

VISUAL ATTENTION DRIVEN IMAGE TO VIDEO ADAPTATION

Joel Baltazar, n.º. 46578, AE de Sistemas, Decisão e Controlo
Pedro Pinho, n.º. 46662, AE de Sistemas, Decisão e Controlo

DEGREE IN ELECTRICAL AND COMPUTER ENGINEERING
Graduation Report
5/2004/L

Supervisor: Prof. Fernando Pereira

September 2005

Acknowledgments

Many people have been part of our graduate education as teachers, friends and colleagues, to whom we now dedicate some words of appreciation.

First and foremost we would like to thank Prof. Fernando Pereira, the best teacher and advisor we could have wished for. We are deeply thankful for his guidance, dedication and active involvement in the development of this thesis. His remarks, always having in mind our best interests, complemented by his good humor were a huge incentive to us.

At the IT Image Group we were surrounded by knowledgeable and friendly people, who provided a pleasant work environment and were always available to help, which greatly contributed to this project, and for that we thank them. A special word of appreciation goes to João Ascenso and Duarte Palma, for providing the face and text detection algorithms, respectively, and their support on how to use them.

Last but not least, we would like to thank those closest to us, whose presence contributed so much to the completion of this graduation thesis. We would like to thank our families, specially our parents, and our friends and colleagues for their absolute confidence, and to whom we are forever indebted for their understanding, endless patience and encouragement when it was most required.

Abstract

Nowadays, the heterogeneity of networks, terminals, and users is growing. At the same time, the availability and usage of multimedia content is increasing, which has raised the relevance of content adaptation technologies able to fulfill the needs associated to all usage conditions. For example, mobile displays tend to be too small to allow one to see all the details of an image. In this report, a solution for this problem is proposed: an automatic adaptation system, that uses visual attention models, creates a video clip that browses through the image displaying its regions of interest in detail.

The report describes the developed architecture for the adaptation system, the processing solutions and also the principles and reasoning behind the algorithms that have been developed and implemented to achieve the objective of this work.

In order to evaluate the performance of the adaptation system, a user study has been conducted. The results of the study are encouraging, since they indicate that users consider the quality of the experience provided by the video clips to be better than the still image experience.

Keywords

Transmoding, Video Adaptation, Image Browsing, Visual Attention, Regions of Interest.

Resumo

Hoje em dia, a heterogeneidade de redes, terminais e utilizadores é cada vez maior. Ao mesmo tempo, a disponibilidade e acesso a conteúdos multimédia está a crescer, o que tem aumentado a relevância das tecnologias de adaptação de conteúdos capazes de preencher as necessidades associadas a todas as condições de utilização. Por exemplo, os ecrãs dos telemóveis são geralmente demasiado pequenos para que uma pessoa consiga captar todos os detalhes de uma imagem. Neste relatório é apresentado uma solução para este problema: um sistema de adaptação automático que, utilizando modelos de atenção visual, cria um vídeo *clip* que percorre a imagem mostrando as suas regiões de interesse em detalhe.

O relatório descreve a arquitectura desenvolvida para o sistema de adaptação, as soluções de processamento, bem como os princípios e razões subjacentes aos algoritmos que foram desenvolvidos e implementados para atingir o objectivo deste trabalho.

De modo a avaliar a performance do sistema de adaptação, realizou-se um inquérito a um conjunto de utilizadores. Os resultados do inquérito são animadores, uma vez que os utilizadores consideram que a qualidade da experiência proporcionada pelos vídeo *clips* é melhor do que a da imagem simples.

Palavras-chave

Transmoding, Adaptação de Vídeo, *Image Browsing*, Atenção Visual, Regiões de Interesse.

Contents

1. Introduction	1
1.1 Project Context and Objectives	1
1.2 Contributions of this Work	2
1.3 Report Organization	3
2. Review on Visual Attention Modeling	5
2.1 Human Visual System	5
2.1.1 Human Visual System Anatomy	5
2.1.2 Visual Cell's Properties and Functions	7
2.1.3 Visual Attention Mechanism	8
2.2 Visual Attention Models	9
2.2.1 Model's Classification	9
2.2.2 Bottom-up Attention Models	9
2.2.3 Top-down Attention Models	15
2.3 Computer Vision Applications	20
2.4 Final Remarks	20
3. Image2Video Adaptation System	21
3.1 Image2Video Adaptation System Architecture	22
3.2 Background Algorithms	23
3.2.1 Saliency Detection Algorithm	24
3.2.2 Face Detection Algorithm	25
3.2.3 Text Detection Algorithm	28
3.3 Final Remarks	30
4. Processing for Image2Video Adaptation	31
4.1 Composite Image Attention Model	31

4.1.1	Saliency Attention Model.....	32
4.1.2	Face Attention Model.....	35
4.1.3	Text Attention Model.....	37
4.2	Attention Models Integration	40
4.2.1	Face-Text Integration	40
4.2.2	Face-Saliency Integration.....	40
4.2.3	Text-Saliency Integration.....	42
4.2.4	Final Attention Value Computation	42
4.2.5	Final AOs Validation	43
4.3	Optimal Path Generation	43
4.3.1	Display Size Adaptation.....	43
4.3.2	Browsing Path Generation	46
4.4	Video Creation	49
4.4.1	Key Frame Types	49
4.4.2	Motion Units	50
4.4.3	Motion Durations and Perception Times	52
4.4.4	Video Display Modes.....	52
4.4.5	Video Directing.....	52
4.5	Final Remarks.....	54
5.	Image2Video Application Interface	55
5.1	Interface Layout	55
5.1.1	Application Menu.....	56
5.1.2	Customize Menu	57
5.2	Running the Application	58
5.3	Final Remarks.....	60
6.	Evaluation Results and Conclusions	61
6.1	User Evaluation Study.....	61
6.1.1	Objectives.....	61
6.1.2	Methodology and Conditions	62
6.1.3	Results Analysis	62
6.2	Summary and Conclusions.....	63
6.3	Future Work	64
A.	CD-ROM Contents.....	65

List of Figures

Figure 1.1: Different terminals access multimedia content through different networks (extracted from [1])	1
Figure 2.1: Human visual system structure (extracted from [8]).....	6
Figure 2.2: Brain areas involved in the deployment of visual attention (from [10]).....	6
Figure 2.3: Receptive field center-surround organization (extracted from [12]).	7
Figure 2.4: Example of a human scan path (from [8]).....	8
Figure 2.5: Architecture of Itti's model	10
Figure 2.6: Mechanisms used to compute a conspicuity map	11
Figure 2.7: Example of the results produced by the Itti model (extracted from [18]).....	12
Figure 2.8: Architecture of the Le Meur's model	13
Figure 2.9: Comparison of the Le Meur's and Itti's FOA locations (from [19])	15
Figure 2.10: Architecture of Oliva's model.....	16
Figure 2.11: Architecture of Frintrop's model.....	18
Figure 2.12: Results obtained with the Frintrop's model searching for name plates	19
Figure 3.1: Image2Video adaptation system	21
Figure 3.2: Image2Video system main architecture	22
Figure 3.3: Example of attention results generated by Itti's bottom-up model	25
Figure 3.4: Architecture of the used face detection algorithm	25
Figure 3.5: Example of color analysis stage results.....	26
Figure 3.6: Example of results generated by the face detection algorithm.....	27
Figure 3.7: Architecture of the text detection algorithm	28
Figure 3.8: Example of image simplification stage results.....	28
Figure 3.9: Example of image segmentation stage results.....	29
Figure 3.10: Example of character detection stage results	30
Figure 3.11: Example of text detection results for the image shown in Figure 3.8 (a)	30
Figure 4.1: Architecture of the saliency attention model.....	32
Figure 4.2: Example of results generated by the AOs validation stage.....	33
Figure 4.3: Example of results generated by the ROIs merging stage	34
Figure 4.4: Structure of the saliency ROIs merging algorithm.....	34
Figure 4.5: Architecture of the face attention model	35
Figure 4.6: Example of face ROIs overlapping and merging	36
Figure 4.7: Position weight matrix	37
Figure 4.8: Architecture of text attention model.....	37
Figure 4.9: Horizontal distance calculation	38
Figure 4.10: Example of text ROIs merging.....	39

Figure 4.11: Structure of the text ROIs merging algorithm.....	39
Figure 4.12: Example of Face-Saliency integration	41
Figure 4.13: Structure of the Face-Saliency ROIs integration algorithm	41
Figure 4.14: Example of Text-Saliency integration.....	42
Figure 4.15: Structure of Split Processing algorithm	44
Figure 4.16: Example of results generated by AG split processing.....	44
Figure 4.17: Structure of Group Processing algorithm.....	45
Figure 4.18: Example of results generated by Group Processing.....	46
Figure 4.19: Example of spatial distribution of AGs.....	47
Figure 4.20: Structure of the Optimal Path Generation algorithm (Main part)	47
Figure 4.21: Structure of the Optimal Path Generation algorithm (Node 1)	48
Figure 4.22: Structure of the Optimal Path Generation algorithm (Node 2)	49
Figure 4.23: Example of browsing path.....	49
Figure 4.24: Examples of the three key-frame types	50
Figure 4.25: Video clip generation algorithm.....	53
Figure 5.1: Main window of the developed Image2Video application interface	56
Figure 5.2: The application menus	56
Figure 5.3: Image2Video About window	57
Figure 5.4: Customize menu window	58
Figure 5.5: Examples of option windows: Optimal Path and Video Creation	58
Figure 5.6: Action buttons and intermediate results selection.....	58
Figure 5.7: Example of AOs labels.....	59
Figure 5.8: Video player window	60
Figure A.1: CD-ROM directory structure.....	65

List of Tables

Table 4.1: Types of video motion units	50
Table 4.2: Default minimal perceptible times.....	52
Table 4.3: Default motion units duration times	52
Table 6.1: Evaluation results for Question 1.....	63
Table 6.2: Evaluation results for Question 2.....	63
Table 6.3: Evaluation results for Question 3.....	63

List of Acronyms

HVS – Human Visual System

fMRI – functional Magnetic Resonance Imaging

LGN – Lateral Geniculate Nucleus

SC – Superior Colliculus

PPC – Posterior Parietal Cortex

IT – Inferotemporal

PFC – Prefrontal Cortex

CRF – Classical Receptive Field

non-CRF – non-Classical Receptive Field

RGB – Red, Green and Blue

FOA – Focus of Attention

ROI – Region of Interest

WTA – Winner-Take-All

IOR – Inhibition of Return

CSF – Contrast Sensitivity Function

PDF – Probability Distribution Function

EM – Expectation Maximization

HSV – Hue-Saturation-Value

iNVT – ilab Neuromorphic Vision C++ Toolkit

RHT – Randomized Hough Transform

AOs – Attention Objects

AV – Attention Value

AGs – Attention Groups

OAGS – Ordered Attention Group Set

BPS – Browsing Path Set

FF – Full-Frame

MF – Medium-Frame

CF – Close-up Frame

PA – Pan

LP – Light Pan

LZI – Local Zoom In

LZO – Local Zoom Out

FZI – Full Zoom In

FZO – Full Zoom Out

ZR – Zoom Rate

SHR – Spatial Horizontal Resolution

LPV – Local Pan Velocity

MPT – Minimal Perceptible Times

TB – Time Based

AIB – Amount of Information Based

Chapter 1

Introduction

In this chapter, the project contextualization and main objectives are presented, as well as the contributions provided by this thesis to achieve those objectives. The chapter ends with a description of the report organization.

1.1 Project Context and Objectives

With the explosion of digital image and video technology, it is nowadays largely felt and recognized that we live in a visual age. Television is no longer the only way to access multimedia content, since recent technological developments have opened new frontiers to the consumption of multimedia content, notably following an everywhere, at anytime paradigm. This has lead to a growing heterogeneity of networks, terminals and users, and an increase in the availability and usage of multimedia content, as Figure 1.1 shows.

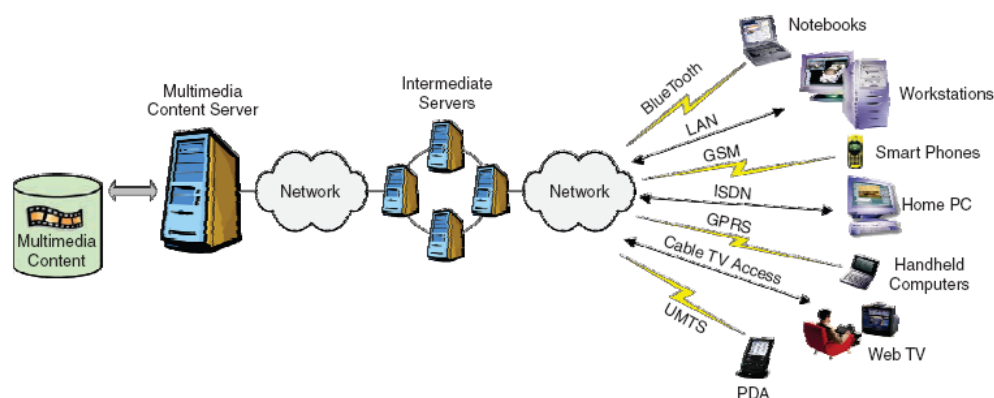


Figure 1.1: Different terminals access multimedia content through different networks (extracted from [1])

The diversity of users, multimedia content and terminals has prompted the development of solutions that allow a Universal Multimedia Access (UMA) [1]. This means that adaptations tools are becoming increasingly important to provide different presentations of the same information that suit different usage conditions. Furthermore, the importance of the user and not the terminal as the final point in the multimedia consumption chain is becoming clear [2]. The vision of mass delivery

of identical content is being replaced by one of mass customization of content centered on the user, and on the user experience. So we are starting to speak about Universal Multimedia Experiences (UME) which provide the users with adapted, informative (in the sense of cognition), and exciting (in the sense of feelings) experiences.

Nowadays people share some of their most important moments with others using visual content such as photographs, that they can easily capture on their mobile devices anywhere, at anytime. Therefore images are very important in mobile multimedia applications. However, mobile devices have several limitations, notably regarding computational resources, memory, bandwidth and display size. Technological advances will solve some of these limitations, but the display size will continue to be the major constraint on small mobile devices such as cell-phones and handheld PC's.

Currently, the predominant methods for viewing large images on small devices are down-sampling or manual browsing by zooming and scrolling. Image down-sampling results in significant information loss, due to excessive resolution reduction. Manual browsing can avoid information loss but is often time-consuming for the users to catch the most crucial information in an image. H. Liu et al. [3] have proposed an adaptation tool that allows the automatic browsing of large pictures on mobile devices. Their work transforms the image into a simple video sequence composed of pan and zoom movements which is able to automate the scrolling and navigation of a large picture on mobile devices.

In this report, we propose an adaptation system whose major objective is to maximize the user experience when consuming an image in a device with a small size display. The processing algorithms developed to reach this purpose imply determining the regions of interest (ROIs) in an image based on some knowledge of the human visual attention mechanism, and generating a video sequence that displays those regions according to certain user preferences, while taking into consideration the limitations of the display's size. User preferences refer to the video display modes the user can choose for the visualization of the adapted video sequence, e.g. the duration of the video. The created video is intended to provide a final, better user experience, compared to the down-sampled still image or the manual scrolling.

1.2 Contributions of this Work

When a person looks to an image, the way he or she analyzes it depends on a series of factors: the context, the type of objects present in the image, a pre-defined object to be searched for, etc. This means that there are a series of characteristics that can stimulate or not the HVS, i.e. which can influence where a person directs its visual attention. For example, regions that have different properties compared to their surroundings are visually salient, and therefore are likely to attract viewer's attention. Faces, which are one of human's most distinctive characteristics, and text, are two kinds of objects that are also likely to attract viewer's attention.

The adaptation system developed in this work is visual attention driven, i.e. knowledge of the human visual system behaviour is used to determine the regions of interest in an image. Therefore a composite image attention model has been developed, which uses three elementary visual attention models to detect salient, faces and text regions. In this system, these are the regions of interest of an image. The different regions of interest provided by the composite image attention model are then integrated into a single image map, which contains all the regions of interest, their location and type (saliency, face, text).

An algorithm has also been developed to optimize the information displayed at each moment on the screen, by determining the optimal browsing path, in terms of user experience, to display the regions of interest.

Based on the determined optimal path and the display user preferences, an algorithm has been developed to create a video sequence, which displays the regions of interest of an image with

maximum quality, i.e. without down-sampling, simulating the browsing path that a human would perform to analyze the image.

Finally, an interface has been developed, which integrates all the algorithms of the application, allowing adjusting their parameters and controlling their execution. The interface also allows visualizing all the algorithm results and the created video sequences. This interface has been used in this work to perform some subjective tests to evaluate the impact of the developed video visualization mode for images in display size constrained devices.

1.3 Report Organization

This thesis is composed of six chapters, including those regarding the introduction and conclusions, chapters one and six, respectively. The first chapter describes the project context and objectives, as well as the contributions of the developed work to fulfill those objectives.

Chapter 2 presents a review on visual attention, since the adaptation system presented in this report is visual attention driven. First, the anatomy and biological characteristics of the human visual system are presented, with special focus given to the visual attention mechanism. Afterwards, several computational models that emulate the human visual attention mechanism are presented. To conclude, the chapter highlights some examples of the usefulness of visual attention modeling in computer vision applications.

Chapter 3 presents the image to video adaptation system developed in this project, which is able to transform/adapt images to video driven by visual attention targeting a final better user experience. The main characteristics of the architecture of the adaptation system are presented, as well as the algorithms that are integrated but not fully developed.

Chapter 4 is dedicated to the detailed presentation of the major processing modules that compose the architecture of the developed adaptation system and which have been fully developed in this work. The objective of this adaptation system is to create a video sequence that provides a better user experience for an image by attending the ROIs of an image in an adequate way. In order to demonstrate how this objective is achieved, the algorithmic solutions developed by the authors of this project for the various processing modules, and the principles behind them are presented in detail in this chapter.

Chapter 5 is dedicated to the presentation of the developed application interface, which integrates the algorithms presented in Chapters 3 and 4. The interface allows controlling the execution of the algorithms, adjusting their parameters and viewing the produced results.

Chapter 6 finalizes the report by presenting the results of the subjective assessment tests that were carried out to evaluate the improved experience performance of the developed system, the project conclusions and future work.

The project described in this report was developed at the Image Group of Instituto de Telecomunicações at Instituto Superior Técnico.

Chapter 2

Review on Visual Attention Modeling

Vision is an area of research which involves the convergence of many different fields of study, including biology, psychology, neuropsychology, neuroscience, philosophy, and computer science. This interdisciplinary research area has delivered greater understanding of the human visual system, and in particular of the visual attention mechanisms.

This chapter presents a review of the most significant findings which have lead to a better understanding of visual attention and its application in computer vision.

2.1 Human Visual System

The design of computer vision systems is usually biologically inspired on the neuronal computational structures involved in processing, storing and interpreting the spatio-temporal information [4]. This section provides a minimum background on the human visual system (HVS) to help the reader understanding why humans need visual attention, and how the models described in this report relate to biology. For a more detailed description, readers are referred to [4, 5, 6].

2.1.1 Human Visual System Anatomy

The human brain cannot process all stimuli in parallel at the same time, so the visual system is organized in a way which reduces the amount of information that must be processed by higher level tasks in the brain. There are processing limitations regarding the wavelengths of light captured by the retina¹, as well as on the spatial and temporal frequencies that can be detected [7]. But more importantly, only a small region of the visual field is analyzed in high resolution by the fovea, the central high resolution area of the retina, which requires the HVS to have an attention mechanism capable of both selecting regions of interest for further processing, and producing successive deployments of the fovea to attend those regions. The eye movements responsible for shifting the fovea onto a given target are called saccades.

Studies based on patient's examination and especially functional magnetic resonance imaging (fMRI) have provided greater knowledge of the human visual attention mechanism. The

¹ The retina is the rear portion of the eye containing photoreceptors and several types of sensory neurons.

studies show that visual perception of what lays around us results from a series of transformations of the information collected by the eye, along the visual pathway, which comprehends several anatomical structures. First, the light which arrives at the eye produces a retinal image. The retina then produces a signal which is sent through the optic nerve to two different brain centers: the lateral geniculate nucleus (LGN) and the superior colliculus (SC), as can be seen in Figure 2.1.

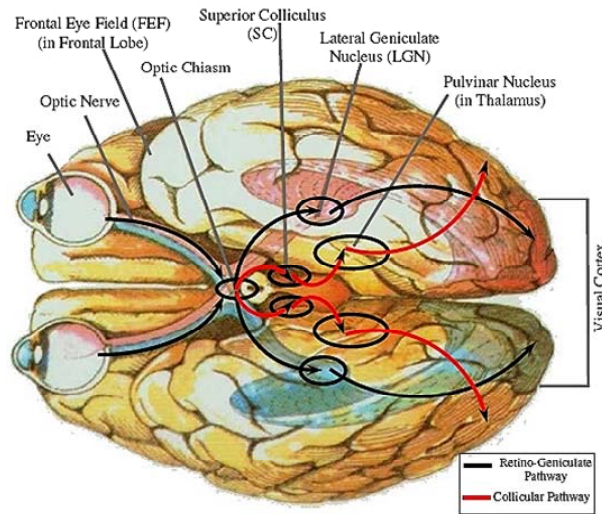


Figure 2.1: Human visual system structure (extracted from [8]).

The cells in the LGN contain topographic maps, i.e., representations of spatially related points in the visual field. The retino-geniculate pathway processes about 90% of the visual information generated by the retina, transmitting from the LGN to the primary visual cortex area, also known as striate cortex, which dedicates 50% of its area to representing the information gathered by the fovea [8]. Visual information is then forwarded to the extrastriate cortex area, and thereafter progresses along two visual streams, the dorsal (where) and ventral (what) streams, as shown in Figure 2.2. Cortical areas along the dorsal stream, namely the posterior parietal cortex (PPC), process spatial information, and direct attention towards eye-catching regions. Areas along the ventral stream, namely the inferotemporal cortex (IT), process and represent object features, identifying visual stimuli [9].

The collicular pathway processes 10% of the visual information generated by the retina, transmitting from the superior colliculus to the pulvinar nucleus. From there information is relayed to the extrastriate cortex. The collicular pathway is an important part of the visual attention mechanism, as the literature often suggests its involvement in both eye movements (see Figure 2.2) and building the global map of attention, which marks regions of interest in the retinal image [8, 9, 10].

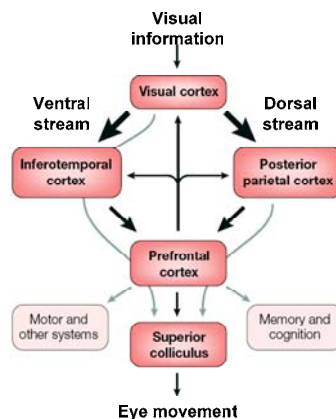


Figure 2.2: Brain areas involved in the deployment of visual attention (from [10]).

Scene understanding involves object recognition and deployment of attention, so the dorsal and ventral streams, this means the ‘where’ and ‘what’ streams, must interact. The prefrontal cortex (PFC) is responsible for eye movements, through the SC, but also for the interaction between the where and what streams, having bidirectional connections to both the PPC and IT. This allows the PFC to integrate spatial attention and stimulus analysis information, and influence the processing that occurs in the dorsal and ventral streams. The PFC provides excitatory signals that simultaneously bias processing, this means it can select the neural pathways needed to perform a certain task, such as identifying an object [11].

2.1.2 Visual Cell’s Properties and Functions

The previous section provides some insight on the several anatomical structures concerning the HVS, as well as some of their functions, which are related to their biological properties.

The neuronal tissue of each anatomical structure comprehends several neuronal layers, which correspond to cells which exhibit a coherent behavior when responding to a certain stimulus, and use neurons to compute and process information. For any particular cell, the region of the visual field in which an appropriate stimulus can produce a response is called receptive field [6].

Cells in the retina and LGN have center-surround receptive field characteristics, as shown in Figure 2.3. In a small central region, stimulus may excite (ON) or inhibit (OFF) a neuron. In the surrounding annular region, the same stimulus will provide an opposite effect to that of the center. This characteristic provides intensity-contrast information about the visual stimulus. There are two types of center-surround cells: On-Center Off-Surround, and Off-Center On-Surround. The center-surround receptive field is sometimes referred to as classical receptive field (CRF) in the literature.

Cells in the visual cortex have different kinds of receptive field characteristics, other than center-surround. For instance, about 80% of the cells have orientation-selective receptive fields [19], this means, they respond to lines in determined orientations, depending on width, orientation, angle and position, acting as edge detectors. Some cells are tuned to detect more elaborate patterns, such as triangles, circles and objects.

Visual cortex cells respond directly only to stimuli within their CRF, but their activity may be modulated by contextual stimuli outside the CRF. Contextual influences are commonly referred to as non classical receptive field (non-CRF) inhibition. Depending on the stimuli configuration, contextual influences can be suppressive or facilitative; this means that the magnitude of the excitatory response to an appropriate stimulus presented inside the CRF can be reduced or increased, respectively.

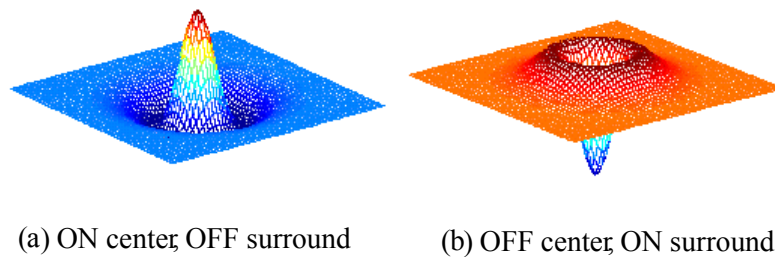


Figure 2.3: Receptive field center-surround organization (extracted from [12]).

The most important information represented by the visual pathway is image contrast, this means, the ratio of the local intensity and the average image intensity. There is evidence that cells can change their sensitivity to compensate for changes in the mean illumination level. Furthermore studies suggest that contrast information is represented at several spatial scales and orientation, and cells respond to stimuli above a certain contrast value called visibility threshold, which depends on the spatial frequency, orientation, and other parameters of the stimuli.

Another important aspect of vision is color and how the HVS processes it. Photoreceptors in the retina are sensitive to three colors: red, green and blue (RGB). Starting at the LGN an opponent-color representation is used throughout the visual pathway. The concept of opponent-color is based on the fact that some pairs of colors can coexist in a single color sensation, while others cannot. For example, humans can perceive orange which is composed of red and yellow, and cyan, which is composed of blue and green. However humans cannot experience a color sensation that is simultaneously blue and yellow, red and green, or black and white. The opponent-color representation of the HVS uses three channels: Red-Green, Blue-Yellow, and Black-White.

2.1.3 Visual Attention Mechanism

Given the size of the human brain, it is impossible to process all the information present in the visual field. section 2.1.1 presents the HVS organization, which has several mechanisms to progressively reduce the overwhelming information captured by the retina, before it reaches the brain for analysis. Visual attention is the most important of all mechanisms.

Attention is defined as the ability of a vision system, either biological or artificial, to rapidly select the regions of the visual field which are more likely to captivate human's interest, and direct the focus of attention (FOA) to those regions. This mechanism allows humans to reduce the amount of information associated to the visual field that must be processed by higher level tasks in the brain.

While exploring a scene, humans move their eyes to attend all the regions of interest in the visual field. The successive eye movements create the so called scan path, defined as a sequence of eye fixations linked by saccades. In Figure 2.4, the red dots represent a fixation or region of interest and the blue lines connecting them represent the saccade from one region to the next. The order by which regions are attended is particular to the human observer and scene [13].

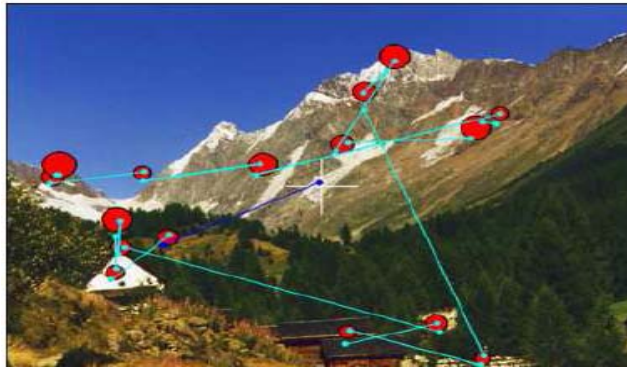


Figure 2.4: Example of a human scan path (from [8])

The HVS focuses on “interesting” image regions guided by two main influences: bottom-up or image derived-cues, and top-down or task derived cues [14].

The bottom-up influence is related to involuntary attention, this means, certain features of the visual field automatically attract our attention, they stand out in the scene, and therefore are called pop-out stimuli. For example, color and orientation are undoubted features that guide attention [7].

The top-down influence is connected to the current task in the human mind, such as looking for a specific object like a face or word. In this case, attention is biased towards whatever information is relevant to the current task, this means that knowledge about a target such as its probable location, physical attributes or context guide our attention by providing a clue to indicate where a target may appear.

2.2 Visual Attention Models

The following sections provide a review of the literature regarding computational models of visual attention, pointing out their main characteristics and applications to computer vision. A complete review on the existing computational models of visual attention can be found in [10] and [15].

2.2.1 Model's Classification

Most visual attention computational models are based on the feature integration theory model proposed by Treisman in [16]. Supported on experimental results, Treisman proposed a model with a pre-attentive stage, involving bottom-up processing, followed by an attentive stage, involving top-down processing. The first stage involves the extraction of feature maps, such as color and orientation, in a parallel way across the entire visual field, providing a map of stimuli activity. The attentive stage works in a serial fashion, attending single scene items or locations until a certain target is identified.

Computational models of visual attention are typically inspired on the characteristics of the HVS, and based on that they can be classified into two classes:

- **Bottom-up:** These models rely on the principle that when humans look at a scene without any prior knowledge or task, their attention will be attracted to locations mainly because of their saliency; this means that regions with different properties from the neighboring regions are considered more informative and are supposed to attract attention. Typically these models provide as an output a saliency map, which marks regions of interest in the image. This map is purely data-driven, i.e., computed using several bottom-up features of the image, such as color, contrast, orientation, etc.
- **Top-down:** These models are based on the fact that humans use knowledge derived from prior experience to determine regions of interest, meaning that relevant information to the current task is favored for further analysis. A usual task is identifying an object, and these models can orientate sensors in a certain direction to search for well-known features of an object, such as color and shape, for example. Although some of these models use bottom-up influences, top-down information is used to control where attention is directed to. Typically, as an output there is a saliency map or a description of the regions of interest, featuring the identity and location of the objects present in the scene.

In the following sections, some of the most relevant models within each class available in the literature are presented.

2.2.2 Bottom-up Attention Models

Koch and Ullman [17] presented a biologically-plausible architecture of the HVS attention mechanism, which has been implemented in detail by Itti et al [18], and has also inspired the development of a model by Le Meur et al [19]. In this section, we present a brief review of Itti's and Le Meur's models and finalize with a summary description of other works which have dealt with the problem of modeling visual attention using bottom-up processing.

A) Itti's Model

This model relies on the fact that the saliency of locations is deeply influenced by the surrounding context, and a unique scalar map, called saliency map, represents the saliency of locations over the entire visual field.

Based on these characteristics, the model has a general architecture which is presented in Figure 2.5, and comprehends four stages:

- 1) **Feature Extraction:** decomposition of the input image into several feature maps, each one sensitive to a different visual feature, such as color or orientation.
- 2) **Conspicuity Operator:** creation of conspicuity² maps, which mark regions of the image which strongly differ from their surroundings, according to a certain feature, and regardless of their size.
- 3) **Conspicuity Maps Integration:** combination of the conspicuity maps into a unique saliency map, which encodes the salient regions of the visual field.
- 4) **Salient Region Selection:** a Winner-Take-All (WTA) mechanism is employed to select the most salient regions one after the other; an inhibition of return (IOR) mechanism prevents the FOA from returning to previously attended locations.

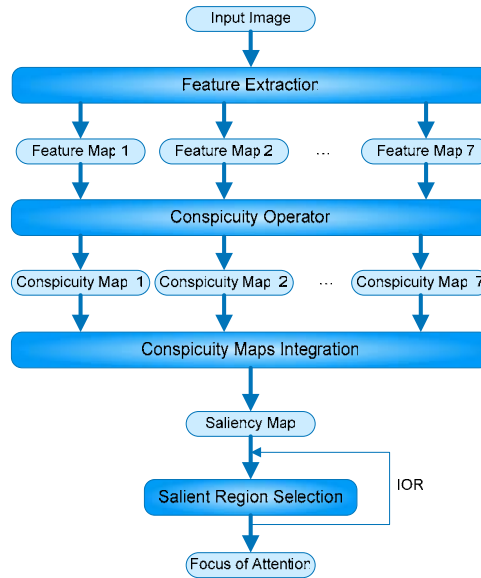


Figure 2.5: Architecture of Itti's model

The four stages of the Itti's model are now explained in more detail.

1) Feature Extraction

The first stage of the model uses a RGB input image to create several representations of the scene based on seven different features, grouped into three types: intensity, color, and orientation.

Intensity is simply obtained by

$$I = 0.3 \bullet R + 0.59 \bullet G + 0.11 \bullet B \quad (2.1)$$

For color, the two features Red-Green and Blue-Yellow are used to account for the opponent color representation of the HVS, calculated using the following equations:

$$RG = \frac{R - G}{I} \quad (2.2)$$

² Obvious to the eye or mind and therefore attracts attention

$$BY = \frac{B - Y}{I} \quad (2.3)$$

Regarding orientation, four different angle values, $\theta \in \{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$ are used to compute the respective feature maps. This is done using Gabor filters, which approximate the receptive field sensitivity profile of orientation-selective neurons in the primary visual cortex.

2) Conspicuity Operator

After obtaining the seven feature maps, the next step is to create the corresponding seven conspicuity maps, which mark regions of the image strongly differing from their surroundings, according to a certain feature, and regardless of their size.

For each feature map, a Gaussian pyramid is created, which is a hierarchy of low-pass filtered and subsampled versions of the original feature map. This multi-resolution representation with nine scales is obtained using a Gaussian filter. The creation of the conspicuity maps requires a conspicuity operator to enhance regions of the image that strongly differ from their surroundings, the so-called conspicuous locations. In this case a multiscale center-surround filter is used to achieve this.

The center-surround filter is implemented by calculating cross-scale differences between fine (c for center) and coarse (s for surround) scales. This is done by first interpolating the coarser scale to the finer scale and carrying out point-by-point subtraction. For each feature, a set of six intermediate conspicuity maps are computed using several scales for both c and s , as Figure 2.6 shows. In the example, the maps are obtained using the center pixel at scale $c \in \{2, 3, 4\}$, and the surround pixel at scale $s = c + \delta$, with $\delta \in \{3, 4\}$.

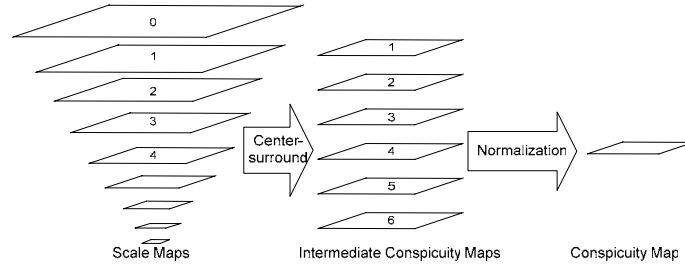


Figure 2.6: Mechanisms used to compute a conspicuity map

To obtain a conspicuity map for each feature, a normalization operator is applied to the intermediate conspicuity maps, simulating a competition for saliency between different scales. The normalization operator that is used globally promotes maps which have a small number of conspicuous locations and suppresses maps which contain homogeneous areas. This behavior was designed based on the biological evidence that neighboring similar features inhibit each other, the so-called non-CRF suppressive interactions found in primary visual cortex cells.

3) Conspicuity Maps Integration

To obtain the final saliency map, the seven conspicuity maps are normalized and integrated.

For each cue (intensity, color and orientation), the different conspicuity maps are normalized using the previously described operator, and summed to yield a unique cue conspicuity map, so that features of the same cue compete directly for saliency.

Finally, the three cue-related conspicuity maps are normalized and summed to form the final saliency map, which provides a representation of the conspicuity of every location in the visual field, and thus provides a guide for the deployment of visual attention.

4) Salient Region Selection

The maximum of the saliency map defines the most salient location of the visual field, this means where the human eye is more likely to be drawn to. A WTA neural network [17] mechanism is used to select the most conspicuous region of the map. This region is considered as the most salient one (“winner”), and the FOA is shifted to this location. This location is subsequently suppressed through an IOR mechanism [20], allowing the next most salient location to become the winner, and preventing the FOA from returning to previously attended regions.

Figure 2.7 shows an example of the operation of the model. The brightest areas of the computed saliency map, Figure 2.7 (b), correspond to the most salient regions. The output image produced by the model, Figure 2.7 (c), shows the first four FOA locations, corresponding to the four most salient regions of the saliency map. Figure 2.7 (d) shows how the IOR mechanism inhibits the FOA locations in the saliency map as they are attended.

One of the most important aspects of this model is that it estimates the time the HVS would take to analyze the scene and produce saccades, i.e. it provides the time it would take until a certain ROI becomes the FOA. The exact details of how this is done are not provided, but the algorithm has a deterministic behavior and does not depend on the computational platform. The 260 ms indicated in Figure 2.7 (c) represent the time that a human would have taken to produce the successive saccades from the first FOA location to the last one. This time, from here after, will be referred to as the identification time of (all) the ROIs. The model allows defining the maximum time for analyzing images, the maximum number of ROIs or both.

In Figure 2.7 (c) the FOA locations are represented by a yellow circle, whose dimension is automatically set to $\max(imageWidth, imageHeight)/12$. The inventors of this model don’t provide any explanation as to why this value is used. However, since the model was presented the inventors have introduced a shape-estimator method, which can extract the shape of the objects surrounding the saliency point from the saliency map, the feature or conspicuity maps which provide the greater contribution to the saliency of the region. The inventors of this model don’t provide any details regarding the implementation of the shape-estimator method.

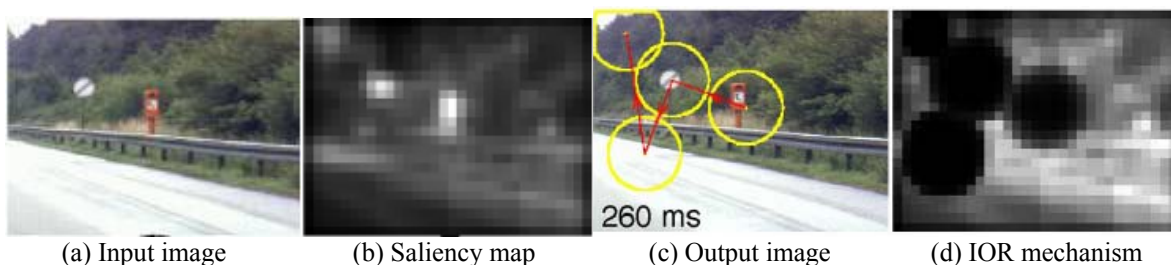


Figure 2.7: Example of the results produced by the Itti model (extracted from [18])

B) Le Meur’s Model

The extraction of visual features, such as color and orientation, is a common task to any bottom-up model, but the way this is done sometimes differentiates a model from another.

The main characteristic of the Le Meur's model is that the visual feature extraction takes into consideration the limited sensitivity of the HVS, i.e. this model simulates the HVS limitations to perceive with good precision all signals.

The model consists in four main stages, as shown in Figure 2.8:

- 1) **Visibility:** as the HVS is not able to perceive all information present in the visual field with the same accuracy, this stage simulates the limited human visual system sensitivity.
- 2) **Perception:** this stage corresponds to a process that produces from the psychovisual space a description useful to the viewers, and not cluttered with irrelevant information.
- 3) **Perceptual Grouping:** this stage refers to the human visual ability to group and to bind visual features to form a meaningful higher-level description structure.
- 4) **Salient Region Selection:** a WTA strategy is used in conjunction with IOR to select salient regions from the saliency map.

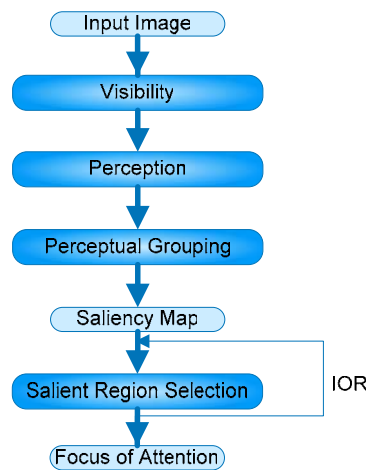


Figure 2.8: Architecture of the Le Meur's model

The four stages of the Le Meur's model are now explained in more detail.

1) Visibility

The objective of this stage is to simulate the limited sensitivity of the HVS. To achieve this four mechanisms are used, which are explained in the following paragraphs.

Transformation of the RGB luminance signal into the Krauskopf color space simulates the three different channels used by the brain to encode visual information. The first channel conveys the luminance signal, also called achromatic component (A). The second and third channels convey the chromatic components: the red and green opponent component (Cr1), and the blue and yellow opponent component (Cr2).

Perceptual channel decomposition is applied to each of the three components, which consists in splitting the 2D spatial frequency domain both in radial frequency and orientation. A total of 17 maps are created for the achromatic component and 5 maps for each of the chromatic components. Each map can be regarded as a group of primary visual cortex cells tuned to a range of spatial frequency and particular orientation.

There are biological evidences that visual cells only respond to stimuli above a certain contrast, the so-called visibility threshold. Therefore a contrast sensitivity function (CSF) is applied, which expresses the sensitivity³ of human eyes.

Sensitivity can be modulated due to the influence of the context, i.e., the visibility threshold can be increased or decreased by the presence of other stimulus. This effect is

³ Sensitivity is equal to the inverse of the contrast threshold.

called visual masking. This model takes into consideration three kinds of masking: intra-channel, inter-channel, and inter-component. Intra-channel masking occurs between signals having the same features (frequency and orientation), and inter-channel masking occurs between signals belonging to different channels of the same component. Inter-component masking occurs between channels of different components.

2) Perception

This stage of the Le Meur's model provides a structural description of the achromatic component mimicking human perception. This is done by reproducing the behavior of visual cells belonging to the primary visual cortex.

The existence of areas with a sharp color and fully surrounded by areas having quite different colors, implies a particular attraction of focusing on the sharp colored area. It is undeniable that chromatic components guide our attention [7], so a chromatic reinforcement mechanism is used to enhance relevant visual features on achromatic maps by taking into account the chromatic context.

The center-surround suppressive interaction mechanism emulates the action of the non-CRF of visual cells belonging to the primary visual cortex. The effect of the suppressive interaction is maximized when the CRF and surround orientations are the same and reduced or absent when they are very different.

3) Perceptual Grouping

This stage of the model focuses on some aspects on the domain of perceptual grouping, producing a meaningful description of the scene.

Center-surround facilitative interactions enhance the activity of visual cells in the primary visual cortex when the stimuli inside the CRF and within the surround are bound to form a contour. This kind of interactions is modeled and used for contour enhancement. A saliency map is computed by summing directly the outputs of the different achromatic channels.

4) Salient Region Selection

This stage of the model is identical to the last stage of the Itti's model, which has already been described in this section.

In order to assess the results the inventors of this model compared it to Itti's model by analyzing their FOA locations and order, despite not providing the later information in the output images. Given the input image presented in Figure 2.9 (a), the saliency map computed by the Le Meur model is shown in Figure 2.9 (b). The first twenty FOA locations for both Le Meur's and Itti's model are shown in Figure 2.9 (c) and Figure 2.9 (d) respectively. The FOA locations are represented by a red or yellow circle with a 25 pixels radius. When comparing the results a FOA location is considered to be coincident with another, if the distance between them is smaller than the radius of the circle. Under this consideration, 90% of the FOA locations coincide. However, when the ordering of both sets of FOA locations is considered, the percentage of coincident points is null, i.e., the hierarchy of salient locations differs between the two models.

Although this model takes into consideration some details of the HVS that the Itti's model does not, such as visual masking and center-surround facilitative interactions, the FOA locations of both models are similar. Since no information is provided on the order of the FOA locations for the Le Meur's model, no conclusions can be drawn as to which model, Itti's or Le Meur's, produces better ordering for the FOA locations.

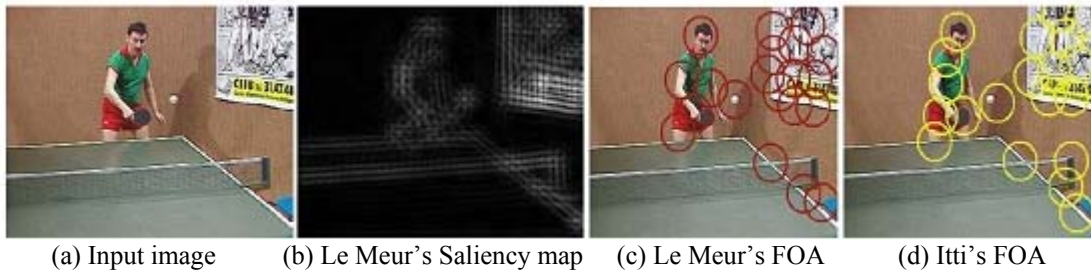


Figure 2.9: Comparison of the Le Meur's and Itti's FOA locations (from [19])

Other Models

Beyond the models that have already been presented in this report, several others have been developed, some of which are now briefly discussed.

For instance, Privitera and Stark [21] evaluated several algorithmic approaches for the detection of regions of interest, comparing the output of such algorithms to eye data captured using standard eye tracking equipment. The algorithm set included: edge detector, center-surround convolution mask, discrete wavelet transform, contrast and orientation operators, etc. The results of their experiences allowed them to conclude that some of the used algorithms could indeed predict human eye fixations. Wavelets, edges and orientation provided good predictions for general images seemed and contrast proved to be a good fixations predictor in terrain images.

Topper [22] introduced an interesting addition to the visual attention literature based on the information theory. In his work, he points out that the strength of a particular feature in an image location does not in itself guarantee that attention will be draw to that region. In an image that has a high degree of variance throughout most of the image one is more likely to attend the more homogeneous regions of the image. Detectors based on the strength of variance or edges would fail. This model suggests that a better approach is to detect parts of the scene that are different from the rest of the scene. In another work, Bruce and Jernigan [23] have reported a similar approach.

2.2.3 Top-down Attention Models

There is converging evidence that the HVS uses a combination of bottom-up and top-down influences to create a saliency map of the visual field [14]. So, in order to simulate with accuracy the HVS attention mechanism top-down models are essential, since they supervise where attention is directed to, based on what information is relevant to the current task.

In this section, we review the two most relevant top-down models: Oliva's model [24], and Frintrop's model [25]. Some other works are briefly described.

A) Oliva's Model

This work proposes a model of attention guidance based on global scene configuration, supported by visual cognition studies that show that humans use context to facilitate object detection in natural scenes. It presents proof that statistics of image features across a natural scene are strongly correlated with the location of a specific object.

The architecture of this model comprehends four modules, as can be seen in Figure 2.10:

- 1) **Feature Extraction:** decomposition of the input image into feature maps, which provide a description of the local image structure.
- 2) **Local Saliency Detection:** a probabilistic definition of saliency is used to detect salient regions of the image.

- 3) **Contextual Priors Computation:** a probabilistic representation of contextual information learns the relationship between context features and the location of an object.
- 4) **Contextual Modulation of Saliency:** contextual priors modulate local saliency to select interesting regions of the image, i.e., regions where a specified object is expected to be.

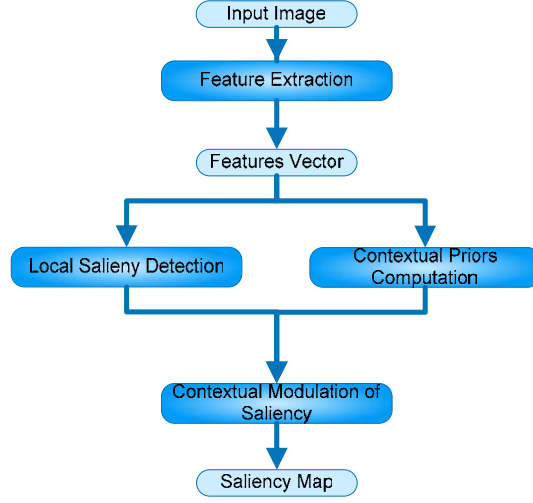


Figure 2.10: Architecture of Oliva's model

Each of the four modules of this model is presented in more detail below.

1) Feature Extraction

Regarding image features, each color component (R, G and B) is decomposed into 4 scales and 4 orientations. A total of 48 feature maps are calculated. For each location, a features vector $\mathbf{v}_l(\mathbf{x}) = \{v_l(\mathbf{x}, k)\}_{k=1, \dots, 48}$ is defined where l stands for local, \mathbf{x} is the location vector and k indexes a feature map tuned to a certain spatial frequency and orientation.

2) Local Saliency Detection

Saliency is defined in terms of the probability of finding a set of local features in an image:

$$S(\mathbf{x}) = p(\mathbf{v}_l)^{-1} \quad (2.4)$$

Saliency is larger in locations where certain local features are more unexpected in the image, and the probability is approximated fitting a Gaussian to the distribution of local features:

$$p(\mathbf{v}_l) = \frac{e^{-\frac{1}{2}(\mathbf{v}_l - \mu)^T X^{-1}(\mathbf{v}_l - \mu)}}{(2\pi)^{\frac{N}{2}} |X|^{\frac{1}{2}}} \quad (2.5)$$

3) Contextual Priors Computation

The contextual features vector \mathbf{v}_c describes the structure of the image. Context is represented using local features $v_l(\mathbf{x}, k)$, taking the absolute value to remove variability due to contrast, and sub-sampling by a factor M:

$$\mathbf{v}(\mathbf{x}, k) = \{|v_l(\mathbf{x}, k)|^2 \downarrow M\} \quad (2.6)$$

In order to further reduce the dimensionality, the image features $v(\mathbf{x}, k)$ are decomposed into the basis functions provided by a principal component analysis:

$$a_n = \sum_{\mathbf{x}} \sum_k |v(\mathbf{x}, k)| \psi(\mathbf{x}, k) \quad (2.7)$$

The functions $\psi(\mathbf{x}, k)$ are the eigenfunctions of the covariance matrix defined by the image features $v(\mathbf{x}, k)$.

The contextual features vector is obtained using the decomposition coefficients of the principal component analysis:

$$\mathbf{v}_c = \{a_n\}_{n=1, \dots, 60} \quad (2.8)$$

The contextual priors probability distribution function (PDF) $p(o, \mathbf{x} | \mathbf{v}_c)$ is learned using a database of images for training. In these images, the location of a specific object o is known. The PDF is modeled using a mixture of Gaussians, and learning is accomplished with the expectation maximization (EM) algorithm [26].

4) Contextual Modulation of Saliency

When looking for an object, saliency based techniques are insufficient to explain the human visual attention mechanism. Human's clearly use a top-down mechanism to locate regions of interest where an object should be, independent of its physical features.

In a statistical framework, object detection can be defined by a probability function which includes contextual information:

$$p(o, \mathbf{x} | \mathbf{v}_l, \mathbf{v}_c) = \frac{p(\mathbf{v}_l | o, \mathbf{x}, \mathbf{v}_c)}{p(\mathbf{v}_l | \mathbf{v}_c)} p(o, \mathbf{x} | \mathbf{v}_c) \quad (2.9)$$

Equation (2.9) defines the probability of the presence of object o at location \mathbf{x} , where \mathbf{v}_l is the local features vector, and \mathbf{v}_c is the contextual features vector. The probability can be decomposed into three factors: the object likelihood $p(\mathbf{v}_l | o, \mathbf{x}, \mathbf{v}_c)$, the local saliency $p(\mathbf{v}_l | \mathbf{v}_c)$, and the contextual priors $p(o, \mathbf{x} | \mathbf{v}_c)$. The object likelihood term, which represents information on the appearance of an object, is not considered in this model to avoid the use of a specific model for object appearance; so Equation (2.9) can be rewritten into

$$S_c(\mathbf{x}) = \frac{p(o, \mathbf{x} | \mathbf{v}_c)}{p(\mathbf{v}_l | \mathbf{v}_c)} = S(\mathbf{x}) p(o, \mathbf{x} | \mathbf{v}_c) \quad (2.10)$$

Equation (2.10) does not require any information regarding the distribution of features of the target. Local saliency, $S(\mathbf{x}) = p(\mathbf{v}_l | \mathbf{v}_c)^{-1}$, provides a measure of how unlike it is to find a set of local measurements within context \mathbf{v}_c , and can be approximated by a distribution of local features within the image as in Equation (2.4).

Equation (2.10) allows the calculation of contextual saliency, multiplying local saliency by contextual priors, and providing as an output a saliency map which highlights regions of the image where the *a priori* specified object o is expected to be.

In order to test the model, the inventors trained the PDF $p(o, \mathbf{x} | \mathbf{v}_c)$ to predict the location of people, and recorded human eye movements when subjects were instructed to count the number of people in the presented scene. By comparing the pattern of eye fixations provided by the contextual model with the recorded human eye movements, the inventor verified that they are similar, and therefore the proposition that top-down information from visual context modulates the saliency of regions during the task of object detection is validated.

B) Frintrop's model

This model performs saccades guided by top-down influences, and then analyzes the FOA locations in further detail to determine if a specified object is in that region. The model consists of two stages, as illustrated in Figure 2.11:

- 1) **Top-Down Guided Attention:** a saliency map is generated using top-down influences, providing FOA locations for further analysis.
- 2) **Object Recognition:** the FOA locations provided by the previous module are analyzed to recognize a specific object.

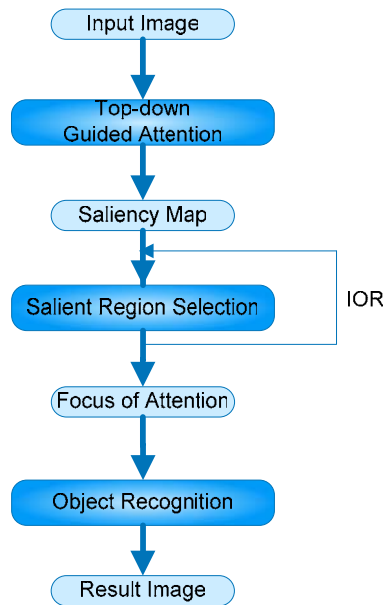


Figure 2.11: Architecture of Frintrop's model

The two stages of the model are explained in detail below.

1) Top-Down Guided Attention

This stage is based on Itti's bottom-up model described in section 2.2.2, but top-down cues are used to enhance the features which are relevant considering the object that is to be identified; this means that the brightest areas of the output saliency map correspond to regions where the object to be identified is likely to be.

The implementation of this stage is influenced by Itti's model but there are some remarkable differences that are now described in detail. First of all, as already mentioned top-down information is used, meaning that specific features of the object to be identified influence the weight of the respective feature conspicuity maps. When all the conspicuity maps are summed to obtain a saliency map, regions with the specific features of the object have a high saliency value. Regarding color, instead of using only two color channels, a Hue-Saturation-Value (HSV) color-space is used to compute six color contrasts: red, yellow, green, cyan, blue and magenta. The center-surround filter mechanism also has a different implementation. In Itti's model center-surround information is computed as the difference between fine and coarse scales; in this model center-surround information is computed by determining the surround of every pixel separately as the average of the surrounding pixels and subtracting the value of the center pixel and its surround. The final part of this stage determines FOA locations, and uses an IOR mechanism similar to Itti's model.

2) Object Recognition

The purpose of this stage is to identify objects near the FOA locations provided by the previous stage.

A classifier, developed by Viola and Jones [27], is responsible for recognizing the specified object. To do this, the features of the object, such as edges or lines are learnt creating a representation of the object. A decision tree with several levels, each one identifying some basic features of the object, is used to recognize the object using the previously learned representation.

To investigate the performance of the system the inventors trained the model to find name plates in an office environment. Furthermore, since cyan was present in the name plates, the cyan feature was given a bigger weight acting as a top-down cue. In 38 images, one of the first 5 FOA laid on the name plate, as Figure 2.13 (a) shows. Furthermore, in 25 images, the name plate was the first FOA. When the classifier was applied only to the region of the first FOA, Figure 2.13 (b), 24 of the 25 focused name plates were recognized. The classifier was also applied to the whole image performing exhaustive search, missing 2 detections and presenting 9 false detections. These results show that top-down guided attention allows to speed up object recognition by restricting the recognition to about 30% of the image, and at the same time detection is more exact than in exhaustive search.



Figure 2.12: Results obtained with the Frintrop's model searching for name plates

Other Models

Itti and Miao [28] proposed a visual attention model combining attention orientation and object recognition. The model is structured in two modules:

- The first module performs bottom-up processing, selecting the most salient locations from the input image, as described in Itti's model in section 2.2.2.
- In the second module, the object recognition model HMAX from MIT [29] is applied to the locations selected by the first module, identifying objects such as cars, paper clips, faces, etc.

The work allowed the inventors to conclude that valuable computation time can be saved by using a bottom-up model to guide a more 'expensive' object recognition model, since recognition is only performed on a few selected locations.

A computational model of depth-based attention was presented by Maki et al. [30], which consists of a parallel stage with pre-attentive cues followed by a later serial stage where the cues are integrated. The cues are relative depth, image flow and motion. The criterion for target selection is the relative depth between objects, since attention is maintained to the closest moving object. The obtained results show that simple algorithms can compute depth information in a fast manner, and therefore it is possible to selectively mask out different moving objects in real scenes and fixate on the closest moving object for some time.

A model that selects the most salient regions in the visual scene based on prior knowledge of similar scenes was introduced by D. Chernyak and Stark [31]. The use of Bayesian conditional probabilities for each region given the scene category allows the prediction of the informative value of that region, providing a saliency map of the image. This map provides an eye movement control module with an ordered set of potential image regions to fixate on.

2.3 Computer Vision Applications

Visual attention modeling can be very useful in several applications such as image adaptation, object recognition, video compression, robot navigation, etc. Thus, just only some of them were chosen to be mentioned here as examples.

In adaptive content delivery for universal access as addressed in this report, one of the essential problems is image adaptation. A method for adapting images based on user attention was proposed by Chen et al. [32]. This method dynamically modifies the image contents adapting it to the different size of client's screen devices. This is done by determining the image's most important regions, through saliency, face and text attention models. Afterwards, an algorithm is used to find the optimal adaptation, allowing the delivery of the most important image regions to the user, given the screen size limitations.

Face detection and recognition are two tasks which are very useful in many applications, such as man-machine interactions [33]. An obsolete form to detect and recognize faces in an image is through a blind search, this means, sequentially shifting, rotating, and resizing a search window over the whole image and testing whether that window contains a face. Siagian and Itti [34] presented a biologically-inspired face detection system which uses a saliency model and a gist⁴ model. Once again, the visual attention model appears as a very important way to improve the performance of a certain system. In this case, a saliency map is obtained through bottom-up processing, and used to focus the system only on smaller regions of the image, improving the overall speed and accuracy of face detection.

In the video compression research area, the neurobiological model of visual attention also proves itself useful. Itti [35] developed an algorithm which uses regions of interest in video streams to achieve video compression. The algorithm selects regions of high saliency in video inputs and classifies them as priority regions. Compression is achieved by coding the regions of interest with higher quality than the regions outside them reaching a higher subjective impact.

2.4 Final Remarks

This chapter demonstrated that visual attention plays a fundamental role in the HVS, selecting regions of interest for high level analysis by the fovea. This mechanism overcomes the human brain processing limitations, and is guided by bottom-up and top-down influences.

Visual attention models are an important tool for complex computer vision systems. As an example, these models can guide such systems to specific areas of the image, thus avoiding processing the entire image.

Several computational models that emulate the human visual attention mechanism have been presented in the literature. There is growing evidence that a complete model of attention must provide an interaction between bottom-up and top-down processing.

The following chapters of this thesis describe a proposal for a visual attention driven image to video adaptation system, which uses bottom-up and top-down processing.

⁴ Rapid but rough analysis of the entire scene yielding a small number of scene descriptors.

Chapter 3

Image2Video Adaptation System

In the last decade, multimedia consumer electronics have become quite heterogeneous, with the arrival and rapid globalization of PC's, mobile phones and PDA's. Television is no longer the only way to access multimedia content. Given the diversity of terminals and users, it is becoming more and more important to have content adaptation tools that are able to fulfill both the needs of different users and usage conditions, in order to maximize user satisfaction.

In this framework, an adaptation system has been developed in this work, whose objective is represented in Figure 3.1. The adaptation system is capable of determining the regions of interest in an image based on some knowledge of the human visual attention mechanisms and generating a video sequence that attends those regions according to certain user preferences, while taking into consideration the limitations of the display's size. User preferences refer to the video display modes the user can choose for the visualization of the adapted video sequence, e.g. video duration.

In practice, the so called Image2Video system is able to transform/adapt images to video driven by visual attention targeting a final better user experience. This is particularly critical when the images are large and the display size small.

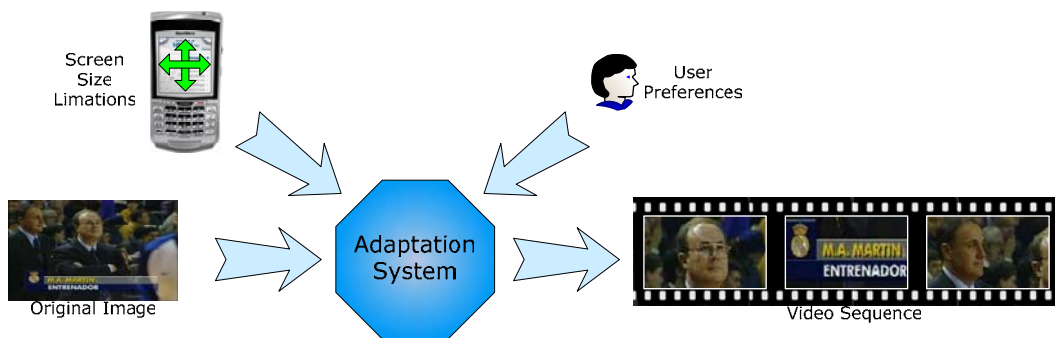


Figure 3.1: Image2Video adaptation system

In this chapter, the main architecture of the proposed adaptation system is presented, as well as the background algorithms used, this means the algorithms that have been integrated in the application but not developed. In the next chapters, the algorithms fully developed by the authors of this report will be presented in more detail.

3.1 Image2Video Adaptation System Architecture

The developed adaptation system is greatly inspired on the knowledge of the attention mechanism of the HVS to determine ROIs in the image, and it uses a multi-stage architecture to perform all the necessary tasks to transform the original image into a (more interesting) video clip.

The multi-stage architecture proposed for the adaptation system is presented in Figure 3.2; it was conceived to produce a high-level description of the most interesting contents in the original image and then combine that description with the user preferences and terminal device limitation descriptions to perform the image-to-video transmoding. Transmoding refers to all media adaptation processes where content in a certain modality is transformed into content in another modality, e.g. video to images, text to speech.

The proposed architecture includes four stages, with the first two being responsible for the determination of a map that identifies all the ROIs of the image, and the remaining two for the generation of the video that displays the image following a path that links the various ROIs to each other (see Figure 3.2).

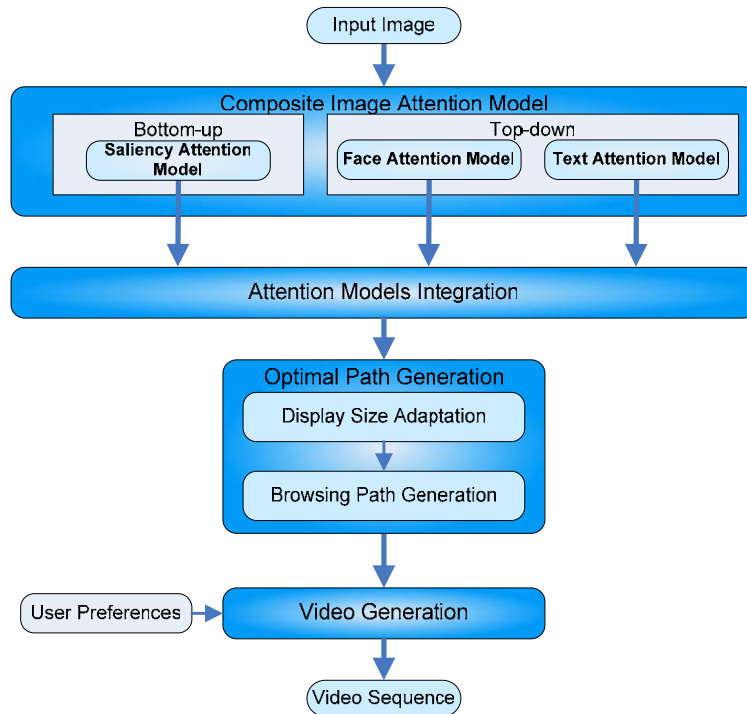


Figure 3.2: Image2Video system main architecture

The main objectives of the four main architectural stages are now presented:

- 1) **Composite Image Attention Model:** the HVS attention mechanism, whose importance has been demonstrated in section 2.1.3, is able to select ROIs in the visual scene for additional analysis; the ROIs selection here is guided by bottom-up and top-down approaches. Based on the knowledge of the human visual attention mechanisms, a composite image attention model has been developed to detect ROIs and provide a measure of the relative importance of each one. The model integrates three elementary visual attention models. The first stage of these visual attention models uses the background analysis algorithms described in section 3.2 to detect the ROIs in the images. The selected three elementary visual attention models are:
 - Saliency attention model: the objective of this model is to identify ROIs without specific semantic value associated objects, i.e. regions with different statistical

properties from the neighboring regions are considered ROIs. To achieve this, the first stage of this model comprises Itti's bottom-up algorithm, which is presented in section 3.2.1. The complete architecture of the model is described in section 4.1.1.

- Face attention model: the objective of this model is to identify ROIs that contain faces. The detection of faces is a task performed daily by humans since they are one of their most distinctive characteristics, providing an easy way to identify someone. Therefore faces are one of the semantic objects present in an image that are more likely to captivate human's attention. The first stage of this model uses a face detection algorithm presented in section 3.2.2 and the complete architecture of the model is described in section 4.1.2.
 - Text attention model: the objective of this model is to identify ROIs that contain text. People spend a lot of their time reading, may it be newspapers, e-mails, SMS, etc. Text is a rich font of information, many times enhancing the message that an image transmits. Therefore text is a kind of semantic object that attracts viewer's attention. The first stage of this model uses a text detection algorithm presented in section 3.2.3 and the complete architecture of the model is described in section 4.1.3.
- 2) **Attention Models Integration**: the ROIs computed by the three elementary attention models in the previous stage are integrated into a single image map, which contains all the ROIs, their location and type (saliency, face, text). A method has been implemented to solve the cases where overlapping exists between different types of ROIs, which is presented in section 4.2.
 - 3) **Optimal Path Generation**: this stage is responsible for generating the path used to display with video the whole image, i.e. the path that transforms the image into video. Two mechanisms are used for this:
 - Display Size Adaptation: given the display size limitations, the ROIs are adapted to fit into it, i.e. they can be split into blocks that fit into display, they can be grouped with other ROIs to increase the presented information or simply remain as they are. This mechanism is explained in section 4.3.1.
 - Browsing Path Generation: this mechanism is responsible for the determination of the browsing path, which takes into consideration the spatial distribution of the ROIs and their type, targeting the provision to the user of the best possible video experience. This mechanism is explained in section 4.3.2.
 - 4) **Video Generation**: the last stage of the adaptation system is responsible for creating a video sequence based on the previously calculated browsing path, a set of directing rules and user preferences, and is presented in detail in section 4.4.

3.2 Background Algorithms

In this work, three major processing algorithms have been integrated in the Video2Image application which had been previously implemented and thus were not developed by the authors of this work. These algorithms are used in the elementary visual attention models to detect ROIs. The integration of these algorithms into the adaptation system was not straightforward, i.e. a lot of work was done to adapt them to the needs of this project.

All of the software in this work runs with Microsoft .NET Framework 1.1, which is an integral Windows component for building and running the next generation of software applications. Originally, the background algorithms were not developed to run with the .NET Framework. Therefore, in order to integrate them into the adaptation system, changes had to be made in the code so that they could be compiled to run in the .NET Framework.

Also the analysis algorithms were developed to run in the command-line and the parameters were fixed, i.e. the code had to be recompiled each time a parameter was changed. Therefore, a custom function has been developed to allow adjusting the different parameters of the algorithms in the developed application interface, without having to recompile the code.

In the following sub-sections, these algorithms are presented, as well as the changes that were introduced in each one so that they provide the desired results in a suitable format for further processing by the elementary visual attention models presented in the next chapter.

3.2.1 Saliency Detection Algorithm

Itti et al [18] have developed and given public access to the ilab neuromorphic vision C++ toolkit (iNVT), which is a comprehensive set of C++ classes for the development of computational neuroscience algorithms whose architecture and function is closely inspired from biological brains. This toolkit comprises the full-implementation of the bottom-up model already presented in section 2.2.2, and it has been integrated into the developed adaptation system proposed in this report to detect ROIs based on their saliency; this means that regions with different properties from the neighboring regions are considered to be ROIs.

This model has been adopted because the tests carried out demonstrated that the detected ROIs, in our subjective opinion, do in fact represent regions of the image that attract our attention. Furthermore, the order of importance attributed to the ROIs also seems adequate.

The toolkit provides several options regarding the different mechanisms that compose the bottom-up model. The tests that were carried out, allowed defining the default configuration for the different mechanisms, which are now presented (see also section 2.2.2):

- **Center-surround mechanism:** in this work, the algorithm uses center scales 1 through 3, center-surround scale differences of 2 through 5, and the saliency map is built at scale 3 (configuration 1-3, 2-5, 3). This configuration tweaks the system towards being more sensitive to smaller objects than using the default configuration 2-4, 3-4, 4.
- **Normalization operator:** the toolkit provides two methods, Maxnorm and Fancy, for map integration. The first method provides smooth and continuous saliency maps; the second method is an iterative procedure that yields the sharpest, best separated ROIs. Therefore, the Fancy method is used in this work, as it provides a good basis for the shape estimator procedure that is explained next.
- **Shape-estimator:** this method proved itself useful since it provides a rough estimation of the objects surrounding the saliency point, as opposed to having a fixed circular FOA which only indicates a region in the image. This method can extract the shape from the saliency map, the feature or conspicuity maps which provide the greater contribution to the saliency of the region. There are two options available to smooth the estimated shape: Gaussian and Chamfer. The second is quicker, but the first provides a shape which is smoother and better adapted to the objects. Therefore, by default, the algorithm extracts the shape from the winning feature map and uses the Gaussian smoothing method.
- **IOR mechanism:** a region in the saliency map can be inhibited based on the shape that has been estimated or a fixed disc size region. The IOR mechanism that proved best was the disc based. The disc size is automatically set to the value obtained by $\max(imageWidth, imageHeight)/12$.

The tests performed allowed verifying that limits regarding the number of ROIs and their identification time have to be defined to avoid the algorithm choosing regions of the image without interest. Therefore, three stop conditions have been defined for the algorithm:

- The maximum time difference between the identification of ROIs is set to 250 ms. Large time intervals indicate the model is trying too hard to choose a ROI, and in most cases

will end up choosing a region without interest. This condition was not available in the original algorithm; this means that it was implemented by the authors of this project.

- The maximum analysis time of the image is set to 500 ms, which is considered to be enough for a user to identify the ROIs of the image.
- The maximum number of ROIs is set to 8, which is considered to be an adequate number of ROIs to be detected in an image.

Figure 3.3 presents scenes analyzed by this bottom-up algorithm, where the yellow shapes represent the ROIs in the image, and the red line connects them by saccades. The church scene has only two ROIs because the first stop condition is verified. The basketball coach scene has five ROIs obtained before the second stop condition is verified. These results demonstrate that the algorithm provides the locations of interesting regions of the image, such as the church in Figure 3.3 (a) and the people in Figure 3.3 (b).

Since the available software implementing the presented algorithm only provides image outputs, the authors of this work had to introduce a method to save the location and shape coordinates of the detected ROIs in a file. This provides a better way to process the ROI information in the following stages of the visual attention model into which this elementary model will be incorporated.

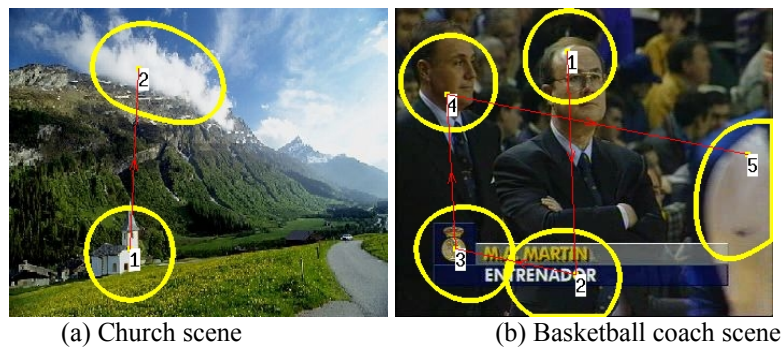


Figure 3.3: Example of attention results generated by Itti's bottom-up model

3.2.2 Face Detection Algorithm

The algorithm here presented was developed and implemented by João Ascenso et al [36] of the IT Image Group, who kindly provided the software implementation to be used in this work. The algorithm finds the location of people's faces in an image, relying on prior knowledge about their color and shape, and consists of three stages, as Figure 3.4 shows:

- 1) **Color Analysis:** this stage identifies an initial set of regions from the input image with skin-like color.
- 2) **Shape Analysis:** this stage performs ellipse detection for the regions with skin-like color, as it recognizes that an ellipse can approximate the outline of a human face shape.
- 3) **Face Selection:** this stage decides if the candidate ellipses provided by the previous stage are faces or not by applying certain constraints.



Figure 3.4: Architecture of the used face detection algorithm

The three stages of the facial detection algorithm are now described in detail to explain how the face description is obtained, which provides the location and size of the detected faces.

1) Color Analysis

The goal of this analysis stage is to identify the skin-color regions present in the image using a color segmentation method. The human skin-color is one of the most robust face features regarding variations that occur in natural images, such as: image conditions, including size, lighting, viewpoint; face appearance, including use of glasses, beard, makeup; or image contents, including background complexity, occlusion by other objects and number of faces.

The efficacy of the color segmentation method depends essentially on the color space that is chosen. This algorithm uses the HSV color space as it produces a good separation between skin-color and other colors with varying illumination.

The method used for color segmentation belongs to a class of techniques called pixel-based segmentation, which rely on the a priori knowledge of the distribution of the skin-color. The inventors of this algorithm used a set of over 80 images as the training set for a Caucasian race skin-color distribution model, which defines a polyhedron in the HSV space. Pixels within the polyhedron are classified as skin-color while the others pixels are classified as belonging to the background.

A median filter is applied to the output of the color segmentation method in order to remove unwanted isolated pixels, while preserving the general spatial appearance. Median filters are commonly used since they perform well in the task of binary noise removal. A contour extraction step is performed to identify the boundaries of candidate face regions. Figure 3.5 (b) presents an example of the results produced by this stage for the input image shown in Figure 3.5 (a). After the color analysis stage, an image which contains the contours of the skin-color regions of the initial image is obtained. It is visible that the faces, the hands and part of the Oscars are selected as skin color regions.

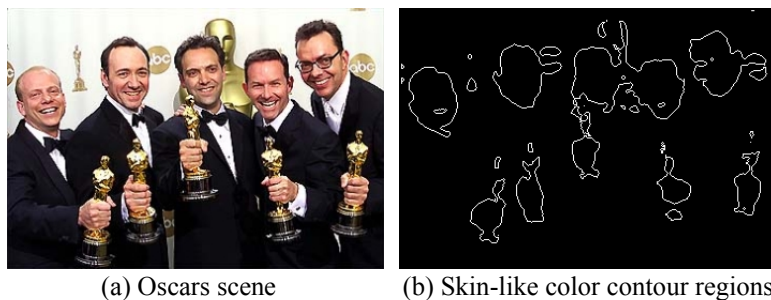


Figure 3.5: Example of color analysis stage results

2) Shape Analysis

This stage is based on the idea of fitting ellipses to the results of the color segmentation stage, selecting some candidate ellipses, and deciding after if they correspond or not to faces by applying certain constraints.

The method to detect elliptical shapes in images is based on the randomized Hough transform (RHT). This method allows the detection of regions whose shape is unknown, but for which there is enough prior knowledge to build an approximate model of the ellipse. This method is able to recognize partial and slightly deformed ellipses, and it may find several occurrences of ellipses in the same iteration. It consists of three major steps:

- Find the ellipses present in the image, limited to a pre-defined maximum number of ellipses.
- For each ellipse found, determine the parameters that describe it, namely the coordinates of the centre, the major and minor axis and the orientation.
- Classify each pixel in the image as belonging to one of the ellipses found or as background pixel.

The implementation of the RHT consists in a series of random trials. For each trial, a triplet of pixels belonging to the same skin-color contour connected component is randomly sampled from the image, and the parameters of an ellipse are calculated. These parameters are recorded in a structure called accumulator. The idea is to accumulate evidence for different ellipse parameters, and the most likely set of parameters will define the candidate ellipses. Each candidate ellipse has a pre-defined width of 10 pixels called band, which forms the elliptical shape shown in Figure 3.6.

3) Face Selection

Since some of the candidate ellipses do not always correspond to faces because face-specific constraints were not applied in the selection of the ellipses, it is necessary to filter the candidate ellipses according to adequate criteria, and select the faces detected. Face selection consists on checking, for each candidate ellipse, if the following criteria are verified:

- Angle sector criterion: evaluates the amount of contour skin-color processed pixels under the band of each candidate ellipse in order to check if it corresponds to a face. The candidate ellipse is divided into a limited number of angular sectors, and for each sector it is checked if it contains contour skin-color processed pixels or not. The ratio between the number of sectors containing at least one pixel and the total number of sectors gives an indication if the candidate ellipse is adequately covered by contour pixels in all its extension. This criterion is checked if this ratio is equal or bigger than an ellipse acceptability threshold, which by default is 75%.
- Orientation criterion: evaluates if a real ellipse is under consideration by calculating for each contour that the candidate ellipse band contains, the angle between the normal to the ellipse at that pixel and the line that links it with the ellipse centre. If the angle is less than a specified threshold, this pixel is valid. Candidate ellipses are considered valid if the ratio between the number of valid pixels and the total number of pixels is inferior to 50%.
- Aspect ratio criterion: given the geometry of the human face, and according to experimentation with the same images used to obtain the skin-color distribution model, the aspect ratio between the major and minor axis of the ellipse must fall in the range of 1.25 to 2.

Figure 3.6 shows the five ellipses that were fitted to the skin-color contour regions, i.e. the regions that were selected as faces from the input image presented in Figure 3.5 (a). The hands and Oscars are eliminated since no ellipse could be well fitted to them. Beside the image outputs, the algorithm creates a text file with the parameters of the ellipses that are considered to be faces.



Figure 3.6: Example of results generated by the face detection algorithm

3.2.3 Text Detection Algorithm

The algorithm here presented was developed and implemented by Duarte Palma [37] of the IT Image Group, who kindly provided the software implementation to be used in this work.

The original objective of this algorithm is the automatic extraction of text in digital video, but it has been adopted in this work to find text in images. The algorithm is based on the spatial segmentation of the image, contrast analysis between the text and the image background and geometrical analysis of the text. As Figure 3.7 shows, the algorithm consists of four stages:

- 1) **Image Simplification:** the objective of this stage is to diminish the influence of noise or a too high number of colors on the detection of the text present in the image.
- 2) **Image Segmentation:** this stage is responsible for splitting the image into homogeneous regions based on texture data (luminance), providing candidate character regions.
- 3) **Character Detection:** the purpose of this stage is to classify the candidate regions provided by the previous stage as being text or not.
- 4) **Word Formation:** this stage is responsible for grouping the characters detected by the previous stage to form words.

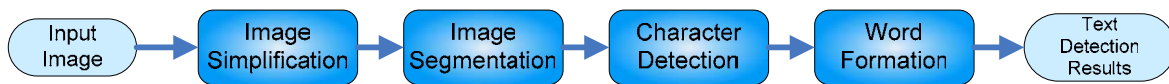


Figure 3.7: Architecture of the text detection algorithm

The four stages of the algorithm are now described in detail.

1) Image Simplification

In this algorithm, image simplification is done trying at the same time to preserve as much as possible the original edges present in the image, since edges are very important for character detection.

To achieve this, an iterative method is used, which applies an edge detector followed by a median filter. In order to preserve the detected edges, the median filter is applied only in the regions where edges haven't been detected. The median filter is applied to remove noise present in the image, which can cause over segmentation of the image in the next stage of the algorithm.

Figure 3.8 (b) presents the outcome of the image simplification stage for the input image shown in Figure 3.8 (a). As can be seen in Figure 3.8 (b), this stage transforms the input color image into a black and white format.

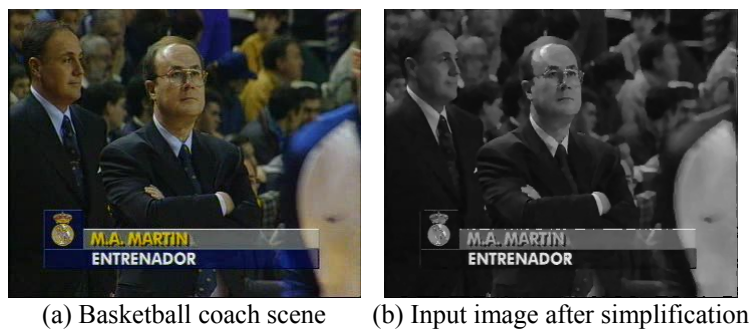


Figure 3.8: Example of image simplification stage results

2) Image Segmentation

This stage performs a hierarchical decomposition of the image in order to determine candidate character regions. The method used adopts a split and merge approach.

The split phase has the objective to form an initial set of regions that will reduce the computational effort associated to the merge phase. It consists on the successive division of the image in four square sub-regions; the criterion to stop dividing a region is:

- The dimension of the region equals one pixel.
- The region fulfils the homogeneity criteria defined by the dynamic range of luminance, i.e., $DR = \max\{R\} - \min\{R\} < Th_{split}$.

The threshold value for the split phase, Th_{split} , is defined based on the trade-off between computational speed and segmentation precision, i.e. as Th_{split} increases, so does the speed of the merge phase. However, values of Th_{split} that are too high can lead to the fusion of several regions. The extensive experiments carried out demonstrated that $Th_{split}=30$ provides good results, and therefore it is the pre-defined value in this report.

The merge phase has the objective of merging the adjoining regions provided by the split phase, which are sufficiently similar according to the adopted homogeneity criteria. The merge of regions is realized iteratively, by successively merging pairs of adjoining regions which together form a homogeneous region. This phase stops when it isn't possible anymore to merge any adjoining regions to form a homogeneous region. The pixels forming a new region are attributed the value of the average of the luminance of the two regions that originated it. The homogeneity criteria is also based on the dynamic range, but in this case two adjoining regions are merged if $\max\{R_a \cup R_b\} - \min\{R_a \cup R_b\} \leq Th_{merge}$ where Th_{merge} is the threshold value. The extensive tests that were carried out demonstrated that $Th_{merge} = 35$ provides good results, and therefore it is the pre-defined value in this work.

Finally, an additional technique is applied to remove small regions which exist mainly due to the presence of noise. Small regions that share an edge with more than one region contribute to the degradation of the character boundaries. In order to identify small regions for posterior merging and improve the character edges, a technique that combines edge detection with dominant contrast local orientation is used.

The regions represented in Figure 3.9 are the candidate character regions, obtained as the result of this stage when applied to the simplified image shown in Figure 3.8 (b).



Figure 3.9: Example of image segmentation stage results

3) Character Detection

In order to detect each text character, the regions provided by the previous segmentation stage are filtered based on geometrical restrictions, related to their height, width, height-to-width ratio and compactness defined by the area-to-bounding⁵ box ratio of a region. The threshold values used depend on the range of character sizes selected for detection.

⁵ The bounding box is the smallest possible rectangle that can involve the whole region.

Figure 3.10 presents the regions that are considered to be characters, i.e. the candidate character regions from the previous stage that fulfilled the geometrical restrictions.



Figure 3.10: Example of character detection stage results

4) Word Formation

The word formation technique is based on the spatial analysis of each region previously classified as text.

To form words, it is considered that text consists of groups of characters aligned in a certain direction with characters sufficiently close to each other to form words. Furthermore, the regions must have a minimum difference in height and similar luminance values. And most importantly a word has to include at least three regions, i.e. three characters. The full details of the word formation stage can be found in [37].

Figure 3.11 shows the text detection results obtained for the basketball coach scene which aren't perfect in particular for the first line.



Figure 3.11: Example of text detection results for the image shown in Figure 3.8 (a)

3.3 Final Remarks

In this chapter the Image2Video adaptation system has been presented. Based on the knowledge of the human visual attention mechanisms, this system is capable of analyzing an input image to determine its ROIs, this means the regions that are more likely to attract human's attention. The system then creates a video sequence that attends the ROIs, targeting a final better user experience. The visual experience improvement is particularly noticeable when the images are large and the display size small.

The complete architecture of the Image2Video adaptation system and the main objectives of the several stages that compose it have been introduced, and will be explained in detail in the next chapter.

Chapter 4

Processing for Image2Video

Adaptation

This chapter is dedicated to the presentation of the four main processing modules that compose the architecture of the developed Image2Video application, each one described in a separate section. The objective of this chapter is to provide a detailed description of the processing solutions but also of the principles and reasoning behind the algorithms that have been developed and implemented to achieve the objective of this work: creating a video sequence that provides a better user experience for an image by attending the ROIs in a visual attention driven way.

In order to develop the algorithms, define criteria and thresholds, the authors of this project performed exhaustive tests with several kinds of images. These tests allowed subjectively evaluating the results of the algorithms, criteria and thresholds, and validating their performance in the adaptation system.

4.1 Composite Image Attention Model

The first stage of the Image2Video system performs a content analysis of the input image to determine the ROIs of the image. In order to do this, the developed composite image attention model uses three elementary visual attention models: saliency attention model, face attention model and text attention model. These models were selected because salient regions, i.e. regions with different (statistical) properties from the neighboring regions are considered more informative and are supposed to attract viewer's attention. Faces and text are distinctive kinds of semantic objects; the first allows identifying someone and the second is a rich font of information. Therefore they are likely to attract viewer's attention.

The selected visual attention models provide globally a description of the objects present in the image that are considered to attract viewer's attention. Based on the work developed by Chen et al. [32], which proposes a method for adapting images based on user attention, the visual attention models provide a set of attention objects (AOs):

$$\{AO_i\} = \{(ROI_i, AV_i)\}, \quad 1 \leq i \leq N \quad (4.1)$$

Frequently, an AO represents an object with semantic value, such as a face, a line of text or a car, meaning that it carries information that can catch the user's attention. Therefore the i^{th} attention object within the image, AO_i , has two attributes: the ROI_i , which is the region of the image that contains the AO_i ; and the attention value (AV_i), which represents an estimate of the user's

attention on the AO. The basis for the AV is that different AOs carry different amounts of information, so it is necessary to quantify the relative importance of each one. The AVs range from 0 to 100%. N is the total number of AOs in the image.

4.1.1 Saliency Attention Model

This visual attention model is based on the fact that certain regions of an image attract human's attention because they have different visual features compared to those of the surrounding area. Its objective is to identify ROIs without any specific semantic value associated. The saliency attention model has four stages, as can be seen in Figure 4.1:

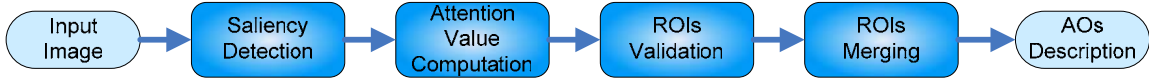


Figure 4.1: Architecture of the saliency attention model

The four stages of the model are now presented in detail.

1) Saliency Detection

This stage of the model corresponds to Itti's bottom-up model presented in section 2.2.2, and whose algorithmic implementation has been presented in section 3.2.1. This stage provides an output image map with the ROIs of the input image, and a description file containing the location coordinates and identification time of each ROI.

2) Attention Value Computation

In order to have the full description of the AOs present in the image, it is necessary to compute its AV. In this model, the AV is determined based on the time difference between the identification of ROIs, which can be done using the following equation:

$$\begin{cases} AV_1 = 100 \\ AV_i = AV_{i-1} - \left(\frac{T_i - T_{i-1}}{\Delta T_{\max}} AV_{i-1} \right) \end{cases} \quad (4.2)$$

T_i is the instant of time when the saliency detection algorithm identifies the ROI_i and ΔT_{\max} is the maximum interval of time that can elapse between the identification of two ROIs. This interval is fixed to $\Delta T_{\max} = 250\text{ms}$, for the reasons already pointed out in section 3.2.1. As said before, the identification time is only dependent on the visual attention model and has nothing to do with processing times in the adopted platform.

Equation (4.2) attributes an AV of 100 to the most important AO, while to the following AOs an AV_i is attributed which decreases relatively to the previous AV_{i-1} according to the time difference between the identification of AOs, normalized by ΔT_{\max} .

3) ROIs Validation

Since the saliency detection stage provides several ROIs, it is necessary to evaluate if all should be considered in the remaining stages of the visual attention model. This stage filters AOs according to their importance relatively to the most important AO, using as filtering criteria a threshold ratio:

$$\frac{AV(AO_i)}{\max AV(AO_j)} \geq 0.10 \quad j = 1, \dots, N \quad (4.3)$$

An AO with a relative importance ratio higher than 10% is validated; otherwise, it is eliminated. This allows eliminating AOs that have small importance, consequence of the long time it took them to be identified by the saliency detection stage.

Figure 4.2 (b) shows the AOs annotated by a blue rectangle, the so-called bounding box, after validating the AOs provided by the saliency detection stage presented in Figure 4.2 (a). In this example only the fifth AO doesn't verify Equation (4.3), and therefore is eliminated.

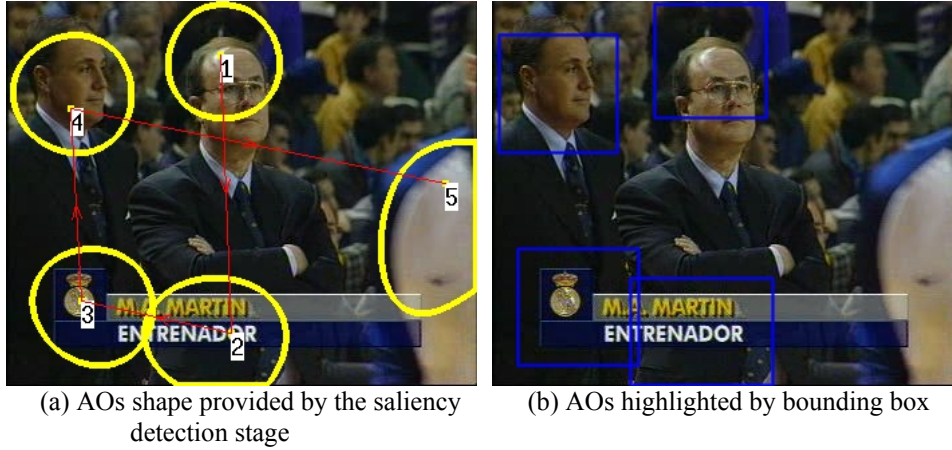


Figure 4.2: Example of results generated by the AOs validation stage

4) ROIs Merging

Sometimes there are ROIs that overlap each other, in which case it is necessary to decide if they correspond to the same saliency region, i.e. if they should be merged into a unique ROI. When ROIs are merged, the new ROI is represented by the smallest bounding box that can contain both ROIs which means that the merged ROI may be much larger than the simple addition of the 2 merged ROIs. This applies to all the merged ROIs that are mentioned in this report.

In order to decide when ROIs should be merged, several experiments were realized, which has lead to the definition of the following criteria:

- Saliency overlap criterion: ROIs in this model exist due to their saliency, i.e. they are different from their surroundings and therefore have a unique semantic value. Therefore ROIs should be merged only if the overlapping area is big, otherwise they should remain independent. Equation (4.4) shows the criterion to be checked for merging; this means that if the overlap area of the ROIs is equal to or bigger than 50% compared to the ROI with the biggest area, they should be merged into a unique ROI.

$$\frac{area(ROI_i \cap ROI_j)}{\max \{area(ROI_i), area(ROI_j)\}} \geq 0.5 \quad (4.4)$$

- Saliency useful area criterion: it is important to avoid that a merged ROI is created with big areas being made of regions that are not ROIs. Equation (4.5) shows the criterion to be checked; this means that the area of the two ROIs must represent at least 70% of the merged ROI.

$$\frac{area(ROI_i) + area(ROI_j) - area(ROI_i \cap ROI_j)}{area(ROI_{merged})} \geq 0.7 \quad (4.5)$$

When both criteria are verified, ROI_i and ROI_j are merged into a unique ROI. The new AO has an $AV = \max\{AV_i, AV_j\}$. Figure 4.3 presents two example results of the merging stage. In the first case, the criteria is verified; therefore the ROIs presented in Figure 4.3 (a) are merged into the ROI presented in Figure 4.3 (b). In the second case, presented in Figure 4.3 (c), the ROI at the center of the image containing the sun overlaps the ROI including the see line, but they remain independent; this happens because the useful area criterion isn't fulfilled.

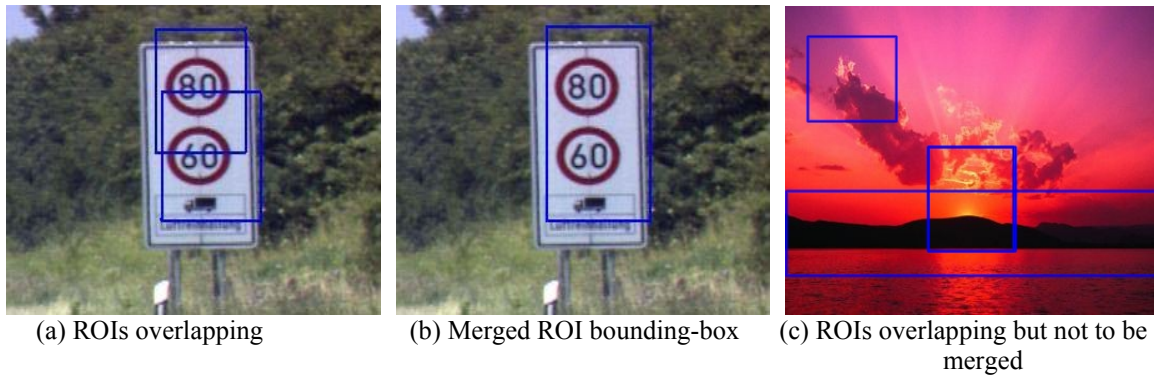


Figure 4.3: Example of results generated by the ROIs merging stage

Figure 4.4 shows the structure used for the developed merging algorithm to detect and solve the cases where overlapping exists between saliency ROIs.

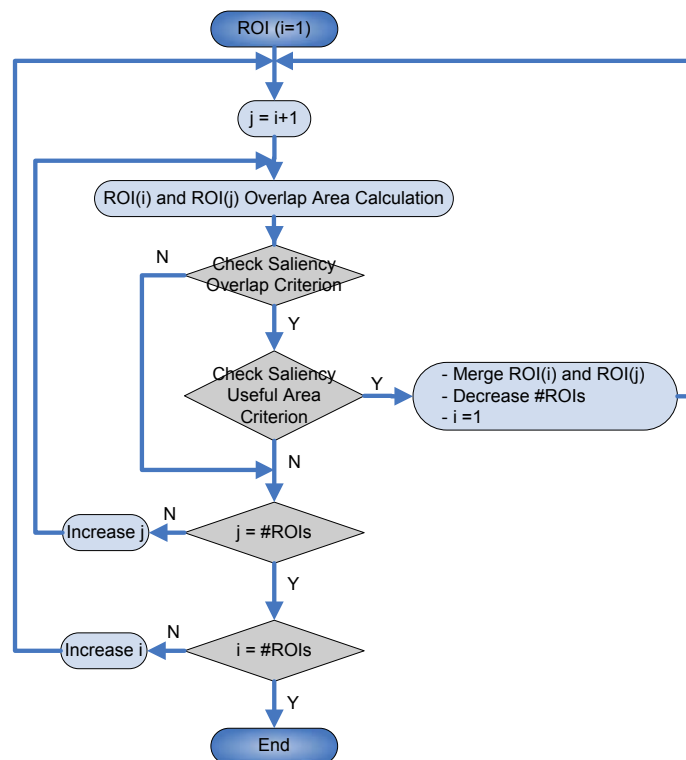


Figure 4.4: Structure of the saliency ROIs merging algorithm

The merging algorithm tests all the possible combinations of ROIs pairs to determine if overlapping exists. Then, the overlapping area of each pair of ROIs is calculated; of course, when no overlapping exists its value is zero.

If overlapping exists, the algorithm determines if the saliency overlap and saliency useful area criteria are fulfilled. If they are, the ROIs are merged into a unique ROI, and therefore the number of ROIs decreases. The algorithm restarts with the new set of ROIs. When the criteria are not fulfilled, either one, the algorithm proceeds to the next iteration to test a new pair of ROIs.

The algorithm stops when all possible combinations of ROIs have been tested, and no more ROIs pairs can be merged.

4.1.2 Face Attention Model

Given the importance that faces have for humans, this model identifies ROIs that contain faces. The model architecture, presented in Figure 4.5, contemplates four stages:

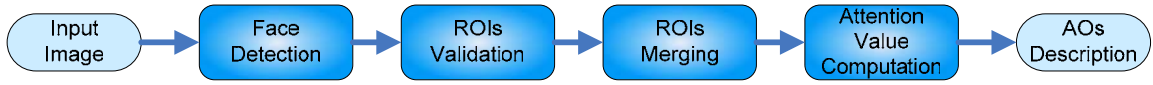


Figure 4.5: Architecture of the face attention model

The four stages of the model are now presented in detail. This model has a difference compared to the saliency attention model: the AV computation is the last stage of the model because it depends on the area of the ROI which can change due to possible merging.

1) Face Detection

This stage corresponds to the face detection algorithm presented in section 3.2.2, which determines the faces present in the image. It provides an output image map with the face ROIs of the input image, and a description file containing the location coordinates of each ROI.

2) ROIs Validation

Although the face detection stage usually doesn't provide face ROIs which aren't indeed faces, i.e. there aren't many false detection errors, a validation procedure has been implemented in order to account for this type of errors. The errors are usually related to skin regions that are not faces, such as hands or arms.

Hands or arms, and faces have different dimensions and shapes. Therefore, in order to eliminate the false detection errors, the criteria expressed in Equation (4.6) states that the face ROIs must have an area that is at least 40% of the area of the face ROI with the bigger area, to be validated; otherwise they are eliminated.

$$\frac{\text{area}(ROI_i)}{\max \text{area}(ROI_j)} \geq 0.40 \quad j = 1, \dots, N \quad (4.6)$$

After this stage, each face ROI is limited by a bounding box.

3) ROIs Merging

There are cases where face ROIs overlap each other. This can happen because faces are very close to each other, or simply because there was an error in the face detection stage, as shown in Figure 4.6. In these cases, it is necessary to decide if the face ROIs are merged or remain independent.

The experiments that were carried out allowed defining the following criteria to decide when face ROIs should be merged:

- Face overlap criterion: when the overlapping area is small this is mostly due to an imperfection in the definition of the ellipses that represent a face ROI; the faces are close and therefore a small overlap area exists. If there is a big overlapping area, this is typically the result of an error in the face detection stage, and it is typically better to merge the ROIs. Equation (4.7) shows the criterion to be checked; this means that if the overlap area of the face ROIs is equal to or bigger than 50% compared to the face ROI with the smallest area in the pair, they should be merged into a unique ROI.

$$\frac{area(ROI_i \cap ROI_j)}{\min\{area(ROI_i), area(ROI_j)\}} \geq 0.5 \quad (4.7)$$

- Face useful area criterion: it is important to avoid that a merged ROI is created with big areas being made of regions that are not ROIs. Equation (4.8) shows the criterion to be checked; this means that the area of the two face ROIs must represent at least 70% of the merged face ROI.

$$\frac{area(ROI_i) + area(ROI_j) - area(ROI_i \cap ROI_j)}{area(ROI_{merged})} \geq 0.7 \quad (4.8)$$

When the criteria are fulfilled, ROI_i and ROI_j are merged into a unique ROI. Figure 4.6 (a) shows an example of a case where face ROIs overlap, due to an error of the face detection stage. Since the defined criteria are fulfilled, the ROIs are merged into a unique face ROI, as shown in Figure 4.6 (b).

The algorithm developed to detect and solve the cases where overlapping exists between face ROIs is identical to the one presented in section 4.1.1 for saliency ROIs, but the face criteria are used instead of the saliency criteria.



Figure 4.6: Example of face ROIs overlapping and merging

4) Attention Value Computation

As it has already been mentioned at the beginning of this section, the AV of the AOs depends on the area of its ROI. Based on the work of Li-Qun Chen et. al [38], the AV is determined using Equation (4.9), which states that the importance of the AO is based on its area and position in the image.

$$AV_i = \sqrt{Area_{ROI_i}} * W^{pos} \quad (4.9)$$

W^{pos} represents the weight of the ROI's position in the image, according to the matrix presented in Figure 4.7. The weight matrix divides the image in three equal parts on the horizontal, and three different-sized vertical parts. The resulting 9 parts of the image have a weight which reflects the fact that human's privilege central zones of the image.

1/3	1/3	1/3	
1	2	1	3/12
4	8	4	4/12
1	2	1	5/12

Figure 4.7: Position weight matrix

4.1.3 Text Attention Model

Text presents a rich font of information, and therefore attracts viewer's attention. For this reason, the composite image attention model includes a text attention model, which comprehends four stages as pictured in Figure 4.8:

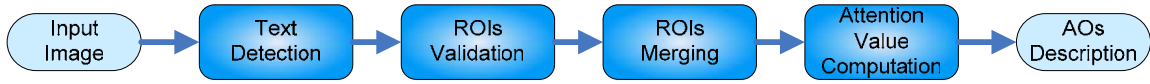


Figure 4.8: Architecture of text attention model

The four stages of the model are now presented in detail.

1) Text Detection

This stage corresponds to the text detection algorithm presented in section 3.2.3, which determines the text present in the image. It provides an output binary image with text regions, i.e. characters annotated in white color on a black background.

2) ROIs Validation

Given the text regions provided by the previous stage, it is necessary to process them to determine which belong together, i.e. which form words or phrases and therefore should be considered a unique ROI.

In order for characters to be grouped into a ROI, characters on the same line cannot be separated by more than 15 pixels, and characters on different lines cannot be separated by more than 15 pixels.

Although the text detection stage usually doesn't provide text ROIs which aren't indeed text, i.e. there aren't many false detection errors, a validation procedure has been implemented in order to account for this type of errors. If there are ROIs whose dimension compared to the biggest text ROI is small, this means less than 10%, it must be the result of an error at the text detection stage. Therefore, this stage filters ROIs according to the ratio

expressed by Equation (4.10); this means that a text ROI is validated if it verifies Equation (4.10).

$$\frac{area(ROI_i)}{\max area(ROI_j)} \geq 0.10 \quad j = 1, \dots, N \quad (4.10)$$

After this stage, each ROI is limited by a bounding box.

3) ROIs Merging

The previous stage determined ROIs which contain words or phrases. Sometimes, and despite being identified as separate ROI by the previous stage, the distance that separates the bounding box of two ROIs is small which can mean that they should be considered as a unique ROI.

The experiments that were carried out allowed defining the following criteria to decide when text ROIs should be merged:

- Text distance criterion: loose words have little semantic value. Equation (4.11) shows the criterion to be checked; this means that if the distance that separates two text ROIs bounding boxes, ROI_i e ROI_j , only in the horizontal direction, is equal to or smaller than 40 pixels they should be merged. Since text is read line by line, it doesn't make sense to merge text ROIs that aren't in the same horizontal space. In Figure 4.9 there are three examples of distances that could be calculated, between ROI_A and ROI_B , ROI_A and ROI_C , ROI_B and ROI_C . The only case for which the horizontal distance can be calculated is for the ROI_A - ROI_B pair, since they are in the same horizontal space, which is the space delimited by the two dashed blue lines. The distance is calculated as the minimum distance between the limits of the bounding boxes.

$$dist(ROI_i, ROI_j) \leq 40 \text{ pixels} \quad (4.11)$$

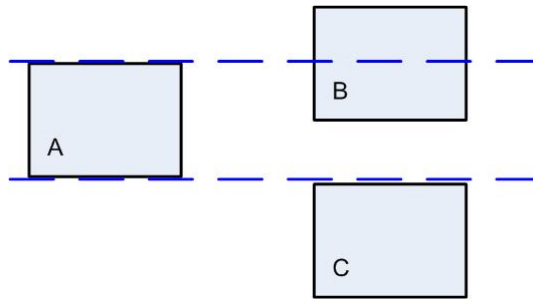


Figure 4.9: Horizontal distance calculation

- Text useful area criterion: it is important to avoid that a merged ROI is created with big areas being made of regions that are not ROIs. Equation (4.12) shows the criterion to be checked; this means that the area of the two text ROIs must represent at least 70% of the merged ROI.

$$\frac{area(ROI_i) + area(ROI_j)}{area(ROI_{merged})} \geq 0.7 \quad (4.12)$$

When the criteria are fulfilled, ROI_i and ROI_j are merged into a unique ROI. Figure 4.10 (a) shows an example of two ROIs which are independent, but since the criteria are fulfilled a merged ROI is formed as shown in Figure 4.10 (b).



Figure 4.10: Example of text ROIs merging

The structure of the algorithm developed to merge text ROIs is shown in Figure 4.11.

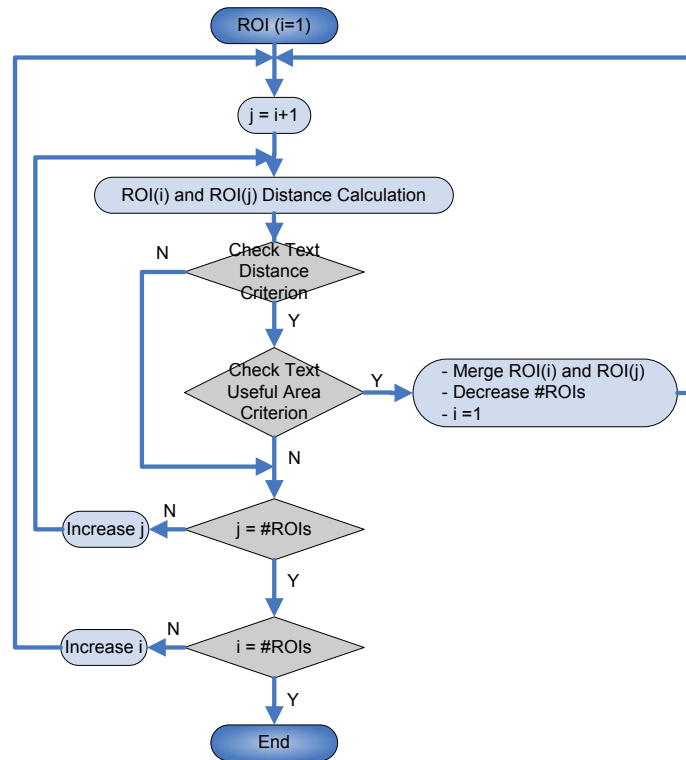


Figure 4.11: Structure of the text ROIs merging algorithm

The algorithm tests all the possible combinations of ROIs pairs to determine if the distance criterion is fulfilled. If it is, and the useful area criterion is also fulfilled, the ROIs are merged into a unique ROI, and therefore the number of ROIs decreases. The algorithm restarts with the new set of ROIs. When the criteria are not fulfilled, either one, the algorithm proceeds to the next iteration to test a new pair of ROIs.

The algorithm stops when all possible combinations of ROIs have been tested, and no more text ROIs pairs can be merged.

4) Attention Value Computation

For each AO, the corresponding AV is calculated in the same way as it is calculated for face ROIs (see section 4.1.2), using the following equation:

$$AV_i = \sqrt{\text{Area}_{ROI_i}} * W^{\text{pos}} \quad (4.13)$$

W^{pos} represents the weight of the ROI's position in the image, according to the matrix presented in Figure 4.7, which reflects the fact the human's privilege central zones of the image.

4.2 Attention Models Integration

The previous stage of the image attention model determines three types of AOs, which can spatially overlap: saliency, face and text AOs.

The attention models integration stage is responsible for integrating all the identified types of AOs into a unique image attention map using pre-defined criteria to solve the cases where spatial overlapping exists between them. To do so, three types of overlapping are considered: Face-Text, Face-Saliency and Text-Saliency.

The criteria used to solve the three overlapping cases, the calculation of the final AVs and the validation of AOs are discussed in the following sections.

4.2.1 Face-Text Integration

The process to solve the cases where the bounding boxes of text and face ROIs overlap states that they should always remain independent. Face and text AOs have completely different semantic values, and if overlapping exists it is due to imperfections in the definitions of their bounding boxes.

Therefore, it has been decided to trust the ROIs identification provided by the text and face detectors when this type of overlapping exists, i.e. when face and text ROIs overlap they remain independent.

4.2.2 Face-Saliency Integration

When face and saliency ROIs overlap, it is necessary to determine if they represent the same object or not. Based on the tests that were carried out, a criterion has been developed to solve the cases where the bounding boxes of saliency and face ROIs overlap. The criterion states that only when the face ROI contains a big part of the saliency ROI, they are likely to represent the same AO: a face. Therefore, as expressed in Equation (4.14), if the overlapping area is equal to or bigger than 25% compared to the area of the saliency ROI, the later is eliminated; otherwise, the two ROIs remain independent.

$$\frac{area(ROI_{face} \cap ROI_{saliency})}{area(ROI_{saliency})} \geq 0.25 \quad (4.14)$$

Figure 4.12 (a) presents an example where the bigger bounding box representing a face ROI is overlapped by a smaller saliency ROI. Since the criterion defined by Equation (4.14) is fulfilled, the saliency ROI is eliminated as shown in Figure 4.12 (b).

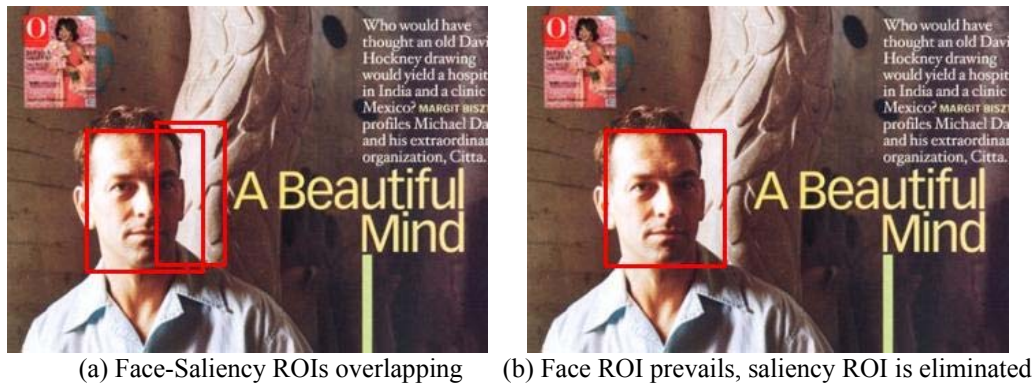


Figure 4.12: Example of Face-Saliency integration

Figure 4.13 shows the structure of the developed algorithm to solve the cases where face and saliency ROIs overlap.

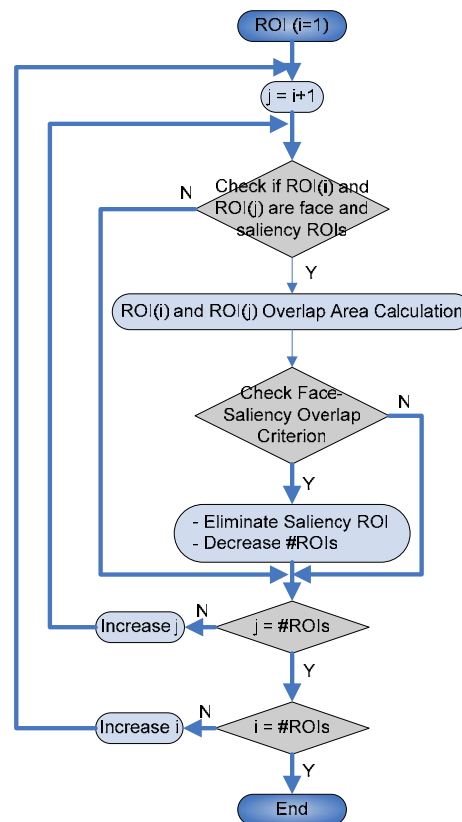


Figure 4.13: Structure of the Face-Saliency ROIs integration algorithm

The algorithm tests all the possible combinations of ROIs. For each pair of ROIs, the algorithm verifies if it is a Face-Saliency pair. When a Face-Saliency pair exists the overlap area is calculated. If the Face-Saliency overlap criterion is fulfilled, the saliency ROI is eliminated, and therefore the number of ROIs decreases. The algorithm then proceeds to the next iteration, i.e. evaluates the next ROIs pair.

The algorithm stops when all possible combinations of Face-Saliency ROIs pairs have been tested.

4.2.3 Text-Saliency Integration

When text and saliency ROIs overlap, it is necessary to determine if they represent the same object or not. Based on the tests that were carried out, a criterion has been developed to solve the cases where the bounding boxes of saliency and text ROIs overlap. The criterion states that only when the text ROI contains a big part of the saliency ROI, it is likely they represent the same ROI: text. Therefore, as expressed in Equation (4.15), if the overlap area is equal to or bigger than 25% compared to the area of the saliency ROI, the later is eliminated; otherwise, the two ROIs remain independent.

$$\frac{\text{area}(ROI_{\text{text}} \cap ROI_{\text{saliency}})}{\text{area}(ROI_{\text{saliency}})} \geq 0.25 \quad (4.15)$$

Figure 4.14 (a) presents an example where the bigger bounding box representing a text ROI is overlapped by a smaller saliency ROI. Since the criterion defined by Equation (4.15) is fulfilled, the saliency ROI is eliminated as shown in Figure 4.14 (b).

The structure of the algorithm developed to detect and solve the cases where overlapping exists between text and saliency ROIs is identical to the one presented in section 4.2.2.

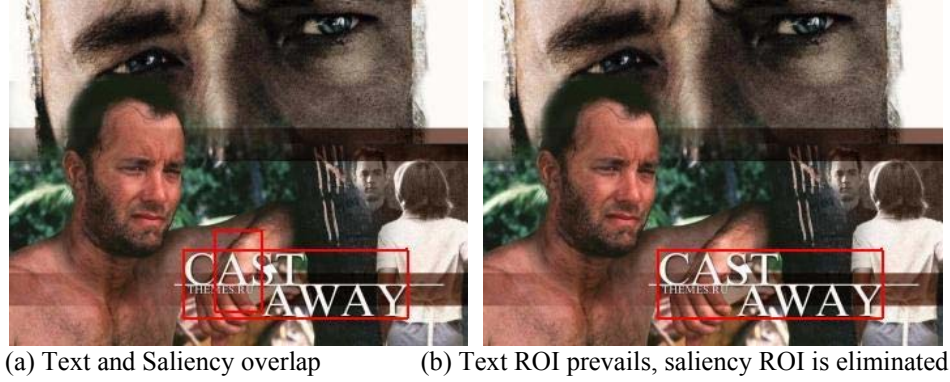


Figure 4.14: Example of Text-Saliency integration

4.2.4 Final Attention Value Computation

After integrating all the AOs by solving the overlapping cases, it is necessary to distinguish the importance that each AO has according to its type: saliency, face or text.

The AVs used until now are normalized on a scale ranging from 0 to 100. In this work, faces are considered more important, followed by text and saliency. This is done because faces are considered to be very important for humans; this means faces are the kind of object that a human will look for first in an image. Text can provide a lot of information, delivering or enhancing the message that an image is intended to transmit; therefore it is considered the second most important type of AOs. Saliency AOs are considered the least important because nothing is known regarding their semantic value, they can be any kind of object.

Therefore to calculate the final AV of each AO, Equation (4.16) is used. The final AV is obtained by multiplying the current AV by the weight corresponding to the type of AO: saliency, face or text.

$$AV_{\text{final}} = AV \times W_m, \text{ with } m \in \{\text{saliency}, \text{face}, \text{text}\} \quad (4.16)$$

Several experiments were realized, which lead to the definition of the following weight values: $W_{\text{saliency}} = 0.2$, $W_{\text{text}} = 0.35$ and $W_{\text{face}} = 0.45$.

4.2.5 Final AOs Validation

AOs that have a relative small AV are considered to provide little information and therefore are eliminated. The AOs that verify the condition in Equation (4.17), i.e. which have a relative AV equal to or bigger than 10% regarding the maximum AV in the image are validated and considered for the remaining stages of the algorithm. The AOs that don't fulfill Equation (4.17) are eliminated.

$$\frac{AV(AO_i)}{\max AV(AO_j)} \geq 0.10 \quad j = 1, \dots, N \quad (4.17)$$

4.3 Optimal Path Generation

After the execution of the previous stage, there is a set of AOs, i.e. there is a description of all the objects present in the image that are considered to considerably attract viewers attention.

Given the objective of this work, i.e. display an image with video, it is necessary to determine the optimal path to transform the image into video showing in detail the AOs of the image. To do so, this stage optimizes the information presented on the screen using two mechanisms that are presented in the following sub-sections.

4.3.1 Display Size Adaptation

The video sequence is created so that AOs are displayed with their maximum quality, i.e. the AOs are displayed with their original spatial resolution. The objective of this mechanism is to optimize the information presented on the screen at each moment.

To do so, the developed algorithm splits or groups AOs to form attention groups (AGs), which have a dimension equal to or smaller than the display size. As the AOs, the AGs have two attributes: the ROI_i , which is the region of the image that contains the AG_i ; and the attention value (AV_i), which represents an estimate of the user's attention on the AG. An AG can contain part of an AO, one or more AOs. When an AO is split into two or more parts, that can fit into the display size, they constitute a new AG called twin. When one or more AOs are grouped, they constitute a new AG. If an AO is neither split nor grouped, it also constitutes an AG.

The display size adaptation process is performed using two methods: first Split Processing and afterwards Group Processing, which are now explained.

1) Split Processing

It is usual that the spatial resolution of the AO is bigger than the spatial resolution of the image display in which case it is necessary to spatially divide the AO in smaller parts that fit the display size. The generated AG twins inherit the AV of the original AO.

Face AOs are never divided since they must be visualized as a whole to have semantic meaning. The adopted method to display face AOs whose spatial resolution is bigger than the spatial resolution of the image display is provided in section 4.4.5. Text AOs can be divided since the resulting AGs will be displayed in a sequential manner that will allow the user to read the text.

Figure 4.15 shows how the developed algorithm checks three conditions to determine which AOs need to be split, and how they are split to fit the display size.

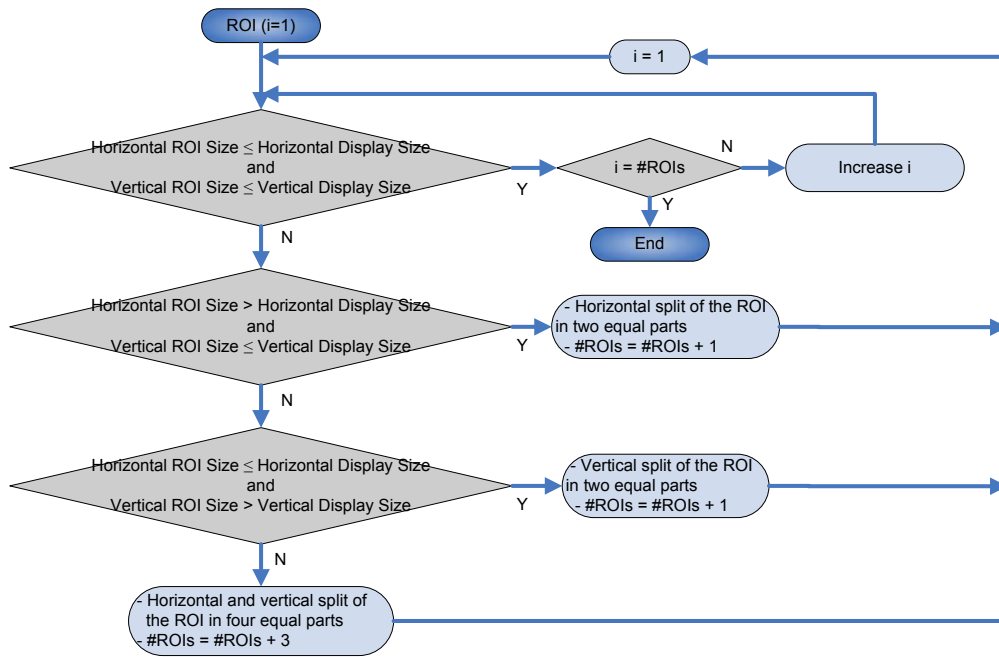


Figure 4.15: Structure of Split Processing algorithm

The algorithm starts by checking if the ROI's horizontal and vertical dimensions are smaller than those of the display size. If they are, the ROI fits the display size, and therefore the algorithm then checks if the actual ROI is the last one of the ROIs set to determine if it should end or proceed to the next ROI.

If the horizontal and vertical dimensions of the ROI are bigger and smaller, respectively, than those of the display size, the algorithm horizontally splits the ROI in two equal parts, and therefore increases the number of ROIs.

If the horizontal and vertical dimensions of the ROI are smaller and bigger, respectively, than those of the display size, the algorithm vertically splits the ROI in two equal parts, and therefore increases the number of ROIs.

If the previous three conditions are not verified, this means that the horizontal and vertical dimensions of the ROI are both bigger than those of the display size, and therefore the algorithm horizontally splits the ROI in two equal parts; each of the new parts is then vertically split in two equal parts. The number of ROIs therefore increases by three.

The algorithm stops when all ROIs fulfill the first condition this means neither the vertical nor the horizontal sizes of the RI are bigger than the corresponding size of the display.

Figure 4.16 presents an example of a ROI whose horizontal dimension exceeds the horizontal size of the display (green rectangle), and therefore is divided into 4 AG twins that fit into the display size.

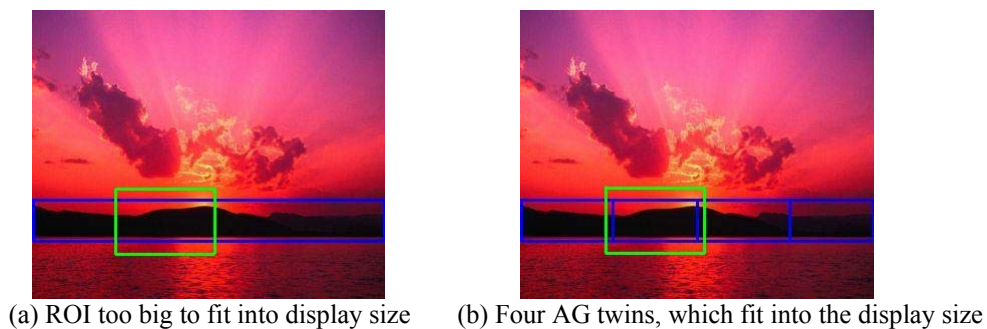


Figure 4.16: Example of results generated by AG split processing

2) Group Processing

Since some AGs are very small compared to the display size, so they can be grouped with others, when possible, forming an AG which provides maximum information to the user on the display. When AGs are grouped, the ROI of the new AG is represented by the smallest bounding box that can contain both AGs, and its AV inherits the highest AV of the grouped AGs.

Figure 4.17 shows how the developed algorithm proceeds to determine which AGs can be grouped.

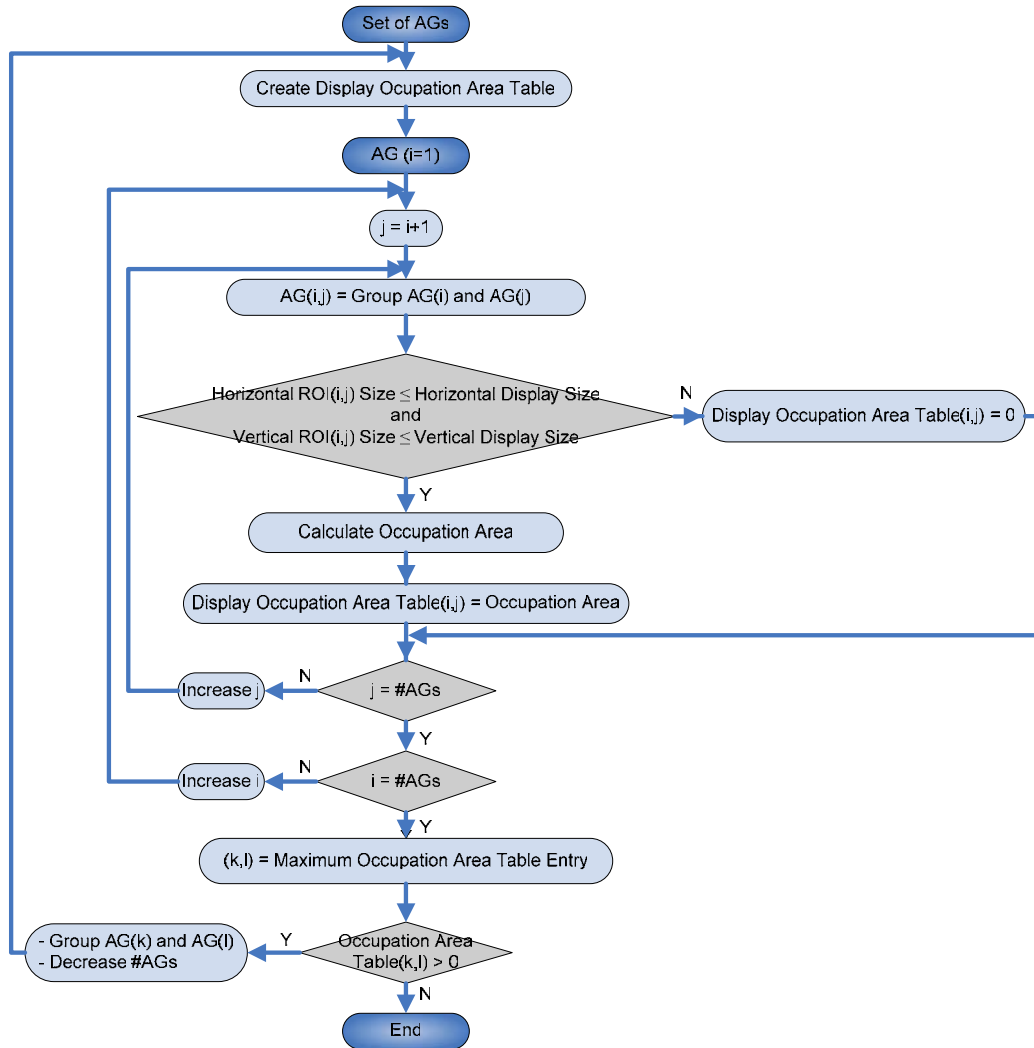


Figure 4.17: Structure of Group Processing algorithm

The Group Processing algorithm starts by creating a table, called Display Occupation Area, with the number of lines and columns equal to the number of AGs.

The algorithm tests all the possible group combinations of AGs. For each pair of AGs, AG_i and AG_j , the algorithm determines if the horizontal and vertical dimensions of the grouped ROI are smaller than those of the display. When the grouped ROI fits into the display size, the percentage of area of the display that is occupied by the ROI is calculated and attributed to the Display Occupation Area table entry (i,j) . Otherwise, entry (i,j) is zero, which means the grouped ROI for that pair does not fit the display size.

After the algorithm finishes testing all the possible pair combinations of AGs, the AG pair that grouped occupies the maximum area of the display is determined; this means that the algorithm searches for the maximum entry (k,l) of the table.

If the maximum entry (k,l) is zero, this means no more AG pairs that fit the display size when grouped exist, and therefore the algorithm ends. Otherwise, the AG pair, AG_k and AG_l is grouped, the number of AGs is decreased and the algorithm proceeds to the next iteration.

When it is possible to group AGs, like in the example presented in Figure 4.18, the new AG inherits the highest AV of the two AGs.



Figure 4.18: Example of results generated by Group Processing

4.3.2 Browsing Path Generation

This mechanism determines the order by which AGs will be displayed, and therefore establishes the path that will be used to display with video the whole image.

AGs are shown in detail, following the order of their AV, i.e. the AG with the highest AV is the first to be displayed. However, in some cases the displaying order can be changed. Changing the order by which AGs are displayed can save displaying time, and also avoid traveling back and forward in the image, which can be unpleasant for the user.

Several experiments were performed to define the criteria to decide in which cases the displaying order of the AGs should be changed. Figure 4.19 presents an example of the spatial distribution of three AGs with $AV(AG_i) > AV(AG_j) > AV(AG_k)$, which is used to explain the developed criteria:

- AGs distance criterion: this criterion states that if the traveled distance for the normal order of the AGs, $AG_i \rightarrow AG_j$, and the distance traveled by attending another AG_k is similar, the displaying order of the AGs could be changed. The criterion is checked if Equation (4.18) is fulfilled, i.e. if the distance ratio is equal to or bigger than 95%.

$$\frac{\text{dist}(AG_i, AG_j)}{\text{dist}(AG_i, AG_k) + \text{dist}(AG_k, AG_j)} \geq 0.95 \quad (4.18)$$

- AGs AV criterion: this criterion states that if the AVs of the second and third most important AGs, AG_k and AG_j , are similar then the displaying order could be changed. This criterion is fulfilled if Equation (4.19) is verified, i.e. if the AVs ratio is equal to or bigger than 90%.

$$\frac{AV(AG_k)}{AV(AG_j)} \geq 0.90 \quad (4.19)$$

When both the criteria are fulfilled, the order by which AG_i , AG_j and AG_k are displayed becomes $AG_i \rightarrow AG_k \rightarrow AG_j$.

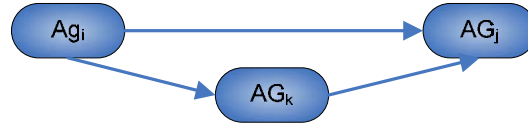


Figure 4.19: Example of spatial distribution of AGs

Figure 4.20 shows the algorithm developed to determine the optimal browsing path for the image to video conversion.

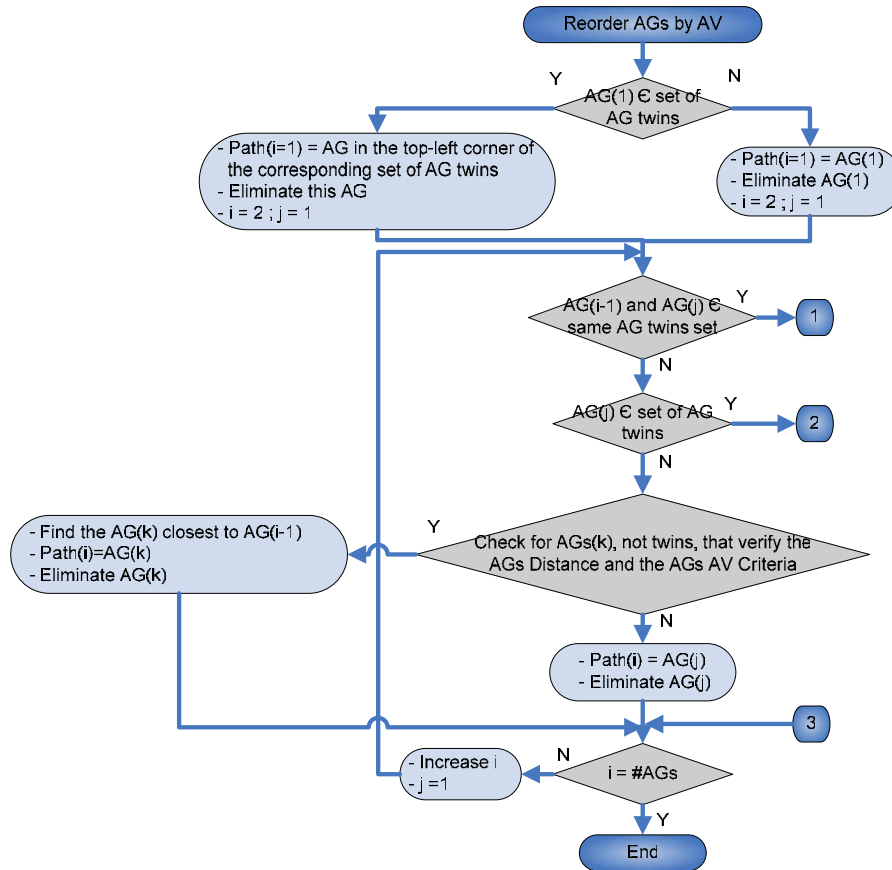


Figure 4.20: Structure of the Optimal Path Generation algorithm (Main part)

The algorithm has three parts, which are now explained:

- **Main part:** the algorithm starts by ordering the AGs according to their AV, forming the ordered AG set (OAGS). The objective of the algorithm is to transform the OAGS into the browsing path set (BPS), according to pre-defined conditions, to obtain the optimal order to display the AGs. The first element of BPS, $AG(i=1)$, is the AG with highest AV, i.e. the first element of the OAGS. However, if it belongs to a set of AG twins, the AG twin situated in the top-left corner becomes the first element of the BPS. As AGs are added to the BPS, they are removed from the OAGS.

The algorithm then checks if the last element of the BPS, $AG(i-1)$, and the next element of the OAGS, $AG(j)$, are both AG twins that belong to the same AG twins set. If they belong to the same AG twins set, the algorithm proceeds to node 1 as shown in Figure 4.21. Otherwise, the algorithm checks if the next element of the OAGS, $AG(j)$, belongs to a set of AG twins; if it belongs to a set of AG twins the algorithm proceeds to

node 2 as shown in Figure 4.22. Otherwise, the algorithm checks if other elements of the OAGS, $AGs(k)$, which aren't AG twins and which verify the AGs distance and AV criteria exist. If such elements of the OAGS exist, the one closest to the last element of the BPS becomes the next element of the BPS, $AG(i)$. Otherwise, $AG(j)$ becomes the next element of the BPS.

The algorithm stops when all the elements of the OAGS have been integrated into the BPS.

- **Node 1:** when the algorithm proceeds to node 1, as shown in Figure 4.21, it determines the $AG(j)$ twin closest to $AG(i-1)$. This procedure privileges spatial horizontal movements as opposed to vertical movements, because the first are more pleasant for the human eye.

When visualizing a set of text AG twins, they should always be visualized from left to right; therefore the algorithm determines if the last element of the BPS belongs to a set of text AG twins, and if it is the right-most AG of the AG twins set. If these conditions are fulfilled, the left-most AG closest to $AG(i-1)$ becomes the next element of the BPS. Otherwise, $AG(j)$ becomes the next element of the BPS.

- **Node 2:** when the algorithm proceeds to node 2, as shown in Figure 4.22, the algorithm checks if other elements of the OAGS, $AGs(k)$, which aren't AG twins and which fulfill the AGs distance and AV criteria exist. If such elements of the OAGS exist, the one closest to the last element of the BPS becomes the next element of the BPS, $AG(i)$. Otherwise, the algorithm determines the $AG(j)$ twin closest to $AG(i-1)$, which is situated in one of the corners of the AG twin set.

When visualizing a set of text AG twins, the first AG twin to be presented is the one located in the top-left corner of the AG twin set; therefore the algorithm checks if the previously determined $AG(j)$ belongs to a set of text AG twins. If $AG(j)$ belongs to a set of text AG twins, the top-left corner of the AG twin set becomes the next element of the BPS. Otherwise, $AG(j)$ becomes the next element of the BPS.

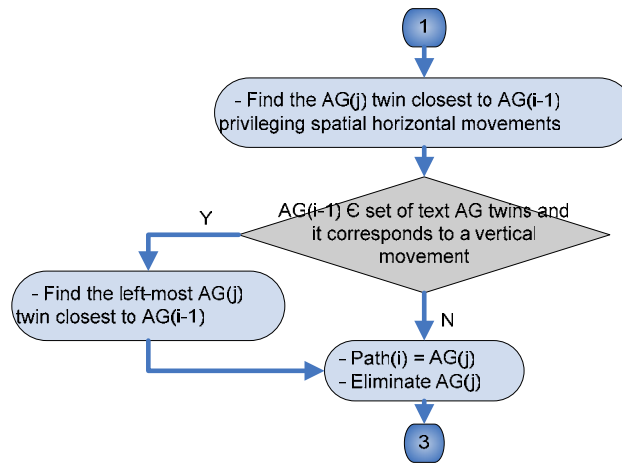


Figure 4.21: Structure of the Optimal Path Generation algorithm (Node 1)

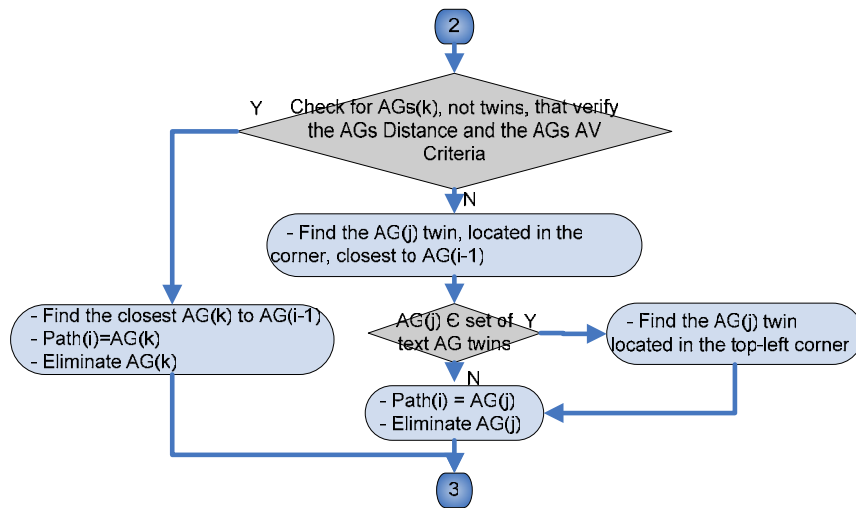


Figure 4.22: Structure of the Optimal Path Generation algorithm (Node 2)

The final result of the Browsing Path Generation mechanism is the BPS, which provides the order by which AGs should be displayed in the video sequence.

Figure 4.23 shows an example of the optimal path to attend the AGs present in the image. The numbers inside the bounding boxes represent the order by which AGs are displayed.

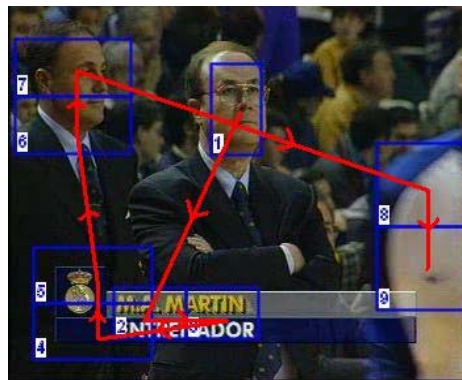


Figure 4.23: Example of browsing path

4.4 Video Creation

This stage of the Image2Video system is responsible for generating the frames that compose the video sequence, which displays the AGs of the image according to the optimal path calculated in the previous stage and taking into consideration the user preferences.

In the next sub-sections, the frame types and motion units that are used to produce the adapted video are defined, as well as the critical times associated to motion units and perception of the AGs. A description of the algorithm that extracts the frames that compose the video is provided, and, finally, the video modes available for the user to choose are presented.

4.4.1 Key Frame Types

It is important to generate video sequences with smooth motion, so that the video experience is pleasant for the user, in this case also more pleasant than the simple image experience. In order to

achieve this target, it is necessary to define the key-frames involved in the motion sequences to be produced.

Typically, in film making, there are three types of video shots: full, medium and close-up. All three have the same spatial resolution, but represent contents of the image with different dimensions, i.e. an object which is small in a full shot can be viewed in detail in a close-up shot where it has a dimension close to the display's size. Based on the three types of video shots, and in order to generate the video sequence, three types of key-frames have been defined:

- **Full-frame (FF):** these frames present the entire content of the image. When the aspect ratio of the image and the display are different, the aspect ratio of the image is preserved.
- **Medium-frame (MF):** these frames are used to contextualize the AGs within the image; therefore a window containing 25% of the image, with the aspect ratio of the display, is used. The tests carried out indicated that this window dimension is adequate to perceive the context of the AG. Their spatial dimensions cannot be smaller to those of the display. The MF of an AG is centered on it, therefore contextualizing the AG in the image.
- **Close-up frame (CF):** these frames have the dimension of the display size and are used to display the AGs in detail.

Figure 4.24 presents an example of the content presented by the three types of key-frames: FF (pink), MF (blue) and CF (red).



Figure 4.24: Examples of the three key-frame types

4.4.2 Motion Units

In order to generate the video, it is necessary to create a sequence of key-frames that displays the AGs of the image according to the calculated optimal path. Considering the three types of key-frames presented in section 4.4.1, six types of motion units are possible, which provide different ways of traveling from one frame to the other.

Table 4.1 presents the six types of motion units, and the type of key-frames involved in each transition.

Table 4.1: Types of video motion units

Motion Unit Type	Transition type
Pan (PA)	MF→MF
Local Pan (LP)	CF→CF
Local Zoom In (LZI)	MF→CF
Local Zoom Out (LZO)	CF→MF
Full Zoom In (FZI)	FF→MF
Full Zoom Out (FZO)	MF→FF

The six types of video motion units can be grouped into 4 categories, which are now presented.

1) Full Zoom

This kind of motion evolution is used at the beginning and at end of the video, with a FZI and FZO, respectively. The first frame of the video is a FF and a FZI is executed until the MF of the first AG is obtained. In a similar manner, at the end of the video, a FZO is executed to go from the MF corresponding to the last AG to the final FF.

The motion speed for this motion unit, called Zoom Rate (ZR), can be calculated using Equation (4.20) based on the difference between the spatial horizontal resolution (SHR) in pixels of the FF and MF and the time interval available to perform the zoom:

$$ZR = \frac{SHR(FF) - SHR(MF)}{\Delta t} \quad (4.20)$$

The ZR provides a measure of the pixel distance covered per second when this type of motion is used. The time interval available to perform the full zoom is presented in section 4.4.3.

2) Local Zoom

This kind of motion evolution is used when a zoom is performed from a MF to a CF (LZI) and from a CF to a MF (LZO).

The motion speed, in this case called local zoom rate, can be calculated in the following manner:

$$LZR = \frac{SHR(MF_i) - SHR(CF_i)}{\Delta t} \quad (4.21)$$

The time interval, Δt , available to perform the LZO or LZI is presented in section 4.4.3.

3) Panning Motion

This kind of motion evolution is used when it is necessary to go from one MF to another MF. In this case, the motion speed called pan velocity can be calculated using Equation (4.22) based on the pixel distance between the center coordinates of the MFs:

$$PV = \frac{center(MF_i) - center(MF_{i+1})}{\Delta t} \quad (4.22)$$

The time interval, Δt , available to perform PA is presented in section 4.4.3.

4) Local Panning Motion

This kind of motion evolution is used when it is necessary to go from one CF to another CF, i.e. when displaying AGs in detail. Therefore, the motion speed, called local pan velocity (LPV), should be small to allow visualizing the AGs correctly.

The LPV can be calculated in the following manner:

$$LPV = \frac{center(CF_i) - center(CF_{i+1})}{\Delta t} \quad (4.23)$$

The time interval, Δt , available to perform the LP presented in section 4.4.3.

4.4.3 Motion Durations and Perception Times

The video sequence must be as smooth as possible in the sense that transitions from key-frame to key-frame should take the time necessary for the user to perceive all the information available. Therefore the time intervals between key-frames, as well as the minimal perceptible times (MPT) to perceive the semantic value of AGs, must be defined carefully.

The experiments carried out allowed establishing the default MPT for the different types of AGs and the duration times for the different motion unit types, which are presented in Table 4.2 and Table 4.3, respectively. These values were defined based on the perception and experience of the authors of this project, i.e. they were considered to be adequate in producing a pleasant and informative video sequence.

Table 4.2: Default minimal perceptible times

Attention Group Type	Minimal Perceptible Time (seconds)
Face	0.5
Text	0.5
Saliency	0.5

Table 4.3: Default motion units duration times

Motion Unit Type	Time (seconds)
PA	1.5
Text LP	1.0
Saliency LP	1.0
LZI / LZO	1.0
FZI / FZO	1.5

4.4.4 Video Display Modes

Different users have different needs and different preferences. Therefore, the user can choose one of three different video display modes for the visualization of the adapted video sequence:

- 1) **Normal:** all the AGs are presented, without any restriction.
- 2) **Time Based (TB):** the user chooses to see an adapted video sequence with a maximum time limit; therefore the video sequence will only show the most important AGs within the time limit.
- 3) **Amount of Information Based (AIB):** the user chooses to see an adapted video sequence with a determined percentage of information, i.e. the video sequence will only show the most important AGs corresponding to the chosen percentage of information.

4.4.5 Video Directing

Based on the key-frames and timings that have been defined in the previous sub-sections, it is now possible to present how key-frames are sequenced to produce a video that displays the AGs defined for an image. The developed algorithm that produces the video clip also takes into consideration the video mode chosen by the user; the video modes were presented in section 4.4.4. Figure 4.25 shows how the video generation algorithm produces a sequence of key-frames to produce the video clip,

which displays the AGs of the BPS. The algorithm uses two counters, one to accumulate the elapsed time of the video, and another to accumulate the percentage of information displayed.

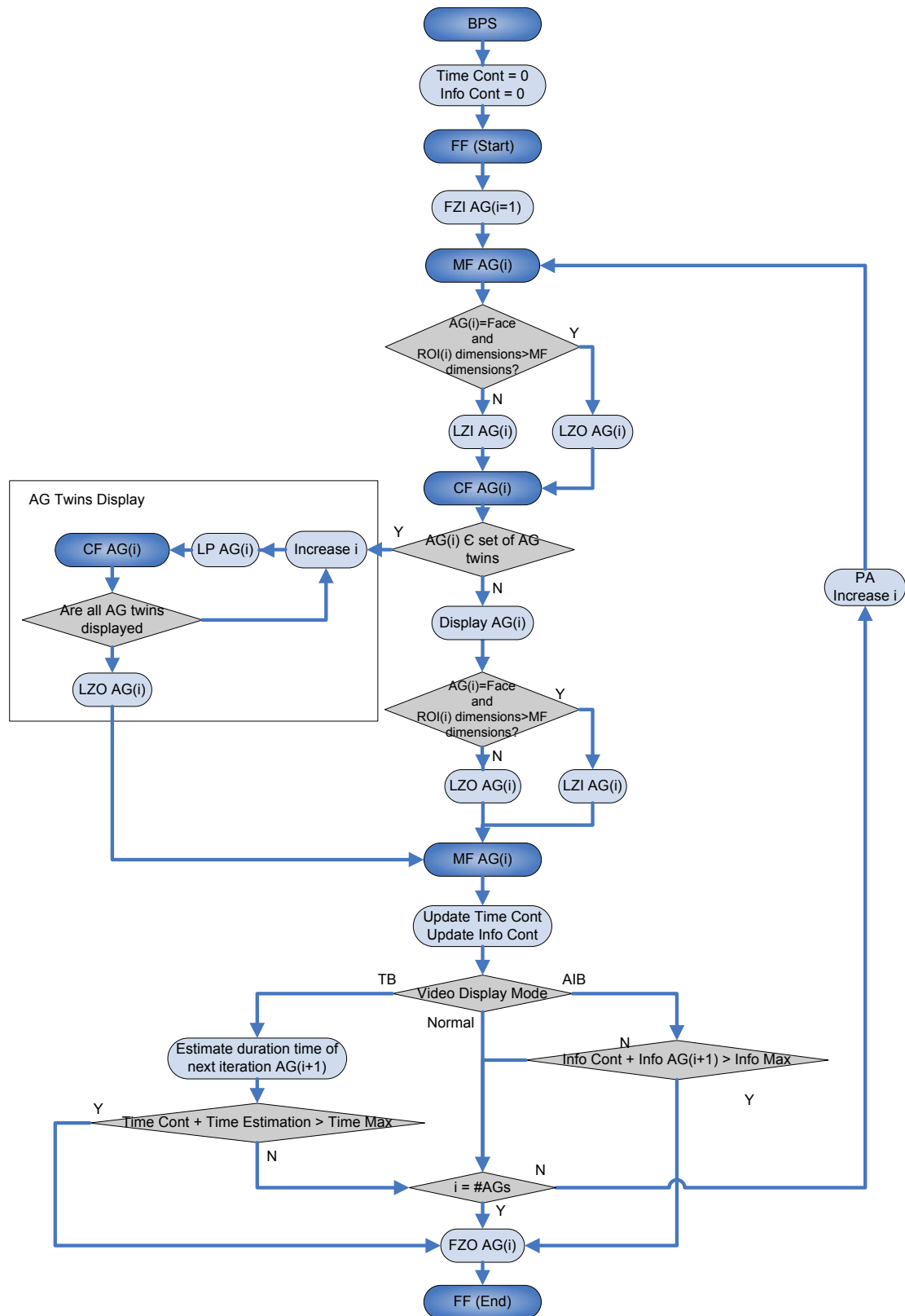


Figure 4.25: Video clip generation algorithm

The first motion unit type to be performed is a FZI to go from the FF that presents the entire image content, to the MF corresponding to the first element of the BPS. Since the MF of an AG is centered on it, this allows contextualizing the AG in the image.

If the AG is a face with bigger dimensions than the corresponding MF, then the corresponding CF will also be bigger than the MF; in these cases, a LZO is performed. Otherwise, a LZI is performed to obtain the CF corresponding to the AG.

The algorithm then checks if the AG belongs to a set of AG twins:

- If it does, the algorithm proceeds to the AG twins display block, which successively shows the AG twins in the set, according to the order established in the BPS. This is done by performing LP motions to display all the AG twins of the set. Afterwards, a LZO is performed to go to the MF of the last AG twin displayed.
- Otherwise, if the AG doesn't belong to a set of AG twins, it is displayed during the defined MPT. The algorithm then checks if the AG is a face with bigger dimensions than the corresponding MF; if it is a LZI is performed. Otherwise, a LZO is performed to obtain the MF corresponding to the AG.

After performing the actions already described, the time and information counters are updated. The algorithm then checks which of the three video display modes the user chose:

- **Normal Video Mode:** If the user selected the normal video mode, the algorithm checks if the current AG is the last one of the BPS; if it is, a FZO is performed to go from the MF of the last AG to the FF, ending the video with the entire image content displayed. Otherwise, a PA motion is performed to go to the MF of the next AG to be displayed.
- **TB Video Mode:** If the user selected the TB video mode, the algorithm estimates the elapsed time to display the next $AG(i+1)$. If the sum of the time counter and the estimated time is bigger than the maximum time defined by the user, a FZO is performed to go from the MF of the last AG to the FF, ending the video with the entire image content displayed. Otherwise, a PA motion is performed to go to the MF of the next AG to be displayed.
- **AIB Video Mode:** If the user selected the AIB video mode, the algorithm determines if the sum of the information counter and the percentage of information provided by the next AG to be displayed, $AG(i+1)$, is bigger than the maximum information percentage limit defined by the user. If it is, a FZO is performed to go from the MF of the last AG to the FF, ending the video with the entire image content displayed. Otherwise, a PA motion is performed to go to the MF of the next AG to be displayed.

4.5 Final Remarks

This chapter presented the processing algorithms developed by the authors of this work to reach the objectives defined for the Image2Video application. The four main stages that compose the architecture of the system were presented in detail, each one with example results to better demonstrate their purpose. This should allow perceiving the evolution of the input image through the adaptation system until a video sequence is created.

Since it is impossible to present videos in this report, a part of Chapter 6 is dedicated to the presentation of the results obtained from the subjective tests that were performed to assess the quality of the videos created by this application.

In the following chapter, the application interface that has been developed to integrate all the processing algorithms, background and developed by the authors, will be presented in detail.

Chapter 5

Image2Video Application Interface

This chapter is dedicated to the presentation of the developed application interface, which integrates all the algorithms described in the previous chapters. The interface allows adjusting parameters and visualizing all the results from the used algorithms.

In order to develop the application interface, the authors of this project learned to program with the object oriented programming language C# language from Microsoft. The application requires the installation of both the Microsoft .NET Framework 1.1 and Image Magick version 6.0.0 C++ toolbox. Additionally the Regional Options in Windows must be set to Portuguese (Portugal).

The following sections provide a guide on how to use the interface, i.e. the actions that have to be performed to produce the algorithm results presented in the previous chapters. Instructions on how to execute the application can be found in Appendix A.

5.1 Interface Layout

When the application interface is run, the main window is presented on the screen with the layout pictured in Figure 5.1. In the main window, there are three types of image boxes:

- **Input image:** as the name suggests this image box situated in the top-left corner of the main window displays the input image chosen by the user.
- **Intermediate results:** the three image boxes situated in the lower part of the main window are used to display the intermediate results of the algorithms.
- **Final result:** the image box situated in the top-right corner displays the final result of the stages that compose the architecture of the adaptation system.

The main window also contains four action buttons box: three buttons to execute the algorithms regarding the stages that compose the architecture of the adaptation system, and one button to open the video player. The Select Intermediate Results box allows the user to choose what is displayed in the intermediate results image boxes. An explanation on how to use the action buttons and the Select Intermediate Results box is provided in section 5.2. The menu's options are described in section 5.1.1.

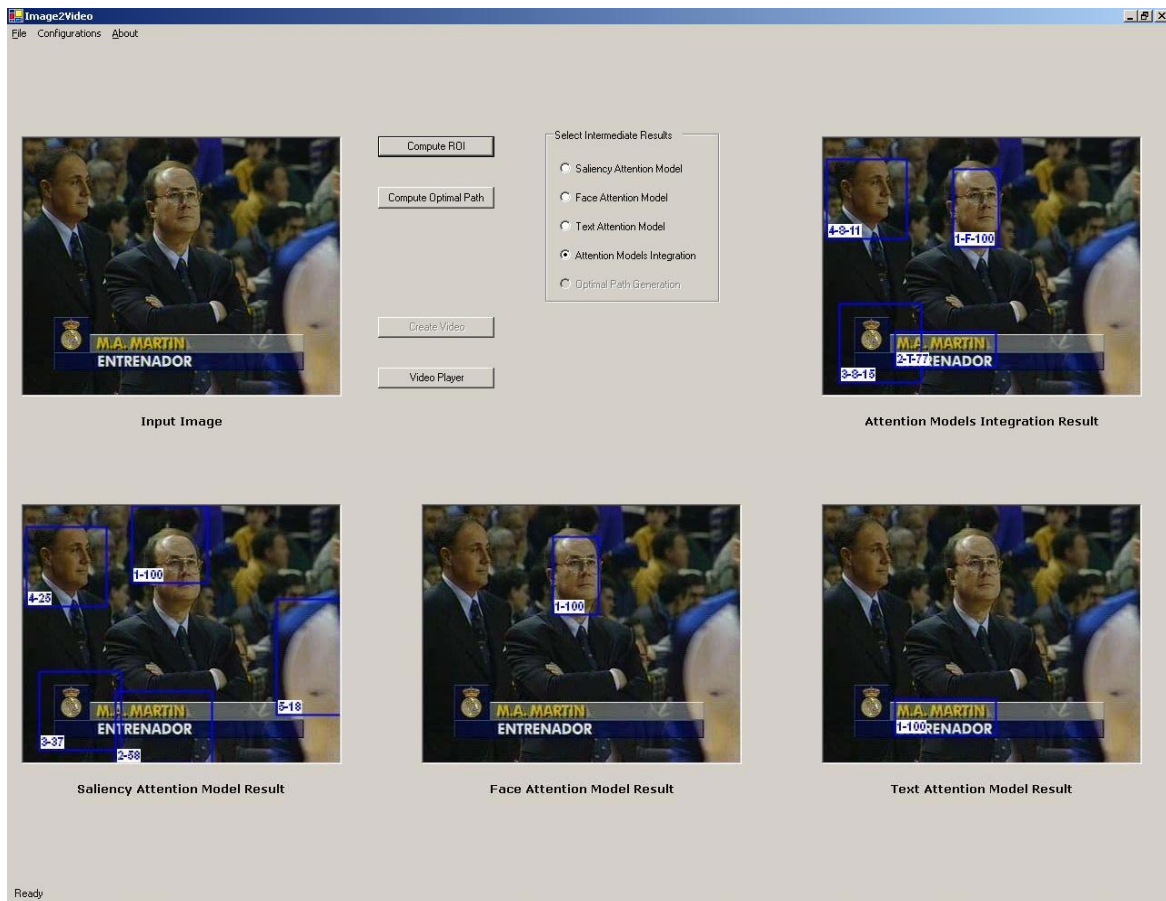


Figure 5.1: Main window of the developed Image2Video application interface

5.1.1 Application Menu

The interface, as shown in Figure 5.2, has three menus. The three menus are now presented.



Figure 5.2: The application menus

1) File

This menu has two options:

- Open: this option provides a dialog box to choose the image file for which a video sequence is to be produced. It starts in the default image database directory, but any other directory can be chosen. The supported file types are jpeg, bitmap and gif.
- Exit: this option allows exiting the application.

2) Configurations

The algorithms that are integrated into the application have several parameters that can be adjusted. Therefore the interface provides a way to adjust and save all the parameters into a unique configuration file. This menu's options are:

- Load: this option allows adjusting the algorithm's parameters according to a configuration file previously saved by the user.
- Save: this option allows the user to save the current parameters for all the algorithms into a configuration file.
- Customize: this option allows the user to adjust the parameters of all the algorithms. More details on this option are presented in section 5.1.2.

3) About

This menu simply provides information regarding the authors of the developed application and the place where it was developed, as shown in Figure 5.3.



Figure 5.3: Image2Video About window

5.1.2 Customize Menu

When this option is chosen from the Configurations menu, the window shown in Figure 5.4 is presented in the computer screen.

As can be seen in Figure 5.4, the window presents all the parameters that can be adjusted regarding the saliency attention model. This is the first of six tabs in this window:

- 1) **Saliency Model**: this tab allows adjusting the parameters of the several stages of the saliency attention model, described in sections 3.2.1 and 4.1.1.
- 2) **Face Model**: this tab allows adjusting the parameters of the several stages of the face attention model, described in sections 3.2.2 and 4.1.2.
- 3) **Text Model**: this tab allows adjusting the parameters of the several stages of the text attention model, described in sections 3.2.3 and 4.1.3.
- 4) **Models Integration**: this tab allows adjusting the parameters of the attention models integration stage, described in section 4.2.
- 5) **Optimal Path**: this tab allows adjusting the parameters of the optimal path generation stage, described in section 4.3; one of the most important options is the display size for which the video sequence is intended to be adapted, as shown in Figure 5.5 (a).
- 6) **Video Creation**: this tab allows adjusting the parameters of the video generation stage, described in section 4.4; one of the most important options is the user's preferred video visualization mode, as shown in Figure 5.5 (b).

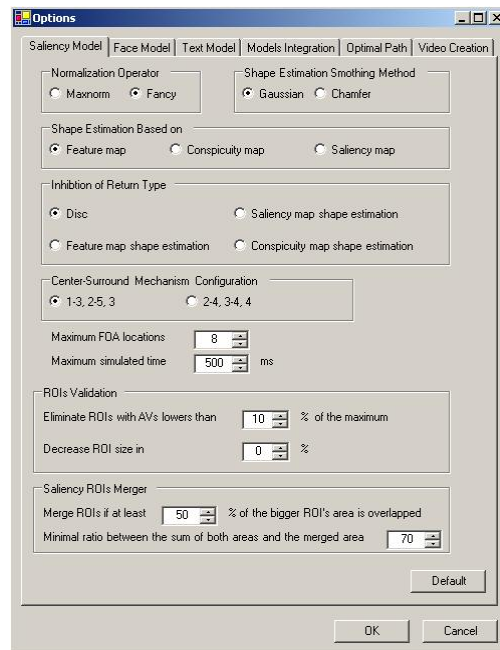
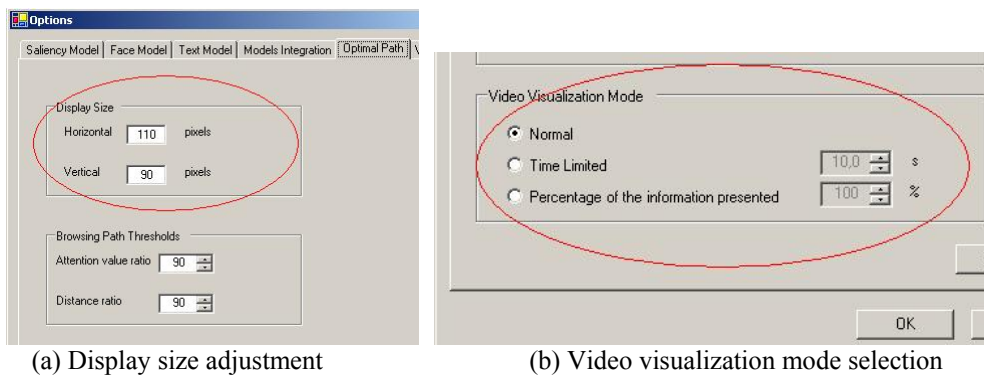


Figure 5.4: Customize menu window



(a) Display size adjustment

(b) Video visualization mode selection

Figure 5.5: Examples of option windows: Optimal Path and Video Creation

5.2 Running the Application

The main window of the Image2Video application interface has four action buttons, highlighted by the red ellipse in Figure 5.6. It also has five selection buttons, grouped in the Select Intermediate Results box, which allow choosing what is displayed in the intermediate results image boxes.

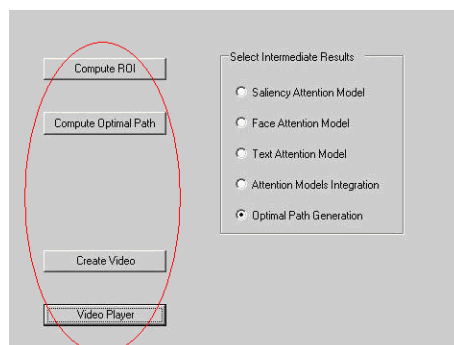


Figure 5.6: Action buttons and intermediate results selection

When the application starts, the video player and ROI computation action buttons are immediately enabled. After pressing the Compute ROI button and the corresponding algorithms are finished, the Compute Optimal Path button is enabled. If this button is pressed, the Create Video button will be enabled after the corresponding algorithms are finished. This allows the user to intuitively follow the correct order of execution of the algorithms associated to the various architectural stages that compose the Image 2Video adaptation system architecture.

A description of the functions related to each action button, and of each result selection button is now presented.

1) Compute ROI

This action button executes the algorithms associated to both the Composite Image Attention Model and Attention Models Integration stages, described in sections 4.1 and 4.2, respectively.

When the processing corresponding to these two algorithms ends, the user can choose which results are displayed on the intermediate results image boxes by selecting one of the buttons in the Select Intermediate Results box. By default, when the processing ends, the Attention Models Integration results button is automatically selected, as shown in Figure 5.1. A description of the results presented in the intermediate results image boxes according to the selected button is now provided:

- Saliency Attention Model: the results provided by the saliency detection, ROIs validation and ROIs merging stages, as well as the AV of each AO are shown.
- Face Attention Model: the results provided by the face detection, ROIs validation and ROIs merging stages, as well as the AV of each AO are shown.
- Text Attention Model: the results provided by the text detection, ROIs validation and ROIs merging stages, as well as the AV of each AO are shown.
- Attention Models Integration: the final results of the integration of the saliency, face and text attention models are shown (see Figure 5.1).

The results presented in the image boxes have blue bounding boxes around the AOs, each one with a label that provides additional information. As shown in Figure 5.7, there are two types of labels:

- O-AV: this label presents the order (O) of importance and the respective AV of the AO; see Figure 5.7 (a) for example.
- O-M-AV: this label additionally presents the model (M) that identified the AO: S for saliency, F for face or T for text; see Figure 5.7 (b) for example.

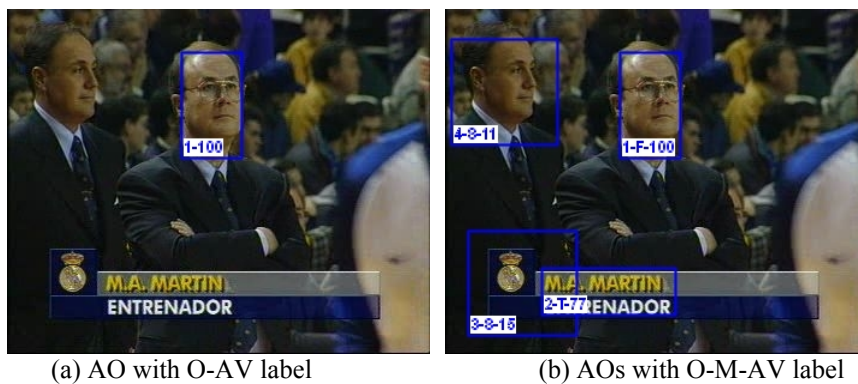


Figure 5.7: Example of AOs labels

2) Compute Optimal Path

This action button executes the Optimal Path Generation algorithms described in section 4.3. The Optimal Path Generation button in the Select Intermediate Results box is automatically selected when the algorithm ends, which allows seeing the results of the Attention Models Integration, and the Split and Group Processing in the intermediate results image boxes.

3) Create Video

This action button executes the video generation algorithm, described in section 4.4. It allows the user to choose the output video filename and one of the compression formats installed in the Windows operating system. The video player shown in Figure 5.8 is automatically open after the video sequence is created.

4) Video Player

This action button allows the user to open and view video sequences that have been previously created with the application. The File menu allows choosing the video file to be displayed; there are also three action buttons, whose functions are now described:

- Play: starts displaying the video file at the normal speed.
- Stop: stops displaying the video file.
- FF: the video is displayed at double the normal speed.

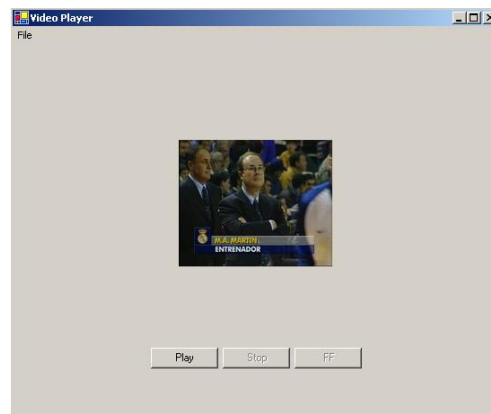


Figure 5.8: Video player window

5.3 Final Remarks

This chapter presented the interface that has been developed to integrate all the algorithms used and developed for the Image 2Video application. The interface allows the user to perceive the evolution of the input image through the adaptation system until a video sequence is created.

A description on how to use the interface, i.e. the actions that have to be performed to produce the algorithm results presented in the previous chapters was given. This description also provides a guide on how to adjust and save different configurations regarding the algorithm parameters, and how to select the results that are to be displayed in the interface. This chapter concludes the presentation of the work that has been developed throughout this project.

The next chapter will present the subjective assessment results, project conclusions and future work.

Chapter 6

Evaluation Results and Conclusions

This chapter finalizes the presentation of the project Image2Video application, which is able to transform/adapt images to video driven by visual attention targeting a final better user experience.

The chapter starts by showing the results of the subjective evaluation study that has been performed, followed by the major conclusions of the work, and ends with the description of future work to be done.

6.1 User Evaluation Study

There is no objective measure to evaluate the performance of the developed adaptation system. Therefore, the authors of this project decided to conduct a user study to evaluate the quality of the experience provided by the video clips created with the Image2Video application. In the following sections the objectives, methodology and results of the user study are presented.

6.1.1 Objectives

The purpose of the user study that has been conducted is not only to evaluate the global performance of the developed adaptation system, but also to assess the performance of the developed algorithms. The user study provides a subjective evaluation of the results provided by the Image2Video application, having three main objectives:

- Evaluate how good the video clip experience is regarding the still image experience in display size constrained devices, to determine the impact of the application on the user.
- Evaluate if all the ROIs of the image are focused in the video clip, and therefore determine the performance of the composite image attention model and attention models integration stages of the adaptation system.
- Compare the ordering of the focused ROIs in the video clip with the order by which the user would focus on ROIs, allowing to determine the adequateness of the calculated AVs and the performance of the optimal browsing path algorithm.

6.1.2 Methodology and Conditions

In order to achieve the objectives presented in the previous section, a set of 8 images with a resolution of 352x288 pixels was selected. The images are divided into four classes, each class with two images:

- **Saliency class:** the images in this class don't contain human faces or text. The purpose of this class is to evaluate the performance of the application for images without faces or text, simply relying on the saliency attention model.
- **Face class:** the images in this class contain human faces and no text. The purpose of this class is to evaluate the performance of the application for images where faces are present, and text isn't.
- **Text class:** the images in this class contain text and no human faces. The purpose of this class is to evaluate the performance of the application for images where text is present and faces aren't.
- **Mix class:** the images in this class contain both human faces and text. The purpose of this class is to evaluate the performance of the application when all types of content are present.

Based on these 8 images, the respective video clips were produced with a resolution of 110x90 pixels, to simulate viewing the image and video clip in a display size constrained device. Using the developed application interface to show the original image and the video clip, a group of 15 volunteers, mainly students from the IT and INESC research groups were invited to give their subjective judgments at the following three questions:

- **Question 1:** How good is the video experience regarding the still image experience?
a) Very bad b) Bad c) Reasonable d) Good e) Very good
- **Question 2:** Are all the interesting regions of the image focused on the video?
a) None b) Some c) Almost all d) All
- **Question 3:** How well does the focused regions order reflect their real relative importance?
a) Very bad b) Bad c) Reasonable d) Well e) Very well

The next section presents the results and analysis of the user study.

6.1.3 Results Analysis

Based on the answers provided by the volunteers involved in this study, Table 6.1, Table 6.2 and Table 6.3 contain the statistical results for all three questions.

Regarding Question 1, the average results show that 39% and 33% of the inquired consider the video experience compared to the still image experience, good and very good respectively. Furthermore, none of the inquired consider it very bad, and only 5% consider it bad. These results allow concluding that the majority of the users prefer the video clip instead of the still image. This preference is more evident for the face class images.

Regarding Question 2, the average results show that 59% of the inquired consider that all of the interesting regions of the image are focused in the video. The results also show that the application performs best in the face class images, and worst in the saliency class images which don't contain faces and text. This indicates that it is more difficult to identify ROI simply using the saliency attention model.

The order by which regions in an image are attended is particular to the human observer and scene. In spite of this, the average results for Question 3 show that the 41% and 33% of the inquired consider that the ordering of the focused regions reflects their real relative importance, well and very well respectively. Furthermore, no one considered the ordering very bad, and only 3% considered it bad.

Table 6.1: Evaluation results for Question 1

Image Class	a)	b)	c)	d)	e)
Saliency	0%	3%	33%	40%	24%
Face	0%	7%	17%	33%	43%
Text	0%	6%	20%	47%	27%
Mix	0%	3%	23%	34%	40%
Average	0%	5%	23%	39%	33%

Table 6.2: Evaluation results for Question 2

Image Class	a)	b)	c)	d)
Saliency	0%	13%	50%	37%
Face	0%	3%	23%	74%
Text	0%	3%	33%	64%
Mix	0%	7%	30%	63%
Average	0%	7%	34%	59%

Table 6.3: Evaluation results for Question 3

Image Class	a)	b)	c)	d)	e)
Total Saliency	0%	3%	30%	37%	30%
Face	0%	3%	20%	57%	20%
Text	0%	3%	17%	40%	40%
Mix	0%	3%	27%	30%	40%
Average	0%	3%	23%	41%	33%

6.2 Summary and Conclusions

Currently, the predominant methods for viewing large images on small devices are down-sampling or manual browsing by zooming and scrolling. Image down-sampling results in significant information loss, due to excessive resolution reduction. Manual browsing can avoid information loss but is often time-consuming for the users to catch the most crucial information in an image.

In this report, an adaptation system has been proposed, with a major objective: maximize the user experience when consuming an image in a device with a small size display. The developed Image2Video system is able to transform/adapt images to video driven by visual attention targeting a final better user experience.

The first two stages of the adaptation system, the image composite attention model and attention models integration were developed to provide a unique image map that contains all the ROIs of the image. Based on the evaluation study results for Question 2, is it possible to conclude that in the majority of cases the application is able to identify the ROIs of the image, and that the first two stages of the system fulfill their objective.

The third stage of the adaptation system, the optimal path generation, was developed to establish the path used to display with video the whole image, i.e. to provide the order by which AGs will be displayed. Based on the evaluation study results for Question 3, is it possible to conclude that the application succeeds in simulating the browsing path that a human would perform to analyze the image.

The main objective of the application developed in this project is to transform images into video, driven by visual attention, targeting a final better user experience in display size constrained devices. Based on the evaluation study results for Question 1, is it possible to conclude that the

developed Image2Video application achieves its main objective, i.e. the quality of the experience provided by the video clips created with the application is better than that provided by the still image experience.

The major limitation of this application is that it only works with images whose resolution is equal to or smaller than 352x288 pixels (CIF). The limitation is originated by the text detection algorithm, which was originally developed to work with CIF images, and therefore has several fixed memory structures which couldn't be changed to work robustly with bigger resolutions. Therefore, the application interface only allows that images with resolutions equal to or smaller than CIF are used.

6.3 Future Work

Although the user evaluation study results show that the application achieves the objective of its development, there is still room for improvements.

This adaptation system has a modular design, i.e. it was developed having in mind future expansions or improvements. This means that the used visual attention models can be replaced, or new ones can be added. Based on the user evaluation study results, either a better saliency attention model or new specific object detectors should be added to improve the detection of ROIs of an image.

In the composite image attention model, each ROI is delimited by a rectangular bounding-box. Instead of rectangles, other shapes such as ellipses could be used to provide a more accurate adaptation to the shape of the object contained in the ROI.

Another possible improvement regards the video generation algorithm. The addition of a new motion unit class consisting of pan and zoom simultaneously would allow substituting the currently used local zoom, panning motion and local panning motion classes. This type of motion would allow creating video sequences with smoother transitions between the focused AGs, and therefore more pleasant for the user.

Appendix A

CD-ROM Contents

The CD-ROM that was delivered together with this printed version of the report contains all the developed code, source and executable. This appendix provides a description of the CD-ROM contents which are shown in Figure A.1.

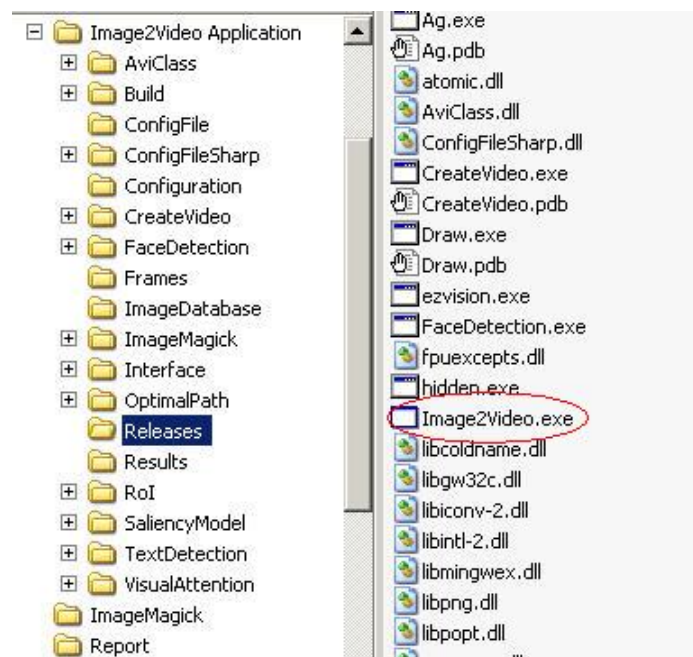


Figure A.1: CD-ROM directory structure

The CD-ROM has three directories:

- **Image2Video Application:** this directory contains all the source code and executables that are part of the application. The sub-directory Releases contains the executables. To run the application interface, the user should double-click the Image2Video.exe file, as highlighted in red in Figure A.1.
- **ImageMagick:** the application requires the installation of the Image Magick C++ version 6.0.0 toolbox to run. This directory contains the executable to install the toolbox.
- **Report:** this directory contains the pdf version of this report.

References

- [1] F. Pereira, I. Burnett, “Universal multimedia experiences for tomorrow”, IEEE Signal Processing Magazine, Special number about universal multimedia access, Vol.20, No. 2, pp. 63-73, March, 2003.
- [2] F. Pereira, “Sensations, perceptions and emotions: towards quality of experience evaluation for consumer electronics video adaptations”, First International Workshop on Video Processing and Quality Metrics for Consumer Electronics, January, 2005.
- [3] H. Liu, X. Xie, W.Y. Ma, H.J. Zhang, "Automatic browsing of large pictures on mobile devices", ACM Multimedia 2003, November, Berkeley, CA, USA.
- [4] M. M. Gupta, G. K. Knopf, *Neuro-vision systems principals and applications*, IEEE Press, New York, USA, 1994.
- [5] B. A. Wandell, *Foundations of Vision*, Sinauer Associates, Inc., Sunderland, Massachusetts, USA, 1995.
- [6] S. Coren, L. M. Ward, J. T. Enns, *Sensation and perception*, Harcourt Brace College Publishers, Fort Worth, USA, 1994.
- [7] J. M. Wolfe and T. S. Horowitz, “What attributes guide the deployment of visual attention and how do they do it”, Nature Reviews Neuroscience, Vol. 5, No. 6, pp. 495-501, June, 2004.
- [8] N. Ouerhani, *Visual attention: from bio-inspired modeling to real-time implementation*, Ph. D. Thesis, Université de Neuchatel, Neuchatel, Switzerland, 2003.
- [9] R. Desimone and J. Duncan, “Neural mechanisms of selective visual attention”, *Annual Review of Neuroscience*, Vol. 18, pp. 193–222, 1995.
- [10] L. Itti and C. Koch, “Computational modeling of visual attention”, *Nature Reviews Neuroscience*, Vol. 2, No. 3, pp. 194-203, March, 2001.
- [11] E. K. Miller, The prefrontal cortex and cognitive control, *Nature Reviews Neuroscience*, Vol. 1, No. 1, pp. 59–65, October, 2000.
- [12] R. Baptista, J. Germano, *Artificial retina: development of a bio-inspired model with configurable implementation*, Graduation Report, Instituto Superior Técnico, Lisbon, 2003.
- [13] C. M. Privitera, L. W. Stark, “Algorithms for defining visual regions-of-interest:

comparison with eye fixations”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 9, pp. 970-982, September, 2000.

[14] S. True, “Visual attention: the where, what, how and why of saliency”, *Current Opinion in Neurobiology*, Vol. 13, No. 4, pp. 428-432, August, 2003.

[15] D. Heinke and G.W. Humphreys, in *Connectionist Models in Psychology* (ed. G. Houghton), Computational models of visual selective attention: A review, Psychology Press, London, UK.

[16] A.M. Treisman and G. Gelade, “A feature-integration theory of attention”, *Cognitive Psychology*, pp. 97-136, 1980.

[17] Ch. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry”, *Human Neurobiology*, Vol. 4, pp. 219-227, 1985.

[18] L. Itti, Ch. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis”, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 20, No. 11, pp. 1254-1259, 1998.

[19] O. Le Meur, P. Le Callet, D. Barba, D. Thoreau, E. François, “From low level perception to high level perception, a coherent approach for visual attention modelling”, in proc. SPIE Human Vision and Electronic Imaging, San Jose, CA, 2004.

[20] R. M. Klein, “Inhibition of return”, *Trends Cogn. Sci.*, Vol. 4, No. 4, pp. 138-147, April, 2000.

[21] C. M. Privitera, L. W. Stark, “Algorithms for defining visual regions-of-interest: comparison with eye fixations”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 9, September 2000.

[22] T. Topper, “Selection Mechanisms in Human and Machine Vision”, University of Waterloo, Ph.D. Thesis, 1991.

[23] Bruce, N.D.B., Jernigan, M.E., “Evolutionary design of context-free attentional operators”, ICIP03(I: 429-432).

[24] A. Oliva, A. Torralba, M. S. Castelhana, J. M. Henderson, “Top down control of visual attention in object detection”, in *IEEE Proceedings of the International Conference on Image Processing*, vol. I, Barcelona, Spain, pp. 253-256, September, 2003.

[25] S. Frintropa and E. Rome, “Simulating visual attention for object recognition”, Proceedings of the Workshop on Early Cognitive Vision, Isle of Skye, Scotland, May 2004.

[26] A. Torralba, “Contextual Priming for Object Detection”, *IJCV*, Vol. 53, pp. 153-167, 2003.

[27] P. Viola, M. Jones, “Robust real-time object detection”, in Proc. 2nd Int’l Workshop on Statistical and Computational Theories of Vision – Modelling, Learning, Computing and Sampling, Vancouver, Canada.

[28] F. Miau, L. Itti, “A neural model combining attentional orienting to object recognition: preliminary explorations on the interplay between where and what”, in: *Proc. IEEE Engineering in Medicine and Biology (EMBS)*, Istanbul, Turkey, October 2001

-
- [29] M. Riesenhuber, T. Poggio, "Hierarchical models of object recognition in cortex", *Nature Neuroscience*, 2(11):1019-1025, 1999 November, 1999.
- [30] A. Maki, J. Eklundh, P. Nordlund, "A Computational model of depth-based attention", *Proceedings of the International Conference on Pattern Recognition (ICPR '96) Volume IV-Volume 7472*, p.734, 25-29, August, 1996.
- [31] M. Riesenhuber, T. Poggio, "Hierarchical models of object recognition in cortex", *Nature Neuroscience*, 2(11):1019-1025, November, 1999.
- [32] L. Chen, X. Xie, X. Fan, W. Ma, H. Zhang, H. Zhou, "A visual attention model for adapting images on small displays", *ACM Multimedia Systems Journal*, Vol.9, No.4, pp. 353-364, 2003.
- [33] T. Fong, I. Nourbakhsh, and K. Dautenhahn, "A Survey of socially interactive robots", *Robotics and Autonomous Systems*, 42(3-4), pages 143-166, 2003.
- [34] C. Siagian, L. Itti, "Biologically-inspired face detection: non-brute-force-search approach", In: *First IEEE International Workshop on Face Processing in Video*, June 2004.
- [35] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention", *IEEE Transactions on Image Processing*, Vol. 13, No. 10, pp. 1304-1318, October, 2004.
- [36] J. Ascenso, P.L. Correia, F. Pereira; "A face detection solution integrating automatic and user assisted tools ", *Proc Portuguese Conf. on Pattern Recognition - RecPad* , Porto , Portugal , Vol. 1 , pp. 109 - 116 , May, 2000.
- [37] D. Palma, "Automatic text extraction in digital video sequences", Instituto Superior Técnico, Lisboa, Master Thesis, (2004).
- [38] L.Q. Chen, X. Xie, W.Y. Ma, H.J. Zhang, H.Q. Zhou, "Image adaptation based on attention model for small-form-factor devices", *The 9th International Conference on Multi-Media Modeling*, Taipei, Taiwan, January, 2003.