV buffer occupancy, bit rate, quality, etc.) with the target results and take an action based on the difference, e.g., whenever the past encoding decisions resulted in a successive decrease of the VBV buffer occupancy, the rate control mechanism compensates this behavior by coarsely encoding the next incoming VOPs.
- **Feedforward rate control** – This rate control approach plans in advance what will be the encoding result of a certain set of encoding parameters and acts before deviations

occur, e.g., based on all or a subset of the input data, the encoder selects the set of encoding parameters that should produce the desired result, e.g., target VBV occupancy and target scene quality.

Being a complex system, the rate control mechanism is composed, typically, of a large number of parameters. Consequently, setting the values for these parameters is a critical task when developing an efficient rate control algorithm, since some of these parameters can have a high impact on the algorithm performance, e.g., the bit allocation weights between I-, P-, and B-VOPs, and the bit allocation weights for the several VOs in the scene.

Typically, the traditional approach used to solve this control problem is a tuning approach based on the assumption that the control system has constant but unknown parameters. In this case, the rate control algorithm parameters are estimated off-line, based on training sequences. Usually, a critical weakness of this constant parameters control is some instability, notably when the operational conditions change significantly, e.g., image content, buffer occupancy, etc.

In this Thesis, this weakness is circumvented by proposing an adaptive approach based on the adaptation of the parameters describing the encoding process and some parameters of the rate controller during the encoding process.

The main outcome of this chapter is, therefore, a proposal for a novel rate control algorithm that can operate in SVO or MVO encoding modes, meeting the important requirements of a generic rate control algorithm for object-based video encoding and in particular the constraints of MPEG-4 compliant video encoding, such as maximizing the quality of the decoded video scenes and meeting the constraints of the video buffering verifier models. The main novelties of this proposed rate control algorithm could be summarized as:

- A bit allocation algorithm for MVO encoding capable of controlling the bit rate allocation for multiple video objects encoded at different temporal resolutions. This module defines a set of nominal bit allocations and corresponding compensation and adaptation mechanisms at different levels of the syntactic video data organization taking into account the changing coding complexities of the different VOs in scene. The problem of bit allocation for multiple video objects encoded at different temporal resolutions was first addressed in the context of this Thesis.
- A video buffering verifier control algorithm operating at scene- and object-level capable of maintaining full MPEG-4 video buffering verifier compliance, notably VBV compliance, through the accurate control of the VBV buffer occupancy. This mechanism feedforwardly defines precise target VBV buffer occupancies for each encoding time instant based on the amount of data to encode, its coding complexity, the coding type weights, and the relative encoding time instant position in the GOS. Additionally, a fine VBV control at MB-level sets an efficient trade-off between accurate VBV control and spatial quality smoothness. This mechanism besides guaranteeing fully compliance with the MPEG-4 video buffering verifier mechanism produces more stable buffer occupancy and leads to smoother quality variations along consecutive encoding time instants, notably when compared with the MPEG-4 Visual Annex L rate control algorithms.
- A rate-distortion modeling approach describing the encoding process at the VOP and MB-level in the form of rate-quantization functions that allow the rate controller to feedforwardly predict the behavior of the scene encoder, and consequently, reduce the amount of compensation needed to bring the scene encoder to the ideal behavior.

- A coding mode control algorithm for selecting the quantization parameter at VOP and MB-level. This algorithm is capable of achieving accurate VOP bit allocations while simultaneously maintaining spatial quality smoothness, by setting a trade-off between smooth MB quantization parameter changes to favor smooth spatial quality, and accurate VBV control.
- The interaction between the different modules composing the rate control algorithm, i.e., scene analysis for resource allocation, spatio-temporal resolution control, bit allocation, video buffering verifier control, rate-distortion modeling, and coding mode control. Although each module has its own merits, it is the interaction between all of them that brings the performance gains of the overall algorithm, notably due to the frequent conflicting goals of some of these modules.

The proposed solution is compared with the usual reference algorithms, namely, VM8 [108] for SVO encoding, and VM5 [102] for MVO encoding, and its relative gains are assessed. In summary, for SVO and MVO encoding, the proposed rate control algorithm outperforms, respectively, the VM8 and VM5 algorithms in terms average quality and quality smoothness.

Chapter 7

Achievements and Future Directions

7.1 Achievements

The object-based audiovisual content representation model addresses the requirements of many multimedia applications supporting new and improved functionalities, such as content-based interactivity, universal accessibility through a wide range of terminals and networks, and improved coding efficiency. In the context of this new content representation framework, an audiovisual scene is understood as a composition of various independent objects with their own characteristics, such as the spatio-temporal localization and shape (in the case of 2D visual objects). These objects can be independently accessed and manipulated allowing the user to experience new ways of consuming the audiovisual content.

This new representation framework is a step forward in the audiovisual world due to the extra flexibility, in terms of content representation, introduced by the object-based approach. In this context, some of the traditional non-normative tools, such as bit rate control, need to be redesigned when this object-based representation is used. This Thesis addressed this challenge proposing various solutions for the object-based video coding rate control problems.

The first achievement of this Thesis concerns, therefore, the analysis of the object-based representation approach impact in terms of rate control. This analysis was provided in Chapter 3, where the objectives and constraints of rate control for frame-based and object-based video coding were presented. In this context, this Thesis identified the new dimensions of object-based rate control and proposed the strategies to address them. As a result, two new object-based rate control strategies were identified and defined: the semantic resolution control and the amount of content control. Additionally, a generic object-based rate control framework was proposed where, besides the natural interfaces of the rate control mechanism with the scene encoder, also the interfaces with the video analysis and the scene authoring modules are described.

The second achievement of this Thesis concerns the need to control the decoding complexity in object-based video encoding. In this type of coding architectures, besides the need to control the bit rate variability of the encoded video scene, it is also necessary to control the amount of memory and the processing capabilities required to decode it. This problem has been dealt with in Chapter 4. After a detailed analysis of the MPEG-4 video buffering verifier mechanism, where its fundamental drawbacks were highlighting, Chapter 4 proposed and discussed an architecture for the integration of this mechanism with the rate control mechanism. Still in this field, considering the drawbacks of the MPEG-4 video buffering verifier mechanism, Chapter 4 proposed alternative models for the video reference memory verifier and for the video complexity verifier allowing a more efficient use of the available decoding resources. This later is related to a new proposed approach for determining the decoding complexity of object-based encoded video scenes based on the MB coding tools relative complexity weights.

The third achievement of this Thesis concerns the problem of rate-distortion modeling. In this context, Chapter 5 proposes a set of efficient (lower fitting error) models for Intra and Inter coding (I-VOPs and P-VOPs) in the form of rate-quantization, distortion-quantization, and rate-distortion functions. In the case of Inter coding, a new modeling approach was proposed to overcome the complexity of estimating bidimensional where the rate-quantization and distortion-quantization functions are approximated by two components: the stationary model and the delta model. The main advantage of these models is the small number of parameters and the low fitting error when compared with the corresponding suggested MPEG-4 model.

The fourth, and major, achievement of this Thesis concerns the proposal of a novel SVO and MVO rate control algorithm made in Chapter 6. This algorithm meets the important requirements of a generic rate control mechanism for object-based video encoding and in particular meets also the constraints of MPEG-4 compliant video encoding, notably, of the MPEG-4 video buffering verifier models. A thorough comparison with the suggested MPEG-4 rate control solutions showed the superior performance of the proposed solution, both in terms of achieving higher average spatial quality and smooth quality variations along time and among the several VOs composing the scenes. This way, the initial goal of defining an efficient rate-control solution for low-delay video encoding has been accomplished,

7.2 Future Directions

This Thesis has been focused on the development of rate control techniques for object-based video coding systems. In particular, the algorithmic development has been directed towards an efficient rate control algorithm for low-delay object-based video encoding that could guarantee efficient interoperability between video codecs targeting compliance with a selected MPEG-4 video profile@level. This section discusses some of the work items directly related to the topics developed in this Thesis, which are worthwhile to be pursued in the future.

EXPLOITATION OF THE NEW RATE CONTROL DIMENSIONS

This Thesis identified new dimensions for rate control, notably, the semantic resolution control and the amount of content control, that were not covered in the work developed here since the effort was put on developing an efficient rate control algorithm solely for the control of the scene encoder. Therefore, the development of adequate mechanisms that implement the interaction between the rate control mechanism and the video analysis (real-time encoding) or the scene authoring (off-line encoding) – see Figure 3.3 – are still open issues that deserve future consideration.

For real-time encoding, the problem to solve is to define adequate feedback mechanisms from the rate control to the scene analysis. In this context, it is necessary to define the type of information that is useful for the video analysis module, and the type of information available at the video analysis module that can be used by the rate control mechanism. In this case, the video analysis algorithms have a strong impact on the type of feedback allowed. In this field, it is important, for example, to extract the relevance of the objects, and eventually to define the automatic prioritization according to this relevance (see, for example, [179]).

In the case of off-line encoding, the problem to solve is to define adequate feedback information from the rate controller to the scene authoring module in order that the author may be able to select, for example, the adequate profile@level to encode the given scene. If the profile@level is not changeable, the author should be able to select how many and which objects can be coded with acceptable quality, or to prioritize the different objects according to their semantic relevance. In this context, it would be very useful to define adequate graphical user interfaces (GUIs) for off-line encoding that could be used to gather information from the user to the video analysis module and to the rate control mechanism (see Figure 3.3). Additionally, these GUIs should also give feedback to the author about the spatio-temporal quality being achieved in order that he/she can suggest possible actions to the video analysis (e.g., eliminate objects or merging them in order to reduce shape information) or to the rate control (e.g., target qualities for the various objects according to their subjective relevance).

RATE-DISTORTION MODELING INTEGRATING HUMAN VISUAL SYSTEM FACTORS

In the context of object-based video coding, which still waits to explode in terms of big successful applications, bit rate control still has a long way to go to fully consider the specific characteristics of each object in the scene and optimize the shape versus texture distortion if departure from the most typical lossless shape model is to be accepted. When dealing with objects, semantics may play a significant role and thus impact on the bit rate allocation since objects with different semantic value, very likely deserve/require different distortion. Besides the continuous search for better RD modeling for the available and emerging coding schemes or more efficient coding optimization techniques, bit rate control did not yet consider enough the impact of the human visual system (HVS) in their modeling and optimization. For example, performing exhaustive optimization for a mean square error distortion metric may reveal itself rather meaningless for specific types of images where this metric is less good in terms of expressing subjective quality evaluation; and the subjective impact is always the last assessment criteria. Another field with high subjective impact is lossy shape coding, since the HVS does not tolerate well visually noticeable distortions in the shape of semantically relevant objects; therefore, to pursue the goal of bit allocation according to the semantic relevance of each object, appropriate lossy/near-lossless shape coding is required. In this case, shape pre-processing techniques that allow shape simplification aiming at reducing the shape data bit rate, without compromising the subjective quality of the decoded shape, should be developed. In this context, a major trend in bit rate control research may be the more intense integration of human visual system factors in the modeling of distortion, both for texture and for shape information.

RATE CONTROL FOR LAYERED AND MULTIPLE DESCRIPTION VIDEO CODING

Another major trend in video coding is scalable coding, both layered or not, like multiple description coding (MDC). The recent MPEG activity in this area has revealed a large number of applications with several industries interested in the development of a new (more efficient and flexible) scalable video coding solution. Nowadays, one of the trends is to consider

scalable coding essential to deal with video streaming to heterogeneous environments, notably different types of transmission channels and receiver characteristics. This focus is justified by the growing number of available networks, both wired and wireless, with many different characteristics, and with a large range of connection bandwidths. For some of them, the available bandwidth can be highly dynamic, and transmission errors, congestions, and information losses can occur in an unpredictable way. In this context, the capabilities of the user devices also have an important role, since they have very different characteristics, notably in terms of the display spatio-temporal-quality resolution, memory and computational power. This scenario has a close relationship with the MPEG-21 Digital Item Adaptation (DIA) standard [180], which contributes to enabling the delivery of video to multiple networks and terminals and the emergence of the “encode once, decode multiple ways” paradigm.

For layered video coding efficient rate control solutions are essential to jointly optimize the distortion for each layer considering a certain rate or vice-versa. For multiple description coding efficient bit allocation techniques are essential to define which data, e.g., transform coefficients, go within each description and with which level of distortion (related to the quantization) in order to minimize the redundancy between descriptions while maximizing the global and elementary qualities.

RATE-DISTORTION-COMPLEXITY OPTIMIZATION

The fact that video encoders may accept nowadays a growing amount of complexity (not to speak about off-line encoding) motivates the increasing usage of optimization techniques such as the Lagrangian optimization. However, it must be recognized that, typically, RD theory only considers as limited resource the rate associated to the transmission channel or storage device. In a world where limited capacity devices such as mobile terminals have a growing importance, other resources, besides the bandwidth, such as the computational resources (complexity) and the battery life, have also to be recognized as scarce and limited.

In fact, Chapter 4 showed that the estimated decoding complexity of a given video scene is highly dependent on the coding tools used for each coding unit (i.e., MB). Consequently, encoding optimization in these scenarios should also take into account the decoding complexity as a scarce resource. This type of recognition implies that the relevant problem to solve is not anymore a RD optimization problem but rather a RDC(omplexity) problem to better express the practical situation. In this context, the problem to solve is to improve the decoding complexity models developed in this Thesis (notably, for other video coding tools, besides the ones considered here) and to devise adequate RDC optimization techniques.

JOINT RATE CONTROL FOR MULTIPLE VIDEO SEQUENCES

A particular case of multiple video object rate control is the simultaneous encoding of multiple video sequences (MVS) in the context of frame-based video encoding, where the video objects are now frame-based video sequences with possible different spatial and temporal resolutions. This scenario is particularly interesting in the context of the new video coding standard H.264/AVC (Advanced Video Coding), where the joint encoding of multiple programs (video sequences) can bring increased compression efficiency due to the so-called statistical multiplexing gain. Therefore, a future research direction could be the exploitation of the proposed rate control architecture and tools (with the necessary adaptations) for encoding multiple video programs compliantly with the recent H.264/AVC video coding standard.

To finalize, it is important to say that, although object-based video coding is still in an early stage of deployment, mainly because many of the object-based functionalities were identified well in advance to market needs, it is expected that in the near future this video representation approach will takeoff. Therefore, it is expected that the techniques proposed in this Thesis, as well as the ones suggested as future work and that still have to be developed or further improved, will be necessary in the near future. This expectation is based on the growing demand for applications that can clearly benefit from an object-based approach. In addition to this, today's users want to have access to those applications anywhere and at anytime, which means that the video content may have to be delivered over a multitude of networks for a multitude of terminals. Consequently, efficient compliant video encoding is of utmost importance to enable these applications, as, for example, video personal communications over mobile networks or over the Internet, as well as the broadcasting of multimedia content over these same networks.

Annex A

Additional Rate and Distortion

Modeling Results for Intra Coding

Table A.1 to Table A.32 present the additional rate-quantization, distortion-quantization, and rate-distortion model parameters results for the *News*, *Kayak*, *Mother and Daughter (M&D)*, and *Football* sequences (see Figure 5.13) in QCIF and CIF formats, indicating for each model parameter its minimum, maximum, and mean value, and standard deviation, measured over all encoded pictures for each sequence.

A.1 Rate-Quantization Model Parameters

Table A.1 – Rate-quantization model parameters for the News sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	4.26	6.38	5.14	0.52
	<i>b</i>	5.80	7.88	6.66	0.51
	<i>c</i>	0.10	0.15	0.13	0.01
II	<i>a</i>	4.61	4.99	4.85	0.10
	<i>b</i>	-0.30	-0.20	-0.26	0.03
	<i>c</i>	0.64	0.67	0.65	0.01
III	<i>a</i>	6.19	7.52	6.89	0.34
	<i>b</i>	0.39	0.62	0.51	0.05
	<i>c</i>	0.87	0.93	0.90	0.01
IV	<i>a</i>	-2.80	-2.31	-2.60	0.12
	<i>b</i>	6.61	7.36	7.06	0.19
	<i>c</i>	0.09	0.11	0.10	0.01

Table A.2 – Rate-quantization model parameters for the News sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	28.26	38.21	36.30	2.82
	<i>b</i>	29.51	39.41	37.53	2.82
	<i>c</i>	0.02	0.03	0.02	0.00
II	<i>a</i>	3.30	3.50	3.42	0.05
	<i>b</i>	-0.02	0.02	-0.00	0.01
	<i>c</i>	0.79	0.84	0.81	0.01
III	<i>a</i>	2.98	3.50	3.26	0.15
	<i>b</i>	-0.11	0.01	-0.05	0.03
	<i>c</i>	0.77	0.81	0.79	0.01
IV	<i>a</i>	-1.05	-0.75	-0.91	0.08
	<i>b</i>	3.98	4.43	4.24	0.12
	<i>c</i>	0.09	0.10	0.09	0.00

Table A.3 – Rate-quantization model parameters for the Kayak sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.10	3.16	2.50	0.20
	<i>b</i>	3.84	4.72	4.17	0.17
	<i>c</i>	0.20	0.27	0.24	0.02
II	<i>a</i>	5.13	7.28	5.92	0.49
	<i>b</i>	-0.89	-0.38	-0.55	0.10
	<i>c</i>	0.53	0.66	0.61	0.03
III	<i>a</i>	8.97	14.14	11.27	1.13
	<i>b</i>	0.89	1.39	1.12	0.11
	<i>c</i>	0.99	1.18	1.07	0.05
IV	<i>a</i>	-5.08	-2.86	-3.65	0.49
	<i>b</i>	7.56	11.34	8.96	0.84
	<i>c</i>	-0.09	0.07	-0.01	0.04

Table A.4 – Rate-quantization model parameters for the Kayak sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.48	5.14	3.46	0.57
	<i>b</i>	4.26	6.56	5.02	0.52
	<i>c</i>	0.13	0.23	0.18	0.02
II	<i>a</i>	4.29	6.73	5.16	0.47
	<i>b</i>	-0.70	-0.22	-0.38	0.09
	<i>c</i>	0.56	0.68	0.64	0.02
III	<i>a</i>	6.07	12.13	8.51	1.21
	<i>b</i>	0.49	1.08	0.79	0.13
	<i>c</i>	0.94	1.08	1.01	0.04
IV	<i>a</i>	-4.42	-2.04	-2.87	0.46
	<i>b</i>	6.05	10.33	7.59	0.83
	<i>c</i>	-0.02	0.09	0.03	0.03

Table A.5 – Rate-quantization model parameters for the M&D sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	4.51	6.31	5.33	0.34
	<i>b</i>	5.74	7.50	6.53	0.33
	<i>c</i>	0.12	0.16	0.14	0.01
II	<i>a</i>	3.34	3.65	3.45	0.08
	<i>b</i>	-0.16	-0.11	-0.13	0.01
	<i>c</i>	0.76	0.79	0.78	0.01
III	<i>a</i>	4.75	5.71	5.17	0.20
	<i>b</i>	0.47	0.67	0.57	0.036
	<i>c</i>	1.03	1.08	1.06	0.01
IV	<i>a</i>	-1.43	-1.15	-1.27	0.06
	<i>b</i>	4.38	4.89	4.58	0.12
	<i>c</i>	-0.02	-0.004	-0.01	0.004

Table A.6 – Rate-quantization model parameters for the M&D sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	86.02	86.06	86.04	0.01
	<i>b</i>	86.91	86.95	86.93	0.01
	<i>c</i>	0.01	0.01	0.01	0.00
II	<i>a</i>	2.34	2.55	2.44	0.04
	<i>b</i>	-0.01	0.02	0.00	0.01
	<i>c</i>	0.92	0.98	0.94	0.01
III	<i>a</i>	2.10	2.61	2.38	0.11
	<i>b</i>	-0.11	0.04	-0.03	0.03
	<i>c</i>	0.90	0.96	0.93	0.01
IV	<i>a</i>	-0.29	-0.07	-0.19	0.04
	<i>b</i>	2.42	2.81	2.62	0.08
	<i>c</i>	0.01	0.02	0.02	0.00

Table A.7 – Rate-quantization model parameters for the Football sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	1.98	34.57	3.67	4.39
	<i>b</i>	3.77	34.89	5.32	4.15
	<i>c</i>	0.03	0.29	0.22	0.05
II	<i>a</i>	1.18	8.06	5.96	1.57
	<i>b</i>	-0.87	0.018	-0.48	0.24
	<i>c</i>	0.56	1.00	0.67	0.09
III	<i>a</i>	1.13	16.56	11.45	3.85
	<i>b</i>	-0.05	1.57	1.055	0.31
	<i>c</i>	0.91	1.30	1.10	0.06
IV	<i>a</i>	-5.51	-0.05	-3.43	1.42
	<i>b</i>	1.22	12.58	8.86	2.70
	<i>c</i>	-0.13	0.037	-0.03	0.036

Table A.8 – Rate-quantization model parameters for the Football sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	3.04	53.26	38.73	21.13
	<i>b</i>	4.80	54.20	40.14	21.07
	<i>c</i>	0.02	0.20	0.05	0.06
II	<i>a</i>	1.22	6.36	4.47	1.29
	<i>b</i>	-0.49	0.06	-0.22	0.17
	<i>c</i>	0.64	1.61	0.78	0.15
III	<i>a</i>	0.27	11.64	6.93	3.16
	<i>b</i>	-0.79	1.24	0.54	0.41
	<i>c</i>	0.52	1.24	1.02	0.10
IV	<i>a</i>	-3.68	0.86	-1.85	1.18
	<i>b</i>	0.38	9.48	6.07	2.29
	<i>c</i>	-0.11	0.05	-0.00	0.03

A.2 Rate-Quantization Model Parameters with a Reduced Number of Model Parameters

Table A.9 – Rate-quantization model parameters for the News sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	2.88	2.93	2.91	0.01
	b	4.37	4.45	4.41	0.02
	$c = 0.2$	-	-	-	-
II	a	4.68	5.04	4.91	0.09
	b	-0.40	-0.36	-0.38	0.01
	$c = 0.6$	-	-	-	-
III	a	8.16	8.98	8.68	0.18
	b	0.87	0.95	0.92	0.01
	$c = 1.0$	-	-	-	-
II, III	a	4.48	4.81	4.70	0.08
	$b = 0$	-	-	-	-
	c	0.76	0.77	0.76	0.00
IV	a	-3.49	-3.07	-3.34	0.09
	b	7.41	8.10	7.86	0.16
	$c = 0$	-	-	-	-

Table A.10 – Rate-quantization model parameters for the News sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	7.08	7.24	7.15	0.03
	b	8.30	8.43	8.36	0.03
	$c = 0.1$	-	-	-	-
II	a	3.30	3.50	3.42	0.05
	b	-0.02	-0.01	-0.01	0.00
	$c = 0.8$	-	-	-	-
III	a	4.99	5.53	5.31	0.13
	b	0.53	0.62	0.58	0.02
	$c = 1.0$	-	-	-	-
II, III	a	3.31	3.49	3.42	0.04
	$b = 0$	-	-	-	-
	c	0.81	0.82	0.81	0.00
IV	a	-1.74	-1.44	-1.61	0.07
	b	4.72	5.17	4.99	0.11
	$c = 0$	-	-	-	-

Table A.11 – Rate-quantization model parameters for the Kayak sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	2.84	3.29	3.08	0.13
	b	4.56	4.97	4.75	0.10
	$c = 0.2$	-	-	-	-
II	a	5.18	7.10	5.92	0.44
	b	-0.66	-0.47	-0.56	0.05
	$c = 0.6$	-	-	-	-
III	a	7.34	13.25	9.56	1.36
	b	0.56	1.09	0.78	0.15
	$c = 1.0$	-	-	-	-
II, III	a	4.88	6.68	5.57	0.41
	$b = 0$	-	-	-	-
	c	0.75	0.85	0.80	0.03
IV	a	-5.54	-2.41	-3.60	0.74
	b	7.09	11.84	8.91	1.07
	$c = 0$	-	-	-	-

Table A.12 – Rate-quantization model parameters for the Kayak sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	6.56	7.27	7.00	0.17
	b	8.30	8.85	8.57	0.15
	$c = 0.1$	-	-	-	-
II	a	4.24	6.41	5.04	0.42
	b	-0.09	0.05	-0.03	0.03
	$c = 0.8$	-	-	-	-
III	a	6.89	12.13	8.40	1.10
	b	0.63	1.04	0.76	0.10
	$c = 1.0$	-	-	-	-
II, III	a	4.16	6.27	4.93	0.41
	$b = 0$	-	-	-	-
	c	0.75	0.83	0.80	0.02
IV	a	-4.95	-2.40	-3.08	0.56
	b	6.43	10.90	7.81	0.91
	$c = 0$	-	-	-	-

Table A.13 – Rate-quantization model parameters for the M&D sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	3.41	3.48	3.45	0.02
	b	4.57	4.69	4.63	0.03
	$c = 0.2$	-	-	-	-
II	a	3.42	3.73	3.54	0.08
	b	-0.42	-0.38	-0.40	0.01
	$c = 0.6$	-	-	-	-
III	a	4.42	4.83	4.57	0.11
	b	0.35	0.40	0.38	0.01
	$c = 1.0$	-	-	-	-
II, III	a	3.27	3.57	3.38	0.07
	$b = 0$	-	-	-	-
	c	0.87	0.89	0.88	0.00
IV	a	-1.27	-1.07	-1.17	0.05
	b	4.34	4.73	4.47	0.10
	$c = 0$	-	-	-	-

Table A.14 – Rate-quantization model parameters for the M&D sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	8.27	8.46	8.37	0.04
	b	9.17	9.33	9.25	0.03
	$c = 0.1$	-	-	-	-
II	a	2.34	2.55	2.44	0.05
	b	-0.09	-0.08	-0.09	0.00
	$c = 0.8$	-	-	-	-
III	a	2.61	2.95	2.77	0.07
	b	0.11	0.17	0.14	0.01
	$c = 1.0$	-	-	-	-
II, III	a	2.35	2.55	2.44	0.04
	$b = 0$	-	-	-	-
	c	0.93	0.95	0.94	0.00
IV	a	-0.40	-0.24	-0.32	0.03
	b	2.60	2.93	2.76	0.07
	$c = 0$	-	-	-	-

Table A.15 – Rate-quantization model parameters for the Football sequence [QCIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	2.95	3.96	3.22	0.21
	b	3.87	5.37	4.88	0.22
	$c = 0.2$	-	-	-	-
II	a	1.19	7.97	6.00	1.53
	b	-0.75	-0.14	-0.60	0.13
	$c = 0.6$	-	-	-	-
III	a	1.36	13.70	9.11	3.03
	b	0.07	0.93	0.63	0.21
	$c = 1.0$	-	-	-	-
II, III	a	1.19	7.48	5.66	1.41
	$b = 0$	-	-	-	-
	c	0.77	0.99	0.83	0.05
IV	a	-5.54	-0.16	-3.22	1.46
	b	1.35	12.61	8.63	2.72
	$c = 0$	-	-	-	-

Table A.16 – Rate-quantization model parameters for the Football sequence [CIF] with a reduced number of model parameters

MODEL	PARAM	MIN	MAX	MEAN	STD
I	a	7.08	10.91	7.75	0.65
	b	8.68	11.11	9.15	0.40
	$c = 0.1$	-	-	-	-
II	a	1.13	6.21	4.42	1.25
	b	-0.21	-0.05	-0.10	0.04
	$c = 0.8$	-	-	-	-
III	a	0.79	10.09	6.26	2.35
	b	-0.38	0.73	0.43	0.23
	$c = 1.0$	-	-	-	-
II, III	a	1.23	6.06	4.34	1.19
	$b = 0$	-	-	-	-
	c	0.80	1.18	0.87	0.07
IV	a	-3.69	0.49	-1.81	1.11
	b	0.78	9.49	6.03	2.19
	$c = 0$	-	-	-	-

A.3 Distortion-Quantization Model Parameters

Table A.17 – Distortion-quantization model parameters for the News sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	4.14	10.46	9.93	0.56
	<i>b</i>	-9.94	-3.07	-9.33	0.61
	<i>c</i>	0.10	0.19	0.11	0.01
II	<i>a</i>	367.50	876.14	611.06	190.52
	$b \times 10^{-3}$	0.83	1.92	1.31	0.34
	<i>c</i>	1.54	1.64	1.58	0.02
III	<i>a</i>	0.92	1.45	1.20	0.10
	<i>b</i>	-2.91	-0.76	-2.04	0.41
	<i>c</i>	1.35	1.48	1.40	0.02
IV	<i>a</i>	0.06	0.08	0.07	0.00
	<i>b</i>	2.35	3.02	2.73	0.13
	<i>c</i>	-6.41	-4.12	-5.50	0.45

Table A.18 – Distortion-quantization model parameters for the News sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	3.08	11.16	10.83	0.48
	<i>b</i>	-10.96	-2.27	-10.64	0.52
	<i>c</i>	0.09	0.23	0.10	0.01
II	<i>a</i>	270.75	425.60	399.04	26.43
	$b \times 10^{-3}$	1.27	2.01	1.45	0.10
	<i>c</i>	1.46	1.55	1.50	0.01
III	<i>a</i>	0.70	0.91	0.79	0.04
	<i>b</i>	-1.18	-0.34	-0.72	0.17
	<i>c</i>	1.33	1.42	1.38	0.01
IV	<i>a</i>	0.04	0.05	0.04	0.00
	<i>b</i>	1.64	1.87	1.74	0.05
	<i>c</i>	-3.32	-2.43	-2.82	0.18

Table A.19 – Distortion-quantization model parameters for the Kayak sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.86	11.92	11.11	0.97
	<i>b</i>	-10.56	1.40	-9.92	1.00
	<i>c</i>	0.08	0.43	0.09	0.03
II	<i>a</i>	166.99	524.62	276.23	70.42
	$b \times 10^{-3}$	1.83	6.33	3.65	1.18
	<i>c</i>	1.51	1.82	1.65	0.07
III	<i>a</i>	1.96	5.68	3.37	0.82
	<i>b</i>	-11.31	-4.91	-7.65	1.27
	<i>c</i>	0.97	1.37	1.17	0.10
IV	<i>a</i>	-0.01	0.11	0.04	0.03
	<i>b</i>	3.92	5.90	4.91	0.42
	<i>c</i>	-15.31	-8.19	-11.20	1.33

Table A.20 – Distortion-quantization model parameters for the Kayak sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	4.11	11.32	10.53	0.62
	<i>b</i>	-10.08	-2.95	-9.52	0.56
	<i>c</i>	0.08	0.19	0.09	0.01
II	<i>a</i>	190.36	437.21	265.17	54.34
	$b \times 10^{-3}$	2.02	4.70	3.27	0.71
	<i>c</i>	1.54	1.74	1.63	0.04
III	<i>a</i>	1.49	3.99	2.56	0.57
	<i>b</i>	-8.61	-2.52	-5.70	1.33
	<i>c</i>	1.07	1.35	1.21	0.07
IV	<i>a</i>	0.01	0.09	0.04	0.02
	<i>b</i>	2.82	5.39	4.08	0.51
	<i>c</i>	-13.73	-5.36	-9.08	1.44

Table A.21 – Distortion-quantization model parameters for the M&D sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.88	11.27	9.36	3.17
	<i>b</i>	-10.55	-1.65	-8.53	3.35
	<i>c</i>	0.08	0.22	0.11	0.05
II	<i>a</i>	153.56	399.91	244.71	74.46
	$b \times 10^{-3}$	3.03	6.35	4.65	0.81
	<i>c</i>	1.22	1.42	1.32	0.05
III	<i>a</i>	1.52	2.52	2.02	0.18
	<i>b</i>	-3.86	-1.61	-2.80	0.41
	<i>c</i>	1.03	1.16	1.08	0.03
IV	<i>a</i>	0.00	0.02	0.01	0.00
	<i>b</i>	2.17	2.77	2.47	0.10
	<i>c</i>	-4.61	-2.87	-3.74	0.32

Table A.22 – Distortion-quantization model parameters for the M&D sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	2.05	11.59	9.00	3.80
	<i>b</i>	-11.24	-1.08	-8.49	4.02
	<i>c</i>	0.08	0.26	0.12	0.06
II	<i>a</i>	100.61	398.68	202.53	119.94
	$b \times 10^{-3}$	2.33	7.93	5.14	1.65
	<i>c</i>	1.11	1.34	1.24	0.06
III	<i>a</i>	1.10	1.78	1.43	0.12
	<i>b</i>	-2.23	-0.78	-1.54	0.24
	<i>c</i>	0.97	1.11	1.04	0.02
IV	<i>a</i>	-0.00	0.01	0.00	0.00
	<i>b</i>	1.38	1.73	1.58	0.06
	<i>c</i>	-2.34	-1.33	-1.89	0.18

A.4 Rate-Distortion Model Parameters

Table A.23 – Distortion-quantization model parameters for the Football sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	1.02	22.64	17.38	4.56
	<i>b</i>	-21.46	1.43	-16.31	4.61
	<i>c</i>	0.04	0.41	0.06	0.03
II	<i>a</i>	26.07	494.17	281.71	110.33
	$b \times 10^{-3}$	2.53	19.91	5.35	3.16
	<i>c</i>	1.14	1.70	1.49	0.12
III	<i>a</i>	1.13	7.77	3.44	1.13
	<i>b</i>	-12.18	-1.00	-6.48	2.32
	<i>c</i>	0.68	1.28	1.10	0.12
IV	<i>a</i>	-0.03	0.08	0.02	0.03
	<i>b</i>	0.83	6.32	4.45	1.34
	<i>c</i>	-15.19	-0.59	-8.84	3.57

Table A.24 – Distortion-quantization model parameters for the Football sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.87	26.72	20.90	6.65
	<i>b</i>	-25.74	1.41	-20.14	6.68
	<i>c</i>	0.03	0.43	0.06	0.03
II	<i>a</i>	59.60	339.45	211.97	65.19
	$b \times 10^{-3}$	2.42	24.01	5.33	2.73
	<i>c</i>	0.81	1.60	1.42	0.15
III	<i>a</i>	0.75	7.36	2.70	1.33
	<i>b</i>	-12.79	-0.24	-4.63	2.80
	<i>c</i>	0.62	1.28	1.08	0.13
IV	<i>a</i>	-0.03	0.04	0.01	0.02
	<i>b</i>	0.67	5.50	3.21	1.28
	<i>c</i>	-11.62	-0.14	-5.94	3.23

Table A.25 – Rate-distortion model parameters for the News sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.47	0.49	0.48	0.00
	<i>b</i>	0.40	0.57	0.50	0.04
	<i>c</i>	1.09	1.12	1.10	0.01
	<i>d</i>	1.79	1.98	1.88	0.05
II	<i>a</i>	3.63	4.14	3.90	0.12
	<i>b</i>	-0.90	-0.58	-0.70	0.05
	<i>c</i>	0.26	0.30	0.28	0.01
	<i>d</i>	0.34	0.47	0.40	0.02
III	<i>a</i>	7.38	8.55	8.09	0.31
	<i>b</i>	1.71	1.89	1.80	0.05
	<i>c</i>	0.63	0.65	0.64	0.01
	<i>d</i>	0.74	0.79	0.77	0.01
IV	<i>a</i>	-2.80	-2.39	-2.62	0.10
	<i>b</i>	6.14	6.85	6.58	0.19
	<i>c</i>	0.40	0.45	0.43	0.01

Table A.26 – Rate-distortion model parameters for the News sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.64	0.68	0.65	0.01
	<i>b</i>	1.12	1.24	1.16	0.02
	<i>c</i>	0.89	0.92	0.91	0.01
	<i>d</i>	-0.30	-0.18	-0.21	0.02
II	<i>a</i>	2.03	2.28	2.17	0.06
	<i>b</i>	-0.39	-0.25	-0.32	0.03
	<i>c</i>	0.30	0.35	0.32	0.01
	<i>d</i>	0.75	0.77	0.76	0.00
III	<i>a</i>	3.51	4.17	3.90	0.16
	<i>b</i>	1.07	1.26	1.19	0.05
	<i>c</i>	0.62	0.66	0.64	0.01
	<i>d</i>	0.82	0.92	0.88	0.02
IV	<i>a</i>	-1.13	-0.94	-1.05	0.04
	<i>b</i>	3.54	3.89	3.75	0.08
	<i>c</i>	0.23	0.26	0.25	0.01

Table A.27 – Rate-distortion model parameters for the Kayak sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.44	0.55	0.50	0.03
	<i>b</i>	0.60	0.97	0.79	0.08
	<i>c</i>	1.14	1.18	1.16	0.01
	<i>d</i>	1.62	2.21	1.91	0.15
II	<i>a</i>	4.75	8.87	6.24	0.80
	<i>b</i>	-4.42	-1.62	-2.59	0.49
	<i>c</i>	0.11	0.20	0.16	0.01
	<i>d</i>	0.84	0.93	0.89	0.02
III	<i>a</i>	10.66	17.18	13.71	1.67
	<i>b</i>	2.28	3.13	2.69	0.21
	<i>c</i>	0.64	0.80	0.72	0.04
	<i>d</i>	0.78	0.97	0.88	0.04
IV	<i>a</i>	-6.35	-3.58	-4.78	0.67
	<i>b</i>	8.01	12.38	9.86	1.05
	<i>c</i>	0.30	0.68	0.44	0.09

Table A.28 – Rate-distortion model parameters for the Kayak sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.45	0.55	0.51	0.02
	<i>b</i>	0.35	0.88	0.58	0.11
	<i>c</i>	1.12	1.16	1.14	0.01
	<i>d</i>	1.75	2.15	1.93	0.07
II	<i>a</i>	3.51	6.74	4.64	0.62
	<i>b</i>	-2.24	-0.62	-1.16	0.30
	<i>c</i>	0.18	0.31	0.25	0.02
	<i>d</i>	0.37	0.48	0.42	0.02
III	<i>a</i>	7.41	14.76	10.25	1.58
	<i>b</i>	1.85	2.62	2.22	0.17
	<i>c</i>	0.63	0.75	0.70	0.02
	<i>d</i>	0.79	0.90	0.84	0.02
IV	<i>a</i>	-5.53	-2.52	-3.63	0.60
	<i>b</i>	6.15	11.18	8.02	0.99
	<i>c</i>	0.31	0.62	0.39	0.07

Table A.29 – Rate-distortion model parameters for the M&D sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.60	0.63	0.62	0.01
	<i>b</i>	0.18	0.35	0.29	0.03
	<i>c</i>	1.12	1.14	1.13	0.00
	<i>d</i>	1.23	1.46	1.31	0.05
II	<i>a</i>	2.56	2.95	2.72	0.09
	<i>b</i>	-0.48	-0.30	-0.37	0.04
	<i>c</i>	0.36	0.43	0.40	0.01
	<i>d</i>	0.18	0.31	0.25	0.02
III	<i>a</i>	4.96	5.91	5.34	0.23
	<i>b</i>	1.64	1.86	1.73	0.05
	<i>c</i>	0.78	0.81	0.79	0.01
	<i>d</i>	0.76	0.84	0.80	0.02
IV	<i>a</i>	-1.82	-1.42	-1.57	0.09
	<i>b</i>	4.14	4.88	4.41	0.17
	<i>c</i>	0.15	0.17	0.16	0.01

Table A.30 – Rate-distortion model parameters for the M&D sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.74	0.81	0.77	0.01
	<i>b</i>	0.75	0.84	0.80	0.02
	<i>c</i>	0.93	0.97	0.95	0.01
	<i>d</i>	-0.29	-0.20	-0.24	0.02
II	<i>a</i>	1.29	1.52	1.42	0.05
	<i>b</i>	-0.29	-0.20	-0.25	0.02
	<i>c</i>	0.36	0.40	0.37	0.01
	<i>d</i>	0.71	0.74	0.72	0.01
III	<i>a</i>	2.18	2.69	2.42	0.11
	<i>b</i>	1.00	1.15	1.07	0.03
	<i>c</i>	0.78	0.81	0.79	0.01
	<i>d</i>	0.74	0.79	0.77	0.01
IV	<i>a</i>	-0.56	-0.34	-0.45	0.05
	<i>b</i>	2.07	2.51	2.29	0.10
	<i>c</i>	0.08	0.09	0.09	0.00

Table A.31 – Rate-distortion model parameters for the Football sequence [QCIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.45	0.87	0.55	0.08
	<i>b</i>	-1.35	1.19	0.81	0.42
	<i>c</i>	1.02	1.19	1.16	0.03
	<i>d</i>	0.80	2.02	1.69	0.24
II	<i>a</i>	0.51	8.33	5.74	1.84
	<i>b</i>	-2.52	-0.02	-1.28	0.64
	<i>c</i>	0.20	0.69	0.30	0.09
	<i>d</i>	-0.21	0.42	0.21	0.11
III	<i>a</i>	0.68	21.64	14.63	5.68
	<i>b</i>	0.63	3.55	2.72	0.62
	<i>c</i>	0.68	0.97	0.77	0.06
	<i>d</i>	0.58	1.21	0.90	0.09
IV	<i>a</i>	-7.80	0.02	-5.01	2.09
	<i>b</i>	0.59	14.66	10.21	3.52
	<i>c</i>	0.03	0.69	0.40	0.18

Table A.32 – Rate-distortion model parameters for the Football sequence [CIF]

MODEL	PARAM	MIN	MAX	MEAN	STD
I	<i>a</i>	0.50	1.34	0.62	0.13
	<i>b</i>	-1.96	0.89	0.33	0.56
	<i>c</i>	1.03	1.17	1.12	0.04
	<i>d</i>	1.07	2.21	1.65	0.20
II	<i>a</i>	0.26	6.06	3.76	1.51
	<i>b</i>	-1.80	0.04	-0.70	0.51
	<i>c</i>	0.22	0.86	0.37	0.12
	<i>d</i>	0.00	0.53	0.32	0.07
III	<i>a</i>	0.28	14.82	8.15	4.11
	<i>b</i>	0.01	2.71	1.89	0.60
	<i>c</i>	0.60	0.97	0.76	0.05
	<i>d</i>	0.67	1.13	0.84	0.06
IV	<i>a</i>	-5.13	0.54	-2.65	1.59
	<i>b</i>	-0.01	10.66	6.44	2.82
	<i>c</i>	0.02	0.48	0.26	0.13

References

- [1] M. J. Riley, I. E. G. Richardson, “Digital Video Communications”, Artech House, 1996.
- [2] ITU-R Recommendation BT.601-5, “Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect Ratios”, 1998.
- [3] SMPTE 295M-1997, “Television – 1920×1080 50Hz – Scanning and Interface”, 1997.
- [4] SMPTE 296M-2001, “Television – 1280×720 Progressive Image Sample Structure – Analog and Digital Representation and Analog Interface”, 2001.
- [5] Y. Wang, J. Ostermann, Y.-Q. Zhang, “Video Processing and Communications”. Prentice Hall, 2002.
- [6] ITU-R Recommendation BT.470-4, “Television Systems,” 1995.
- [7] ITU-T Recommendation H.261 (1993), “Video Codec for Audiovisual Services at p×64 kbit/s”, 1993.
- [8] ITU-T Recommendation H.263 (1996), “Video Coding for Low Bitrate Communication”, 1996.
- [9] ISO/IEC 11172-2:1993, “Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1,5 Mbit/s – Part 2: Video”, 1993.
- [10] ISO/IEC 13818-2:1996, “Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video”, 1996.
- [11] D. Marpe, T. Wiegand, G. Sullivan, “The H.264/MPEG-4 Advanced Video Coding Standard and its Applications”, *IEEE Communications Magazine*, vol. 44, no. 8, pp. 134–143, August 2006

- [12] B. Girod, A. Aaron, S. Rane, D. Rebollo-Monedero, "Distributed Video Coding", *Proceedings of the IEEE*, vol. 93, no. 1, pp. 71–83, January 2005.
- [13] P. Nunes, F. Pereira, "Rate Control in Object-based Video Coding Frameworks", Doc. ISO/IEC JTC1/SC29/WG11 M3593, Dublin MPEG meeting, July 1998.
- [14] P. Nunes, F. Pereira, "Rate Control for Scenes with Multiple Arbitrarily Shaped Video Objects", *Proceedings of the 1997 Picture Coding Symposium (PCS'97)*, pp. 303–308, Berlin, Germany, September 1997.
- [15] F. Pereira, P. Nunes, "Levels for Visual Profiles", Chapter in *The MPEG-4 Book*, F. Pereira, T. Ebrahimi (Eds.), Prentice Hall 2002.
- [16] P. Nunes, F. Pereira, "MPEG-4 Compliant Video Encoding: Analysis and Rate Control Strategies", *Proceedings of the 34th ASILOMAR Conference*, Pacific Grove, CA, USA, October 2000.
- [17] P. Nunes, F. Pereira, "Object-Based Rate Control for the MPEG-4 Visual Simple Profile", *Proceedings of the 1999 Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'99)*, Berlin, Germany, May 1999.
- [18] P. Nunes, F. Pereira, "Implementing the MPEG-4 Natural Visual Profiles and Levels", Doc. ISO/IEC JTC1/SC29/WG11 M4878, Vancouver MPEG Meeting, July 1999.
- [19] P. Nunes, F. Pereira, "Implementing the MPEG-4 Video Buffering Verifier for the Core Profile", Doc. ISO/IEC JTC1/SC29/WG11 M5168, Melbourne MPEG Meeting, October 1999.
- [20] J. Valentim, P. Nunes, F. Pereira, "Evaluating MPEG-4 Video Complexity for an Alternative Video Verifier Mechanism Complexity Model", Doc. ISO/IEC JTC1/SC29/WG11 M7028, Singapore MPEG Meeting, March 2001.
- [21] J. Valentim, P. Nunes, F. Pereira, "Evaluating MPEG-4 Video Decoding Complexity", *Proceedings of the 2nd Workshop and Exhibition on MPEG-4 (WEMP'01)*, San Jose, CA, USA, June 2001.
- [22] J. Valentim, P. Nunes, F. Pereira, "An Alternative Complexity Model for the MPEG-4 Video Verifier Mechanism", *Proceedings of the 2001 International Conference on Image Processing (ICIP'01)*, Thessaloniki, Greece, October 2001.
- [23] J. Valentim, P. Nunes, F. Pereira, "Evaluating MPEG-4 Video Decoding Complexity for an Alternative Video Complexity Verifier Model", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 1034–1044, November 2002.
- [24] P. Nunes, F. Pereira, "Rate and Distortion Modeling Analysis for MPEG-4 Video Intra Coding", *Proceedings of the 2004 Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'04)*, Lisboa, Portugal, April 2004.
- [25] P. Nunes, F. Pereira, "Rate and Distortion Models for MPEG-4 Video Encoding", *Proceedings of SPIE 49th Annual Meeting: Applications and Digital Image Processing XXVII*, Denver, CO, USA, vol. 5558, August 2004.

References

- [26] P. Nunes, F. Pereira, "Scene Level Rate Control Algorithm for MPEG-4 Video Encoding", *Proceedings of 2001 SPIE Visual Communications and Image Processing (VCIP'01)*, San Jose, CA, USA, vol. 4310, pp. 194–205, January 2001.
- [27] J. Valentim, P. Nunes, L. Soares, F. Pereira, "IST MPEG-4 Video Compliant Framework". Doc. ISO/IEC JTC1/SC29/WG11 MPEG2000/M5844, Noordwijkerhout MPEG meeting, March 2000.
- [28] ISO/IEC 14496-1:2004, "Information Technology – Coding of Audio-Visual Objects – Part 1: Systems (3rd Ed.)", 2004.
- [29] ISO/IEC 14496-2:2004, "Information Technology – Coding of Audio-Visual Objects – Part 2: Visual (3rd Ed.)", 2004.
- [30] ISO/IEC 14496-3:2005, "Information Technology – Coding of Audio-Visual Objects – Part 3: Audio (3rd Ed.)", 2005.
- [31] ISO/IEC 14496-4:2004, "Information Technology – Coding of Audio-Visual Objects – Part 4: Conformance Testing (2nd Ed.)", 2004.
- [32] ISO/IEC 14496-5:2001, "Information Technology – Coding of Audio-Visual Objects – Part 5: Reference Software (2nd Ed.)", 2001.
- [33] ISO/IEC 14496-6:2000, "Information Technology – Coding of Audio-Visual Objects – Part 6: Delivery Multimedia Integration Framework (DMIF) (2nd Ed.)", 2000.
- [34] ISO/IEC 14496-7:2004, "Information Technology – Coding of Audio-Visual Objects – Part 7: Optimized Reference Software for Coding of Audio-Visual Objects (2nd Ed.)", 2004.
- [35] ISO/IEC 14496-8:2004, "Information Technology – Coding of Audio-Visual Objects – Part 8: Carriage of ISO/IEC 14496 Contents over IP Networks (1st Ed.)", 2004.
- [36] ISO/IEC 14496-9:2004, "Information Technology – Coding of Audio-Visual Objects – Part 9: Reference Hardware Description (1st Ed.)", 2004.
- [37] ISO/IEC 14496-10:2005, "Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding (3rd Ed.)", 2005.
- [38] ISO/IEC 14496-11:2005, "Information Technology – Coding of Audio-Visual Objects – Part 11: Scene Description and Application Engine (1st Ed.)", 2005.
- [39] ISO/IEC 14496-12:2005, "Information Technology – Coding of Audio-Visual Objects – Part 12: ISO Base Media File Format (2nd Ed.)", 2005.
- [40] ISO/IEC 14496-13:2004, "Information Technology – Coding of Audio-Visual Objects – Part 13: Intellectual Property Management and Protection (IPMP) Extensions (1st Ed.)", 2004.
- [41] ISO/IEC 14496-14:2003, "Information Technology – Coding of Audio-Visual Objects – Part 14: MP4 File Format (1st Ed.)", 2003.

- [42] ISO/IEC 14496-15:2004, “Information Technology – Coding of Audio-Visual Objects – Part 15: Advanced Video Coding (AVC) File Format (1st Ed.)”, 2004.
- [43] ISO/IEC 14496-16:2004, “Information Technology – Coding of Audio-Visual Objects – Part 16: Animation Framework eXtension (AFX) (1st Ed.)”, 2004.
- [44] ISO/IEC 14496-17:2006, “Information Technology – Coding of Audio-Visual Objects – Part 17: Streaming Text Format (1st Ed.)”, 2006.
- [45] ISO/IEC 14496-18:2004, “Information Technology – Coding of Audio-Visual Objects – Part 18: Font Compression and Streaming (1st Ed.)”, 2004.
- [46] ISO/IEC 14496-19:2004, “Information Technology – Coding of Audio-Visual Objects – Part 19: Synthesized Texture Stream (1st Ed.)”, 2004.
- [47] ISO/IEC 14496-20:2006, “Information Technology – Coding of Audio-Visual Objects – Part 20: Lightweight Application Scene Representation (LASER) and Simple Aggregation Format (SAF) (1st Ed.)”, 2006.
- [48] ISO/IEC FDIS 14496-21:2006, “Information Technology – Coding of Audio-Visual Objects – Part 21: MPEG-J Graphics Framework eXtensions (GFX) (1st Ed.)”, 2006.
- [49] ISO/IEC FCD 14496-22:2006, “Information Technology – Coding of Audio-Visual Objects – Part 22: Open Font Format (1st Ed.)”, 2006.
- [50] F. Pereira, “MPEG-4: Why, What, How and When?”, *Signal Processing: Image Communication*, vol. 15, no. 4–5, pp. 271–279, January 2000.
- [51] F. Pereira, T. Ebrahimi, “The MPEG-4 Book”, Prentice Hall, 2002.
- [52] MPEG Convener, “MPEG-4 Project Description”, Doc. ISO/IEC JTC1/SC29/WG11 N1177, Munich MPEG meeting, January 1996.
- [53] MPEG Requirements, “MPEG-4 Requirements”, Doc. ISO/IEC JTC1/SC29/WG11 N5866, Trondheim MPEG meeting, July 2003.
- [54] MPEG Requirements, “MPEG-4 Overview”, Doc. ISO/IEC JTC1/SC29/WG11 N4668, Jeju MPEG meeting, March 2002.
- [55] ISO/IEC 14772-1:1997, “The Virtual Reality Modeling Language”, 1997.
- [56] MPEG Applications and Operational Environments subgroup, “Proposal Package Description (PPD) – Revision 3”, Doc. ISO/IEC JTC1/SC29/WG11 N998, Tokyo MPEG meeting, July 1995.
- [57] MPEG Video “Introduction to Multi-view Video Coding”, Doc. ISO/IEC JTC1/SC29/WG11 N7328, Poznan MPEG meeting, July 2005.
- [58] MPEG Requirements, “MPEG-4 Applications”, Doc. ISO/IEC JTC1/SC29/WG11 N2724, Seoul MPEG meeting, March 1999.
- [59] MPEG Industry Forum, <http://www.mpegif.org>.

References

- [60] Third Generation Partnership Project, <http://www.3gpp.org>.
- [61] ITU-T Recommendation H.264 (2005), “Advanced Video Coding for Generic Audiovisual Services”, 2005.
- [62] ISO/IEC 15444-12:2005, “Information Technology – JPEG 2000 Image Coding System – Part 12: ISO Base Media File Format”, 2005.
- [63] ISO/IEC 15444-3:2002, “Information Technology – JPEG 2000 Image Coding System – Part 3: Motion JPEG 2000”, 2002.
- [64] MPEG Systems, “ISO/IEC 14496-22 – Open Font Format”, Doc. ISO/IEC JTC1/SC29/WG11 N7519, Poznan MPEG meeting, July 2005.
- [65] P. Correia, “Video Analysis for Object-based Coding and Description”, Ph.D. Thesis, Instituto Superior Técnico, Portugal, December 2002.
- [66] J. Ostermann, “Coding of Binary Shape in MPEG-4”, *Proceedings of the 1997 Picture Coding Symposium (PCS'97)*, pp. 659–662, Berlin, Germany, September 1997.
- [67] MPEG Video, “Core Experiments on MPEG-4 Video Shape Coding”, Doc. ISO/IEC JTC1/SC29/WG11 N1471, Maceió MPEG meeting, November 1996.
- [68] N. Yamaguchi, T. Ida, T. Watanabe, “A Binary Shape Coding Method Using Modified MMR”, *Proceedings of the 1997 IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, CA, USA, October 1997.
- [69] N. Brady, F. Bossen, N. Murphy, “Context-based Arithmetic Encoding of 2D Shape Sequences”, *Proceedings of the 1997 IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, CA, USA, October 1997.
- [70] P. Gerken, “Object-based Analysis-Synthesis Coding of Image Sequences at Very Low Bit Rates”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 4, no. 3, pp. 228–235, June 1994.
- [71] S. Lee, D.-S. Cho, S. Cho, S. Son, E. Jang, J.-S. Shin, “Binary Shape Coding Using 1-D Distance Values from Baseline”, *Proceedings of the 1997 IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, CA, USA, October 1997.
- [72] P. Nunes, F. Pereira, F. Marqués, “Multi-Grid Chain Coding of Binary Shapes”, *Proceedings of the 1997 IEEE International Conference on Image Processing (ICIP'97)*, Santa Barbara, CA, USA, October 1997.
- [73] MPEG Video, “MPEG-4 Video Verification Model 19.0”, Doc. ISO/IEC JTC1/SC29/WG11 N6184, Hawaii MPEG meeting, December 2003.
- [74] C. K. Chui, “An Introduction to Wavelets”, Academic Press, 1992.
- [75] ISO/IEC 15444-1:2004, “Information Technology – JPEG 2000 Image Coding System: Core Coding System”, 2004.
- [76] L. Soares, “Error Resilience for Object-based Video Coding”, Ph.D. Thesis, Instituto

Superior Técnico, Portugal, April 2004.

- [77] A. Vetro, H. Sun, Y. Wang, “MPEG-4 Rate Control for Multiple Video Objects”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 186–199, February 1999.
- [78] J. Ronda, M. Eckert, F. Jaureguizar, N. García, “Rate Control and Bit Allocation for MPEG-4”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1243–1258, December 1999.
- [79] MPEG Requirements, “MPEG-4 Profiling Policy”, Doc. ISO/IEC JTC1/SC29/WG11 N2199, Tokyo MPEG meeting, March 1998.
- [80] MPEG Industry Forum, “MPEG-4 – The Media Standard”, November 2002.
- [81] T. Sikora, T. Ebrahimi, “Tutorial on MPEG-4 Visual”, Workshop and Exhibition on MPEG-4 (WEMP’01), San Jose, CA, USA, June 2001.
- [82] D. Wood, “Everything You Wanted To Know About Video Codecs – But Were Afraid to Ask”, EBU Technical Review, July 2003.
- [83] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, G. Sullivan, “Rate-Constrained Coder Control and Comparison of Video Coding Standards”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 688–703, July 2003.
- [84] T. Cover, J. Thomas, “Elements of Information Theory”. Wiley series in telecommunications, John Wiley & Sons, Inc., New York, 1991.
- [85] S. Liew, C.-Y. Tse, “A Control-Theoretic Approach to Rate-Controlled Video Compression”, *Proceedings of the 1996 IEEE International Conference on Image Processing (ICIP’96)*, Lausanne, Switzerland, September 1996.
- [86] Y. Shoham, A. Gersho, “Efficient Bit Allocation for an Arbitrary Set of Quantizers”, *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 9, pp. 1445–1453, September 1988.
- [87] S.-W. Wu, A. Gersho, “Rate-Constrained Optimal Block-Adaptive Coding for Digital Tape Recording of HDTV”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, no. 1, pp. 100–112, March 1991.
- [88] A. Reibman, B. Haskell, “Constraints on Variable Bit-Rate Video for ATM Networks”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 4, pp. 361–372, December 1992.
- [89] W. Verbiest, L. Pinnoo, B. Voeten, “The Impact of the ATM Concept on Video Coding”, *IEEE Journal on Selected Areas in Communications*, vol. 6, no. 9, pp. 1623–1632, December 1988.
- [90] A. Ortega, “Video Transmission over ATM Networks”. In B. Sheu et al., editors, *Microsystems Technology for Multimedia Applications*, IEEE Press, May 1995.
- [91] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for

References

- Real-Time Applications”, Audio-Video Transport Working Group RFC 1889, January 1996.
- [92] M. Gormish, J. Gill, “Computation-Rate-Distortion in Transform Coders for Image Compression”, *Proceedings of SPIE: Image and Video Processing*, vol. 1903, pp. 146–152, San Jose, CA, USA, April 1993.
- [93] I. Richardson, Y. Zhao, “Adaptive Algorithms for Variable-Complexity Video Coding”, *Proceedings of the 2001 International Conference on Image Processing (ICIP’01)*, vol. 1, pp. 457–460, Thessaloniki, Greece, October 2001.
- [94] Y. Zhao, I. Richardson, “Complexity Management of Video Encoders”, *Proceedings of the 10th ACM International Conference on Multimedia*, pp. 647–649, Juan-les-Pins, France, December 2002.
- [95] MPEG Test, “MPEG-4 Test/Evaluation Procedures”, Doc. ISO/IEC JTC1/SC29/WG11 N0999, Tokyo MPEG Meeting, July 1995.
- [96] G. Bjontegaard, K. Lillevold, “H.263 Anchors – Technical Description”, Doc. ISO/IEC JTC1/SC29/WG11 M0322, Dallas MPEG Meeting, November 1995.
- [97] R. Danielson, “Simple Rate Control for Video VM Simulations”, MoMuSys Doc. WG2-0040, April 1996.
- [98] MPEG Video, “MPEG-4 Video Verification Model 4.0”, Doc. ISO/IEC JTC1/SC29/WG11 N1380, Chicago MPEG meeting, September 1996.
- [99] T. Chiang, Y.-Q. Zhang, “A Rate Control Scheme using a New Rate-Distortion Model”, Doc. ISO/IEC JTC1/SC29/WG11 M0436, Dallas MPEG Meeting, 1995.
- [100] T. Chiang, Y.-Q. Zhang, “A New Rate Control Scheme using Quadratic Rate Distortion Model”, *Proceedings of the 1996 International Conference on Image Processing (ICIP’96)*, vol. 1, pp. 73–76, Lausanne, Switzerland, September 1996.
- [101] T. Chiang, Y.-Q. Zhang, “A New Rate Control Scheme using Quadratic Rate Distortion Model”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 246–250, February 1997.
- [102] MPEG Video, “MPEG-4 Video Verification Model 5.0”, Doc. ISO/IEC JTC1/SC29/WG11 N1469, Maceió MPEG meeting, November 1996.
- [103] H.-J. Lee, T. Chiang, Y.-Q. Zhang, “Scalable Rate Control for Very Low Bit Rate (VLBR) Video”, *Proceedings of the 1997 International Conference on Image Processing (ICIP’97)*, vol. 2, pp. 768–771, Santa Barbara, CA, USA, October 1997.
- [104] H.-J. Lee, T. Chiang, Y.-Q. Zhang, “Scalable Rate Control for MPEG-4 Video”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 6, pp. 878–894, September 2000.
- [105] A. Vetro, H. Sun, Y. Wang, “Joint Shape and Texture Rate Control for MPEG-4 Encoders”, *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems (ISCAS’98)*, vol. 5, pp. 285–288, Monterey, CA, USA, May 1998.

- [106] P. Nunes, F. Marqués, F. Pereira, A. Gasull, “A Contour-based Approach to Binary Shape Coding using a Multiple Grid Chain Code”, *Signal Processing: Image Communication*, vol. 15, no. 7–8, pp. 585–599, May 2000.
- [107] A. Vetro, H. Sun, Y. Wang, “MPEG-4 Rate Control for Multiple Video Objects”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 186–199, Feb. 1999.
- [108] MPEG Video, “MPEG-4 Video Verification Model 8.0”, Doc. ISO/IEC JTC1/SC29/WG11 N1796, Stockholm MPEG meeting, July 1997.
- [109] J.-W. Lee, A. Vetro, Y. Wang, Y.-S. Ho, “Bit Allocation for MPEG-4 Video Coding with Spatio-Temporal Tradeoffs”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 488–502, June 2003.
- [110] J. Ribas-Corbera, S. Lei, “Optimal Quantizer Control in DCT Video Coding for Low-Delay Video Communications”, *Proceedings of the 1997 Picture Coding Symposium (PCS’97)*, pp. 749–754, Berlin, Germany, September 1997.
- [111] J. Ribas-Corbera, S. Lei, “Rate Control in DCT Video Coding for Low-Delay Communications”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 1, pp. 172–185, February 1999.
- [112] J. Ribas-Corbera, S. Lei, “A Frame-Layer Bit Allocation for H.263+”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 7, pp. 1154–1158, October. 2000.
- [113] ITU-T SG16/Q15, “Video Test Model Number 10 (TMN10)”, T. Gardos (Ed.), April 1998.
- [114] J. Ronda, M. Eckert, S. Rieke, F. Jaureguizar, Á. Pacheco, “Advanced Rate Control for MPEG-4 Coders”, *Proceedings of the 1998 SPIE Visual Communications and Image Processing (VCIP’98)*, vol. 3309, pp. 383–394, San Jose, CA, USA, January 1998.
- [115] Z. He, Y. K. Kim, S. Mitra, “ ρ -Domain Source Modeling and Rate Control for Video Coding and Transmission”, *Proceedings of the 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’01)*, vol. 3, pp. 1773–1776, Salt Lake City, UT, USA, May 2001.
- [116] Z. He, S. Mitra, “A Unified Rate-Distortion Analysis Framework for Transform Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1221–1236, December 2001.
- [117] Z. He, S. Mitra, “A Linear Source Model and a Unified Rate Control Algorithm for DCT Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 11, pp. 970–982, November 2002.
- [118] Z. He, Y. K. Kim, S. Mitra, “Object-Level Bit Allocation and Scalable Rate Control for MPEG-4 Video Coding”, *Proceedings of the 2001 Workshop and Exhibition on MPEG-4 (WEMP’01)*, pp. 63–66, San Jose, CA, USA, June 2001.

References

- [119] Z. He, Y. K. Kim, S. Mitra, “Low-Delay Rate Control for DCT Video Coding via ρ -Domain Source Modeling”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 8, pp. 928–940, August 2001.
- [120] Z. He, S. Mitra, “ ρ -Domain Bit Allocation and Rate Control for Real Time Video Coding”, *Proceedings of the 2001 IEEE International Conference on Image Processing (ICIP’01)*, vol. 3, pp. 546–549, Thessaloniki, Greece, October 2001.
- [121] Z. He, S. Mitra, “Optimum Bit Allocation and Accurate Rate Control for Video Coding via ρ -Domain Source Modeling”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 840–849, October 2002.
- [122] Y. Sun, I. Ahmad, “New Rate Control Algorithm for MPEG-4 Video Coding”, *Proceedings of the 2002 SPIE Visual Communications and Image Processing (VCIP’02)*, vol. 4671, pp. 698–709, San Jose, CA, USA, January 2002.
- [123] Y. Sun, I. Ahmad, “A Robust and Adaptive Rate Control Algorithm for Object-based Video Coding”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 10, pp. 1167–1182, October 2004.
- [124] Y. Sun, I. Ahmad, J. Luo, X. Wei, “Synchronous and Asynchronous Multiple Object Rate Control for MPEG-4 Video Coding”, *Proceedings of the 2003 IEEE International Conference on Image Processing (ICIP’03)*, vol. 3, pp. 801–804, Barcelona, Spain, September 2003.
- [125] Y. Sun, I. Ahmad, “Asynchronous Rate Control for Multi-Object Videos”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 8, pp. 1007–1018, August 2005.
- [126] A. Jagmohan and K. Ratakonda, “MPEG-4 One-Pass VBR Rate Control for Digital Storage”, *Proceedings of the 2002 International Conference on Image Processing (ICIP’02)*, vol. 3, pp. 709–712, Rochester, New York, USA, June 2002.
- [127] A. Jagmohan, K. Ratakonda, “MPEG-4 One-Pass VBR Rate Control for Digital Storage”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 5, pp. 447–452, May 2003.
- [128] C. Hung, D. Lin, “Towards Jointly Optimal Rate Allocation for Multiple Videos with Possibly Different Frame Rates”, *Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS’00)*, vol. 2, pp. 13–16, Geneva, Switzerland, May 2000.
- [129] J. Yang, X. Fang, H. Xiong, “A Joint Rate Control Scheme for H.264 Encoding of Multiple Video Sequences”, *IEEE Transactions on Consumer Electronics*, vol. 51, no. 2, pp. 617–623, May 2005.
- [130] B. Maglaris, D. Anastassiou, P. Sen, G. Karlsson, J. Robbins, “Performance Models of Statistical Multiplexing in Packet Video Communications”, *IEEE Transactions on Communications*, vol. 36, no. 7, pp. 834–844, July 1988.
- [131] D. Reininger, D. Raychaudhuri, B. Melamed, B. Sengupta, J. Hill, “Statistical Multiplexing of VBR MPEG Compressed Video on ATM Networks”, *Proceedings of*

- the Twelfth Annual Joint Conference of the IEEE Computer and Communications Societies. Networking: Foundation for the Future (INFOCOM'93)*, vol. 3, pp. 919–926, San Francisco, CA, USA, March 1993.
- [132] L. Wang, A. Vincent, “Joint Rate Control for Multi-Program Video Coding”, *IEEE Transactions on Consumer Electronics*, vol. 42, no. 3, pp. 300–305, August 1996.
 - [133] L. Böröczky, A. Y. Ngai, E. F. Westermann, “Statistical Multiplexing Using MPEG-2 Video Encoders”, *IBM Journal of Research and Development*, vol. 43, no. 4, pp. 511–520, July 1999.
 - [134] MPEG Test Model Editing Committee, “MPEG-2 Test Model 5”, Doc. ISO/IEC JTC1/SC29/WG11 N400, Sydney MPEG meeting, April 1993.
 - [135] F. Pan, Z. Li, L. Pang, G. Feng, “Reducing Frame Skipping in MPEG-4 Rate Control Scheme”, *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, vol. 4, pp. 3409–3412, Orlando, FL, USA, May 2002.
 - [136] F. Pan, Z. Li, K. Lim, G. Feng, “A Study of MPEG-4 Rate Control Scheme and its Improvements”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 5, pp. 440–446, May 2003.
 - [137] Z. Chen, K. Ngan, “Improved Single Video Object Rate Control for MPEG-4”, *Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03)*, vol. 3, pp. 85–88, Hong Kong, April 2003.
 - [138] K. Ngan, T. Meier, Z. Chen, “Improved Single-Video-Object Rate Control for MPEG-4”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 5, pp. 385–393, May 2003.
 - [139] Z. Chen, K. Ngan, C. Zhao “Improved Rate Control for MPEG-4 Video Transport over Wireless Channel”, *Signal Processing: Image Communication*, vol. 18, no. 10, pp. 879–887, November 2003.
 - [140] Z. Chen, K. Ngan, “Optimal Bit Allocation for MPEG-4 Multiple Video Objects”, *Proceedings of the 2004 International Conference on Image Processing (ICIP'04)*, vol. 2, pp. 761–764, Singapore, October 2004.
 - [141] Z. Chen, K. Ngan, “Object-based Rate Control for MPEG-4 Video Object Coding”, *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems (ISCAS'04)*, vol. 3, pp. 973–976, Vancouver, Canada, May 2004.
 - [142] J. Valentim, “An Alternative Complexity Model for the MPEG-4 Visual Standard”. Master Thesis, May 2001.
 - [143] V. Bhaskaran, K. Konstantinides, “Image and Video Compression Standards: Algorithms and Architectures”, Kluwer Academic Publishers, 1995.
 - [144] L. Davisson, “Rate-Distortion Theory and Application”, *Proceedings of the IEEE*, vol. 60, no. 7, pp. 800–808, July 1972.

References

- [145] T. Berger, J. Gibson, “Lossy Source Coding”, *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2693–2723, October 1998.
- [146] C. Shannon, “A Mathematical Theory of Communication”, *Bell System Technical Journal*, vol. 27, pp. 397–423, 623–656, 1948.
- [147] T. Berger, “Rate-distortion Theory”, Prentice Hall, Englewood Cliffs, NJ, 1971.
- [148] C. Shannon, “Coding Theorems for a Discrete Source with a Fidelity Criterion”, *IRE National Convention Record*, vol. 7, part 4, pp. 142–163, NY, March 1959. (Also in Claude Elwood Shannon: Collected Papers, N. J. A. Sloane and A. D. Wyner, Eds., IEEE Press, Piscataway, NJ, 1993).
- [149] A. Wyner, “Fundamental Limits in Information Theory “, *Proceedings of the IEEE*, vol. 69, no. 2, pp. 239–251, February 1981.
- [150] A. Ortega, K. Ramchandran, “Rate-Distortion Methods for Image and Video Compression”, *IEEE Signal Processing*, vol. 15, no. 6, pp. 23–50, November 1998.
- [151] A. Netravali, B. Haskell, “Digital Pictures: Representation, Compression, and Standards”, Plenum Press, NY, 1995.
- [152] J. Breitenbach, E. Weisstein, “Calculus and Analysis”, MathWorld – A Wolfram Web Resource, <http://mathworld.wolfram.com/Infimum.html>.
- [153] E. Weisstein, “Terminology” *MathWorld* – A Wolfram Web Resource, <http://mathworld.wolfram.com/Closed-FormSolution.html>.
- [154] G. Schuster, A. Katsaggelos, “Rate-Distortion Video Compression”, Kluwer Academic Publishers, 1997.
- [155] A. Gersho, R. Gray, “Vector Quantization and Signal Compression”, Kluwer Academic Publishers, 1992.
- [156] H.-M. Hang, J.-J. Chen, “Source Model for Transform Video Coder and Its Application – Part I: Fundamental Theory”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 287–298, April 1997.
- [157] N. Jayant, P. Noll, “Digital Coding of Waveforms: Principles and Applications to Speech and Video Coding”, Prentice-Hall, 1984.
- [158] R. Gonzalez, R. Woods, “Digital Image Processing”, Addison-Wesley, 1992.
- [159] H. Gish, J. Pierce, “Asymptotically Efficient Quantizing”, *IEEE Transactions on Information Theory*, vol. 14, pp. 676–683, September 1968.
- [160] L.-J. Lin, A. Ortega, “Bit-Rate-Control using Piecewise Approximated Rate-Distortion Characteristics”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 4, pp. 446–459, August 1998.

- [161] W. Ding, B. Liu, “Rate-Control for MPEG Video Coding and Recording by Rate-Quantization Modeling”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6 no. 1, pp. 12–20, February 1996.
- [162] ITU-T/SG15, “Video Codec Test Model, TMN8”, Doc. Q15-A-59, Portland, June 1997.
- [163] L.-J. Lin, A. Ortega and C.-C. J. Kuo “Cubic Spline Approximation of Rate and Distortion Functions for MPEG Video”, *Proceedings of IS&T/SPIE, Digital Video Compression: Algorithms and Technologies 96*, San Jose, CA, USA, February 1996.
- [164] L.-J. Lin, A. Ortega and C.-C. J. Kuo “Rate-Control Using Spline-Interpolated R-D Characteristics”, *Proceedings of 1996 SPIE Visual Communications and Image Processing (VCIP’96)*, Orlando, FL, USA, March 1996.
- [165] J. Bai, Q. Liao, X. Lin, X. Zhuang, “Rate-Distortion Model Based Rate-Control for Real-time VBR Video Coding and Low-delay Communications”, *Signal Processing: Image Communication*, vol. 17, no. 2, pp. 187–199, February 2002.
- [166] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, “Numerical Recipes in C: The Art of Scientific Computing – 2nd Ed.”, Cambridge University Press, 1992.
- [167] M. Osborne, G. Smyth, “A Modified Prony Algorithm for Exponential Function Fitting”, *SIAM Journal of Scientific Computing*, vol. 16, no. 1, pp. 119–138, January 1995.
- [168] Z. He, A. Mitra, “Optimal Bit Allocation and Accurate Rate Control for Video Coding via p-Domain Source Modeling”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 840–849, October 2002.
- [169] M. Effros, “Optimal Modeling for Complex System Design”, *IEEE Signal Processing*, vol. 15, no. 6, pp. 51–73, November 1998.
- [170] R. Iserman, “Digital Control Systems I (2nd Ed.)”, Springer Verlag, Berlin, 1989.
- [171] G. Keesman, I. Shah, R. Klein-Gunnawiek, “Bit-Rate-Control for MPEG Encoders”, *Signal Processing: Image Communication*, vol. 6, no. 6, pp. 545–560, February 1995.
- [172] K. Åström, B. Wittenmark, “Adaptive Control (2nd Ed.)”, Addison-Wesley, 1995.
- [173] CCITT SGXV, “Description of Reference Model 8 (RM8)”, Doc. 525, June 1989.
- [174] K. Ogata, “Modern Control Engineering (2nd Ed.)”, Prentice hall, 1990.
- [175] M. Galassi et al., “GNU Scientific Library Reference Manual (2nd Ed.)”, March 2006, <http://www.gnu.org/software/gsl/>.
- [176] MPEG Video and Test Groups, “Call for Proposals on Scalable Video Coding Technology”, Doc. ISO/IEC JTC1/SC29/WG11 N6193, Waikoloa MPEG meeting, December 2003.
- [177] E. Weisstein, “Probability and Statistics”, MathWorld – A Wolfram Web Resource,

References

- <http://mathworld.wolfram.com/VariationCoefficient.html>.
- [178] G. Bjontegaard, “Calculation of Average PSNR Differences Between RD-curves”, Doc. VCEG-M33, Austin, USA, April 2001.
- [179] P. Correia, F. Pereira, “Estimation of Video Object’s Relevance,” *Proceedings of the X European Signal Processing Conference (EUSIPCO’00)*, pp. 925–928, Tampere, Finland, September 2000.
- [180] ISO/IEC 21000-7:2004, “Information technology – Multimedia Framework (MPEG-21) – Part 7: Digital Item Adaptation”, 2004

