



UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Extracção Automática de Texto em Sequências de Vídeo

Duarte Manuel da Conceição Palma
(Licenciado)

Dissertação para obtenção do Grau de Mestre em
Engenharia Electrotécnica e de Computadores

Orientador: Doutor Fernando Manuel Bernardo Pereira

Júri:

Presidente: Doutor Fernando Manuel Bernardo Pereira

Vogais: Doutor Nuno Manuel Carvalho Ferreira Guimarães

Doutor Mário Alexandre Teles de Figueiredo

Julho 2004

**... nenhum homem
é uma ilha ...**

Resumo

Os avanços tecnológicos registados nos últimos anos na área da tecnologia audiovisual levaram à explosão do uso da informação audiovisual, nomeadamente acedida via Internet, por um vasto número de utilizadores. O aumento vertiginoso da produção de conteúdos audiovisuais tem levado os operadores de televisão e outros produtores de conteúdos audiovisuais a demonstrar interesse na construção de bibliotecas digitais que permitam o arquivo de conteúdos multimédia para posterior reutilização. Para satisfazer esta necessidade são necessários sistemas capazes de tratar a informação audiovisual em termos de armazenamento, transmissão, procura e visualização. Desta forma, a existência de sistemas, automáticos e eficientes, de descrição, indexação e procura de conteúdos multimédia torna-se necessária e, para isso, vários métodos de processamento têm vindo a ser desenvolvidos ao longo dos últimos anos. Em muitos destes métodos privilegia-se a informação textual, existente nas imagens e nos vídeos, que por ser uma fonte de informação com elevado valor semântico torna possível a indexação e procura de conteúdos de forma acessível e intuitiva a produtores e consumidores. Neste contexto, a presente Tese considera o problema da extracção automática de texto em imagens e vídeos. Assim, objectivos a atingir com a presente Tese são:

1. Revisão bibliográfica das principais técnicas disponíveis para a extracção automática de texto em imagens e vídeo, nomeadamente técnicas de segmentação, classificação e reconhecimento;
2. Desenvolvimento, implementação e avaliação de um mecanismo de extracção automática de texto para imagens sem limitações significativas em termos de conteúdo;
3. Extensão do mecanismo desenvolvido para imagens para extracção automática de texto em sequências de vídeo e avaliação do seu desempenho para vários tipos de conteúdo.

Palavras-chave: Extracção de texto, segmentação, detecção de fronteiras, detecção de caracteres, seguimento de texto, formação de palavras, reconhecimento de texto.

Abstract

The technological advances seen in the last years in the area of audiovisual technology have led to a boom in the usage of audiovisual information, namely accessed through the Internet, by a growing number of users. The increasing amount of audiovisual information being deployed has led television operations and audiovisual content producers to show interest in building digital libraries which could allow the storage of audiovisual information for future retrieval. To address this need, it is necessary to have systems capable of treating audiovisual information targeting its efficient storage, transmission, searching and viewing. Thus the necessity of automatic and efficient systems capable of analyzing, describing, filtering and retrieving multimedia information. With this objective in mind, several methods have been developed in last few years. Many of these methods target the textual information that exists in images and video sequences since it is a source of highly semantic information and thus allows the filtering and searching of this information by producers and consumers in a more intuitive and natural way. In this context, this Thesis will study the problem of automatically extracting text from images and video sequences. Therefore the objectives of this Thesis are:

1. Bibliographic study and comparative analysis of the most relevant analysis techniques available for extracting text in images and video sequences, namely segmentation, classification and recognition techniques;
2. Development, implementation and evaluation of an automatic text extraction mechanism for images without significant limitations in terms of the text characteristics;
3. Extension of the mechanism developed for images for the automatic extraction of text in video sequences and evaluation of its performance using several types of content.

Keywords: Text extraction, segmentation, edge detection, character detection, text tracking, word formation, text recognition.

Agradecimentos

Para começar, gostaria de agradecer ao Professor Fernando Pereira o acompanhamento rigoroso e metódico que sempre me dedicou em todas as fases da tese, revelando, sempre, uma incrível dedicação e quase infinita paciência.

À minha esposa, Elsa Duarte, pelo apoio incontestável, pelo incentivo e pela paciência demonstrada durante o desenvolvimento da Tese.

Aos meus filhos, que durante dois longos anos, não puderam, de forma plena, usufruir da cumplicidade do pai.

Ao colega João Ascenso por todo o incentivo, tempo perdido e apoio prestado sobretudo na parte de programação.

A todos os colegas do Grupo de Imagem pelo bom ambiente que sempre criaram, pelas sugestões que foram dando e pela disponibilidade permanente que, em muito, facilitou o trabalho desenvolvido.

A todos os colegas da Marinha e em especial ao Hugo Coelho por todo o incentivo e apoio que me deram.

Para finalizar, um agradecimento muito especial a Todos Aqueles que, de forma subtil e indelével, ajudaram a tornar esta tese possível através dos seus sopros inspirativos e incentivos magnânicos.

Índice

Capítulo 1 Introdução	1
1.1 Contexto e Motivação.....	2
1.2 Objectivos	3
1.3 Conceitos e Terminologia Relevantes.....	4
1.4 Organização da Tese	8
Capítulo 2 Extracção de Texto em Imagens e Vídeo: Revisão Bibliográfica.....	11
2.1 Arquitectura Básica.....	12
2.2 Técnicas de Segmentação de Imagem e Vídeo	15
2.2.1 Segmentação Espacial.....	17
2.2.2 Segmentação Temporal.....	25
2.2.3 Combinação da Segmentação Espacial e Temporal	28
2.3 Técnicas de Classificação das Regiões.....	29
2.3.1 Ferramentas de Descrição de Regiões.....	30
2.3.2 Métodos Utilizados na Classificação das Regiões.....	35
2.4 Técnicas de Seguimento	36
2.5 Técnicas de Reconhecimento de Texto	37
2.5.1 Métodos de Decisão Teórica.....	37
2.5.2 Métodos Estruturais	39
2.6 Sistemas de Extracção de Texto mais Relevantes	40
2.6.1 Extracção de Texto Gráfico e de Cena em Imagens	40
2.6.2 Extracção Automática de Texto Gráfico para Indexação de Vídeo.....	50
2.6.3 Extracção de Texto em Imagens e Vídeos	62
2.6.4 Extracção de Texto em Sequências de Vídeo com Integração de Múltiplas Tramas	74
2.7 Comentários Finais	85
Capítulo 3 Algoritmo Para Extracção de Texto em Imagens	91
3.1 Arquitectura Básica.....	91
3.2 Detecção de Texto.....	93
3.2.1 Simplificação.....	94
3.2.2 Segmentação.....	102
3.2.3 Detecção de Caracteres	114
3.2.4 Detecção de Palavras.....	119
3.3 Reconhecimento de Texto.....	127
3.3.1 Sistema OCR Comercial.....	127

3.3.2	OCR Integrado na Aplicação.....	128
3.4	Avaliação de Desempenho.....	129
3.4.1	Métricas de Desempenho.....	129
3.4.2	Condições e Metodologia de Avaliação do Desempenho	131
3.4.3	Resultados e Comentários.....	135
3.5	Comentários Finais	146
Capítulo 4	Extracção de Texto em Sequências de Vídeo.....	149
4.1	Arquitectura Básica.....	150
4.2	Detecção de Texto em Vídeo	152
4.2.1	Monitorização do Texto	155
4.2.2	Formação de Sequências de Texto.....	156
4.2.3	Análise de Movimento	158
4.3	Reconhecimento de Texto.....	179
4.4	Avaliação de Desempenho.....	180
4.4.1	Métricas de Desempenho.....	180
4.4.2	Condições e Metodologia de Avaliação do Desempenho	181
4.4.3	Resultados e Análise	185
4.5	Comentários Finais	195
Capítulo 5	Sumário e Trabalho Futuro	197
Bibliografia	201

Lista de Figuras

Figura 1.1 - Exemplos de objectos: (a) objecto simples constituído por uma única região; (b) objecto complexo constituído por mais do que uma região [MPEG7-Visual01].	5
Figura 1.2 – Exemplo de contorno: (a) imagem com um objecto principal; (b) contorno do objecto principal em (a) [MPEG7-Visual01].	5
Figura 1.3 – Exemplos de <i>bounding boxes</i> : (a) forma de objecto simples e <i>bounding box</i> correspondente; (b) forma de objecto complexo e <i>bounding box</i> correspondente [MPEG7-Visual01].	6
Figura 1.4 – Exemplos dos dois tipos de texto: (a) texto de cena; (b) texto gráfico.	8
Figura 2.1 – Exemplo de procura baseada na extracção de texto em imagens ou vídeos.	12
Figura 2.2 – Arquitectura básica para a extracção de texto em sequências de vídeo.	13
Figura 2.3 – Exemplo da extracção de texto para uma trama de vídeo: (a) imagem original; (b) imagem segmentada; (c) imagem com as regiões classificadas como texto; (d) imagem depois do refinamento da detecção através da exploração da redundância temporal; (e) texto resultante da aplicação de um sistema OCR à imagem resultante da fase de seguimento [Lienhart00].	15
Figura 2.4 – Exemplo de segmentação baseada na amplitude: (a) imagem original; (b) segmentação com um único limiar da imagem em (a) [Pham02].	18
Figura 2.5 – Exemplo de segmentação espacial baseada na textura: (a) imagem original constituída por vários tipos de textura; (b) regiões correspondentes à segmentação da imagem em (a) [Liu02].	19
Figura 2.6 – Exemplos de texturas: (a) textura aleatória; (b) textura determinística [MPEG7-Visual01].	20
Figura 2.7 – Exemplos da detecção de fronteiras: (a) imagem original; (b), (c) e (d) resultado da detecção de fronteiras utilizando os operadores de Prewitt, Roberts e Robison, respectivamente, para imagem em (a).	22
Figura 2.8 – Exemplos de segmentação usando <i>split-and-merge</i> : (a) imagem original; (b) fim da fase de <i>splitting</i> da imagem em (a); (c) fim da fase de <i>merging</i> aplicada à imagem resultante do <i>splitting</i> .	25
Figura 2.9 – Exemplo do 1º passo da segmentação temporal baseada na detecção de alterações: a diferença entre as componentes de luminância das imagens (a) e (b) é apresentada em (c) [Correia02].	26
Figura 2.10 – <i>Bream</i> (trama 1): (a) Imagem com um objecto; (b) contorno do objecto em (a).	30
Figura 2.11 – Exemplos de objectos simples e complexos, com as respectivas regiões e buracos [MPEG7-Visual01].	31
Figura 2.12 – Exemplos de imagens com: (a) cor altamente estruturada; (b) cor altamente não-estruturada [MPEG7-Visual01].	33

Figura 2.13 – Exemplos de movimentos de câmara de filmar: (a) <i>tracking</i> , <i>booming</i> e <i>dollying</i> ; (b) <i>panning</i> , <i>tilting</i> e <i>rolling</i> [MPEG7-Visual01].	34
Figura 2.14 – Exemplo de uma cadeia de quatro códigos: (a) 4 linhas de código direccionais e (b) representação de uma região por uma cadeia de códigos [Gonzalez93].	39
Figura 2.15 – Arquitectura do sistema de extracção de texto em imagens proposto em [Messelodi99].	41
Figura 2.16 – Exemplo do efeito da normalização da intensidade da luminância: (a) imagem original e (b) resultado da normalização utilizando uma janela com um tamanho de 13 <i>pixels</i> da imagem em (a) [Messelodi99].	42
Figura 2.17 – Resultado da binarização aplicada à imagem anteriormente normalizada, Figura 2.16: (a) imagem com texto normal e (b) imagem com texto inverso [Messelodi99].	43
Figura 2.18 – Resultado da aplicação dos filtros heurísticos às imagens binárias resultantes da segmentação (Figura 2.17): (a) imagem com texto normal e (b) imagem com texto inverso [Messelodi99].	44
Figura 2.19 – Exemplo do efeito da formação de linhas utilizando a primeira condição, i.e. a proximidade entre regiões, com $Th_{dist}=30$ <i>pixels</i> : (a) imagem com o conjunto inicial de regiões classificadas como texto e (b) imagem com os três subconjuntos resultantes da divisão da imagem em (a) realçados a cores diferentes [Messelodi99].	46
Figura 2.20 – Exemplo do cálculo da direcção de subconjunto: (a) subconjunto de caracteres (cinzento) e os seus centros (pontos pretos); (b) histograma calculado segundo o declive dos segmentos de recta formados entre cada par de centros e (c) projecção do histograma [Messelodi99].	48
Figura 2.21 – Arquitectura do sistema de extracção de texto gráfico em sequências de vídeo proposto em [Lienhart00].	51
Figura 2.22 – Exemplo da segmentação da cor utilizando a técnica <i>region-growing</i> : (a) trama original; (b) resultado da segmentação da cor [Lienhart00].	52
Figura 2.23 – Exemplo da imagem binária de contraste dilatada [Lienhart00].	53
Figura 2.24 – Segmentação depois da análise geométrica (246 regiões) [Lienhart00].	54
Figura 2.25 – Exemplo de texto rodeado por uma aura para melhorar a sua legibilidade [Lienhart00].	54
Figura 2.26 – Exemplo da direcção da escrita, correspondente à linha vermelha [Lienhart00].	55
Figura 2.27 – Segmentação depois da análise de textura (242 regiões) [Lienhart00].	56
Figura 2.28 – Exemplo da formação das palavras: (a) imagem original e (b) resultado da formação de palavras para a imagem em (a) [Lienhart00].	58
Figura 2.29 – Exemplo do reconhecimento de texto utilizando o OCR Recognita V3.0: (a) imagem depois da formação de palavras; (b) resultado do reconhecimento do texto existente na imagem em (a) [Lienhart00].	58
Figura 2.30 – Arquitectura do sistema de extracção de texto em imagens e vídeo proposto em [Lienhart02].	63

Figura 2.31 – Exemplo da integração dos resultados da classificação efectuada pela rede neuronal para as várias resoluções de modo a formar o mapa saliente [Lienhart02].	65
Figura 2.32 – Exemplo com o perfil das projecções utilizadas na formação de palavras ou linhas de texto: (a) projecção horizontal e (b) projecção vertical [Lienhart02].	66
Figura 2.33 – Relação entre a fase de análise do vídeo e a fase de seguimento do texto [Lienhart02].	68
Figura 2.34 – Exemplos de blocos de texto segundo o método proposto em [Li02].	74
Figura 2.35 – Arquitectura do sistema de extracção de texto em sequências de vídeo com integração de múltiplas tramas proposto em [Li02].	75
Figura 2.36 – Processos de detecção e seguimento de texto para sequências de vídeo [Li02].	76
Figura 2.37 – Arquitectura do detector de texto nas tramas de vídeo [Li02].	77
Figura 2.38 – Exemplos das curvas obtidas para MSE_r e MSE_b para três sequências de vídeo. Linha de estrelas (*) representa a desigualdade entre a trama de referência e a trama corrente (MSE_r). A linha sólida (–) representa a desigualdade entre a trama n e a trama $n+1$ (MSE_b). (a) ilustra o seguimento do texto numa sequência de vídeo com o fundo simples, (b) ilustra o seguimento do texto numa sequência de vídeo com o fundo complexo e (c) ilustra o seguimento dos números nas camisolas de jogadores de futebol [Li02].	79
Figura 2.39 – Exemplos de trajectórias: (a) correspondente à sequência da Figura 2.38(b) e (b) correspondente à sequência da Figura 2.38(c) [Li02].	81
Figura 2.40 – Exemplos de binarização: (a) bloco de texto extraído de uma trama de vídeo; (b) binarização com Th global da imagem em (a); (c) binarização com Th adaptativo utilizando o método de Niblack's da imagem em (a)[Li02].	83
Figura 2.41 – Exemplos do aumento da precisão do reconhecimento quando se utilizam várias tramas na extracção do texto: (a) resultado do OCR utilizando uma trama quando o texto é estático e o fundo possui movimento; (b) resultado do OCR utilizando varias tramas quando o texto é estático e o fundo possui movimento; (c) resultado do OCR utilizando uma trama quando o texto e o fundo possuiu movimento e (c) resultado do OCR utilizando várias tramas quando o texto e o fundo possuem movimento [Li02].	85
Figura 3.1 – Arquitectura básica proposta para o algoritmo de extracção de texto em imagens.	92
Figura 3.2 – Arquitectura do processo de detecção de texto.	94
Figura 3.3 – Arquitectura do filtro iterativo para simplificação da imagem.	95
Figura 3.4 – Arquitectura do esquema de detecção de fronteiras de Canny.	96
Figura 3.5 – Exemplo das várias fases da filtragem de uma imagem com o filtro iterativo proposto (três iterações): (a) imagem original; (b) detecção de fronteiras; (c) fronteiras classificadas como candidatas a caracteres; (d) filtragem mediana da imagem original fora das zonas de fronteira classificadas como candidatas a caracteres.	99

Figura 3.6 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico <i>open-close</i> com reconstrução: (a) e (b) imagens originais; (c) e (d) imagens filtradas com o filtro morfológico <i>open-close</i> com reconstrução usando uma janela de 3×3 <i>pixels</i> ; (e) e (f) imagens filtradas com o filtro iterativo proposto (três iterações).....	100
Figura 3.7 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico <i>open-close</i> com reconstrução a uma imagem onde os caracteres se encontram muito próximos uns dos outros: (a) imagem original; (b) imagem filtrada com o filtro iterativo proposto (três iterações); (c) imagem filtrada com o filtro morfológico <i>open-close</i> com reconstrução utilizando uma janela de 3×3 <i>pixels</i>	101
Figura 3.8 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico <i>open-close</i> com reconstrução a uma imagem onde as fronteiras entre os caracteres e o fundo da imagem são pouco contrastadas: (a) imagem original; (b) imagem filtrada com o filtro iterativo proposto (três iterações); (c) imagem filtrada com o filtro morfológico <i>open-close</i> com reconstrução utilizando uma janela de 3×3 <i>pixels</i>	101
Figura 3.9 – Arquitectura do algoritmo de segmentação das imagens adoptado.	102
Figura 3.10 – Exemplo da fase de <i>split</i> : (a) valor da luminância para os <i>pixels</i> da imagem; (b) <i>ID</i> dos <i>pixels</i> antes do início da fase de <i>split</i> (cada <i>pixel</i> uma região); (c) regiões formadas depois da fase de <i>split</i> com $Th_{split}=3$ e (d) <i>ID</i> das regiões no final da fase de <i>split</i>	103
Figura 3.11 – Exemplo da aplicação da fase de <i>split</i> : (a) imagem original; (b), (c) e (d) imagens divididas em 26337, 19020 e 15546 regiões depois da fase de <i>split</i> com Th_{split} igual a 30, 45 e 60, respectivamente.	104
Figura 3.12 – Exemplo da aplicação da fase de <i>merge</i> a uma imagem dividida com $Th_{split}=30$: (a) imagem original; (b), (c) e (d) imagens segmentadas depois da fase de <i>merge</i> com Th_{merge} igual a 35, 50 e 65, respectivamente.	105
Figura 3.13 – Exemplo da aplicação da eliminação de pequenas regiões rodeadas por uma única região: (a) imagem segmentada depois da fase de <i>merge</i> ; (b) imagem segmentada depois de eliminadas as pequenas regiões.	106
Figura 3.14 – Exemplo do efeito da sobresegmentação originada pelo ruído existente nas imagens: (a) imagem original; (b) imagem segmentada depois das fases de <i>split</i> e <i>merge</i> ; (c) resultado ideal da eliminação das pequenas regiões e (d) resultado utilizando o algoritmo proposto.	107
Figura 3.15 – Arquitectura do processo de melhoramento de fronteiras através da eliminação de pequenas regiões.	108
Figura 3.16 – Exemplo da aplicação do tensor de inércia: (a) imagem original; (b) imagem com as regiões onde o contraste local é superior a 0.001 e a anisotropia é superior a 0.5.	111
Figura 3.17 – Exemplo da aplicação da eliminação de pequenas regiões: (a) imagem original; (b) imagem segmentada antes de eliminadas as pequenas regiões; (c) imagem depois de eliminadas as pequenas regiões completamente rodeadas por uma única região e (d) imagem depois de eliminadas as pequenas regiões em cuja vizinhança existe mais de uma região.	113

Figura 3.18 – Exemplo do melhoramento das fronteiras dos caracteres devido à eliminação de pequenas regiões: (a) imagem original; (b) imagem segmentada antes da eliminação de pequenas regiões; (c) imagem depois de efectuada a eliminação de pequenas regiões.	114
Figura 3.19 – Arquitectura do processo proposto para a análise de contraste.	115
Figura 3.20 – Exemplo da aplicação da análise do contraste para efectuar a classificação das regiões: (a) imagem original; (b) imagem de fronteiras; (c) imagem de contraste com ($Th_{cont}=10$); (d) resultado da dilatação, com $n=3$, da imagem binária resultante da adição das imagens (b) e (c); (e) imagem segmentada; (f) imagem com as regiões classificadas como provável texto depois de efectuada a análise de contraste.	117
Figura 3.21 – Exemplo da aplicação da análise geométrica: (a) imagem segmentada; (b) imagem com as regiões classificadas como texto depois da aplicação da análise geométrica, sem a aplicação prévia da análise do contraste; (c) imagem com as regiões classificadas como texto depois da aplicação da análise geométrica e posteriormente à aplicação da análise do contraste.	119
Figura 3.22 – Exemplo da detecção de palavras: (a) imagem original; (b) imagem segmentada depois de efectuada a análise de contraste e a análise geométrica; (c) imagem depois da formação de palavras.	123
Figura 3.23 – Exemplo de sobreposição de palavras: (a) e (c) imagens com sobreposição de palavras antes de efectuada a rotação do texto; (b) e (d) as imagens com sobreposição de palavras depois da rotação do texto.	124
Figura 3.24 – Exemplo de texto vertical (a) e inclinado (b).	125
Figura 3.25 – Exemplo da rotação de texto: (a) e (b) originais de texto vertical e inclinado, respectivamente; (c) e (d) texto detectado antes da rotação; (e) e (f) texto detectado depois da rotação.	127
Figura 3.26 – Cálculo dos vectores de características para o reconhecimento óptico de caracteres [Lienhart95]: (a) divisão do carácter em nove segmentos (b) os 16 elementos de direcção.	129
Figura 3.27 – Exemplos de imagens que fazem parte do conjunto de teste: (a) imagens onde predomina o texto de cena; (b) imagens onde predomina o texto gráfico.	135
Figura 3.28 – Exemplo de regiões falsamente classificadas como texto devido à sua forma e posicionamento: (a) imagem original e (b) imagem binária com o resultado da detecção de texto.	138
Figura 3.29 – Exemplo de falhas na detecção de texto devido ao baixo contraste existente entre o texto e fundo da imagem: (a) imagem original; (b) imagem com o resultado da segmentação e (c) imagem binária com o resultado da detecção de texto.	138
Figura 3.30 – Exemplo de falhas na detecção de texto devido ao contacto entre os caracteres: (a) imagem original; (b) imagem com o resultado da segmentação e (c) imagem binária com o resultado da detecção de texto.	139
Figura 3.31 – Exemplo de caracteres danificados: (a) e (b) imagens originais; (c) e (d) imagens binárias com texto detectado.	141
Figura 3.32 – Exemplo da dificuldade evidenciada pelo OCR OmniPage Pro 12.0 em reconhecer (directamente) texto em imagens com fundos complexos: (a)	

imagem original e (b) resultado do reconhecimento de texto efectuado pelo OCR.	144
Figura 3.33 – Exemplo da influência do algoritmo de detecção de texto proposto, no reconhecimento de texto efectuado pelo OCR OmniPage Pro 12.0: (a) e (b) imagens originais; (c) e (d) resultados do reconhecimento de texto efectuado pelo OCR; (e) e (f), imagens fornecidas ao OCR pelo algoritmo de detecção de texto; (g) e (h) resultados do reconhecimento de texto efectuado pelo OCR em conjunto com o algoritmo de detecção de texto.	146
Figura 4.1 – Exemplos de imagens para as quais se pode justificar a extracção de texto para: (a) analisar o evento desportivo; (b) classificar o programa em questão.	150
Figura 4.2 – Arquitectura básica do algoritmo de extracção de texto em sequências de vídeo.	151
Figura 4.3 – Relação entre a monitorização do vídeo (1ª fase), a formação de sequências de texto (2ª fase) e a análise do movimento (3ª fase).	153
Figura 4.4 – Arquitectura do processo de detecção de texto em vídeo.	154
Figura 4.5 – Arquitectura do processo de detecção de texto para cada trama de vídeo analisada.	156
Figura 4.6 – Exemplos de sequências de texto.	157
Figura 4.7 – Exemplo dos resultados obtidos com a análise do movimento: (a) tramas da sequência de vídeo com texto; (b) imagens com a detecção do texto para cada trama individual; e (c) imagem final resultante da integração de todo o texto existente na sequência.	158
Figura 4.8 – Exemplo das vantagens do seguimento do texto ao nível do carácter versus seguimento ao nível da palavra.	159
Figura 4.9 – Arquitectura do processo de análise de movimento em sequências de vídeo. ..	160
Figura 4.10 – Exemplo da divisão da <i>bounding box</i> de um carácter em quatro sectores.	162
Figura 4.11 – Exemplo da formação de cadeias de carácter: as figuras geométricas correspondem a regiões classificadas como caracteres. C_1 e C_2 representam cadeias de caracteres válidos; C_3 e C_4 representam cadeias de caracteres inválidos.	165
Figura 4.12 – Exemplo da formação de uma palavra P_1 formada a partir de três cadeias de carácter, CC_1 , CC_2 e CC_3 . Os quadrados azuis representam as tramas onde as cadeias de carácter e a palavra são detectadas, os quadrados vermelhos representam tramas onde as cadeias de carácter e a palavra não são detectadas e os quadrados cinzentos representam tramas onde a palavra está incompleta.	170
Figura 4.13 – Exemplo da interpolação das regiões em falta numa cadeia de carácter. A imagem (a) ilustra a cadeia de carácter antes da recuperação de caracteres; a imagem (b) ilustra o efeito da expansão da cadeia de carácter e as imagens (c) e (d) ilustram a interpolação dos caracteres em falta, baseada na trama de trás e na trama da frente, respectivamente. Os quadrados a azul e a vermelho, representam as tramas onde os caracteres foram detectados e onde a sua detecção falhou, respectivamente.	172
Figura 4.14 – Exemplo da recuperação de caracteres	174

Figura 4.15 – Exemplo da recuperação de regiões: (a) imagens originais; (b) resultados da detecção de texto aplicada às tramas individualmente; e (c) resultado da detecção de texto com análise de movimento e recuperação de caracteres perdidos.	176
Figura 4.16 – Exemplo do tipo de imagens utilizadas para visualizar os resultados: (a) imagem binária com a representação de cada palavra na sua localização original; (b) imagem binária com a representação de cada palavra extraída depois da rotação para a direcção horizontal; (c) imagem binária com a representação de todas as palavras extraídas da sequência de vídeo, segundo a direcção horizontal.	179
Figura 4.17 – Exemplo do efeito da recuperação de regiões: (a) imagem original; (b) resultado da detecção de texto efectuada sobre uma trama individualmente; (c) resultado da detecção de texto com análise de movimento e recuperação de caracteres perdidos.	179
Figura 4.18 – Exemplos com vários tipos de texto em vídeo: (a) texto de cena; (b) texto gráfico com movimento e (c) texto gráfico fixo.	182
Figura 4.19 – Exemplos de sequências de vídeo que fazem parte do conjunto de teste: (a) sequências onde o texto possui movimento; (b) sequências onde o texto está fixo e o fundo da imagem se movimenta; e (c) sequências onde se movimenta quer o texto, quer o fundo da imagem, com movimentos semelhantes.	184
Figura 4.20 – Exemplo de falha no seguimento do texto devido a alterações fortes na direcção do movimento do texto: (a) sequência de texto; (b) resultado da detecção de texto.	189
Figura 4.21 – Exemplo da imagem resultante da integração do texto existente numa sequência de vídeo: (a) sequência de vídeo; (b) imagem resultante da integração do texto existente na sequência de vídeo.	190
Figura 4.22 – Exemplo de reconhecimento de texto com o OmniPage Pro 12.0: (a) tramas representativas do texto existente no vídeo; (b) resultados do reconhecimento efectuada pelo OmniPage Pro 12.0 para as tramas em (a).	193

Lista de Tabelas

Tabela 2.1 – Desempenho em termos da classificação das regiões para as várias condições heurísticas.....	49
Tabela 2.2 – Desempenho em termos da selecção de linhas.....	50
Tabela 2.3 – Desempenho em termos de detecção de texto.	60
Tabela 2.4 – Desempenho em termos de reconhecimento do texto.	60
Tabela 2.5 – Desempenho em termos de pesquisa textual.	61
Tabela 2.6 – Desempenho em termos de detecção do texto.....	72
Tabela 2.7 – Desempenho em termos de segmentação do texto.	73
Tabela 2.8 – Desempenho em termos de reconhecimento do texto.	73
Tabela 2.9 – Desempenho em termos de reconhecimento de texto para os vários tipos de conteúdos.	84
Tabela 2.10 – Sumário das vantagens e desvantagens das técnicas de segmentação apresentadas.	86
Tabela 2.11 – Sumário das vantagens e desvantagens dos métodos de classificação maioritariamente utilizados na extracção de texto.	87
Tabela 2.12 – Sumário das vantagens e desvantagens dos dois tipos de abordagens apresentados para efectuar o seguimento do texto.....	87
Tabela 2.13 – Resumo das características dos sistemas de extracção de texto apresentados.....	88
Tabela 3.1 – Parâmetros utilizados para a avaliação do desempenho.....	135
Tabela 3.2 – Resultados médios obtidos para a detecção de texto horizontal para a totalidade das imagens.	136
Tabela 3.3 – Resultados médios obtidos para a detecção de todo o texto para a totalidade das imagens.	136
Tabela 3.4 – Resultados médios obtidos para a detecção de texto em termos de caracteres não detectados e caracteres danificados.....	140
Tabela 3.5 – Resultados médios obtidos para o reconhecimento do texto horizontal.	141
Tabela 3.6 – Resultados médios obtidos para o reconhecimento de todo o texto.....	142
Tabela 3.7 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da <i>ground truth</i> , utilizando unicamente o OCR OmniPage Pro 12.0 e utilizando o algoritmo de detecção de texto em conjunto com o OCR OmniPage Pro 12.0.	143
Tabela 4.1 – Parâmetros utilizados para a avaliação do desempenho.....	185
Tabela 4.2 – Resultados médios obtidos para a detecção de todo o texto que faz parte da <i>ground truth</i> para a totalidade dos vídeos.	186

Tabela 4.3 – Resultados médios obtidos para a detecção de texto gráfico com movimento e texto gráfico fixo.	187
Tabela 4.4 – Resultados médios obtidos para a detecção de todo o texto que faz parte da <i>ground truth</i> , quer para o conjunto de teste de 60 imagens, quer para o conjunto de teste das 13 sequências de vídeo.	188
Tabela 4.5 – Resultados médios obtidos para o reconhecimento de todo o texto na <i>ground truth</i> para a totalidade dos vídeos.	191
Tabela 4.6 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da <i>ground truth</i> , utilizando unicamente o OCR OmniPage Pro 12.0 e utilizando o algoritmo de detecção de texto proposto em conjunto com o OCR OmniPage Pro 12.0.	193
Tabela 4.7 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da <i>ground truth</i> , quer para o conjunto de teste de 60 imagens, quer para o conjunto de teste das 13 sequências de vídeo.	194

Acrónimos

ART – *Angular Radial Transform*

CC – Correctamente Criadas

CCD – Caracteres Correctamente Detectados

CCR – Caracteres Correctamente Reconhecidos

CIF – *Common Intermediate Format*

CGT – Caracteres da *Ground Truth*

CSO – Caracteres na Saída do OCR

CSS – *Curvature Scale Space*

FC – Falsamente Criadas

GD – Gama Dinâmica

ISO – *International Standards Organization*

LMS – *Least Mean Square*

MPEG – *Motion Picture Experts Group*

MLEV – *Multi Layer Eigen Vectors*

MRF – *Markov Random Fields*

MSE – *Mean Square Error*

NC – Não Criadas

OCR – *Optical Character Recognition*

PPI – *Pixels Per Inch*

RCE – Rácio de Caracteres Errados

RCET – Rácio de Caracteres Errados Trama

RCR – Rácio de Caracteres Reconhecidos

SSD – *Sum of Squared Differences*

TDC – Total de Caracteres Detectados

TPS – Tramas Por Segundo

Capítulo 1

Introdução

A tecnologia nasce com o Homem e não lhe é meramente adjectiva. A posição vertical e a consequente libertação da mão possibilitou-lhe tornar-se *homo faber*, revelando-o imediatamente como *homo sapiens*. A produção de objectos técnicos, por mais rudimentares que sejam, mostrou-o racional, organizador, pesquisador, descobridor. A aliança, hoje tão profunda e claramente perceptível, entre ciência e tecnologia encontra-se já, de facto, embora de forma apenas incoativa, nas primeiras manifestações da história do Homem. Essa história encontra na evolução da tecnologia um dos critérios para a sua divisão em grandes períodos ou fases. De acordo com Heidegger, filósofo Alemão do séc. XX, e na perspectiva da longa duração, é lícito distinguir três idades principais: a idade do instrumento, a idade da máquina e a idade da cibernética [Heidegger29]. De um período a outro, de uma fase a outra, é-nos dado assistir à progressiva autonomização da tecnologia em relação ao seu indispensável produtor, o Homem. Na primeira fase, o Homem utiliza meios mecânicos que quase mais não são que simples prolongamentos da própria mão. Na segunda fase, ele serve-se de instrumentos mais poderosos que lhe poupam esforços físicos e que dispõem já de uma certa auto-suficiência: máquina a vapor, máquina eléctrica, etc ... Finalmente, na terceira fase, ele constrói ‘escravos mecânicos’ dirigidos que o libertam de grandes esforços de atenção e aplicação mentais, lhe dão uma grande segurança e lhe permitem conquistas sobre a natureza de outro modo irrealizáveis.

Encontramo-nos hoje, e de forma irrefreável, em plena era da cibernética e nela imperam os meios audiovisuais. O recurso a estes meios, que vão especialmente ao encontro da vista e do ouvido, alicerçam-se no facto de serem estes os dois sentidos mais particularmente privilegiados, isto é, aqueles através dos quais o indivíduo depressa chega ao estágio da percepção organizada e adquire a grande maioria das experiências úteis à sua educação e cultura. Por outro lado, dado que a criança, o adolescente e a maioria dos adultos denunciam uma maior receptividade visual e que a proporção de imagens visuais é muito maior que a de imagens sonoras, os meios audiovisuais favorecem predominantemente a vista [Lindsay91].

O interesse do homem pelas imagens acompanha-o desde a Pré-História: ‘livros de imagens’ poderiam ser consideradas as paredes das cavernas, as fachadas das igrejas, os seus claustros e

os capitéis. Com a imprensa sobrevém a gravura em madeira, depois a gravura em aço e ainda a litografia. Na época contemporânea, a fotografia, o filme e a televisão iniciam o verdadeiro reinado do audiovisual com absoluta supremacia da imagem que assume hoje, e cada vez mais, importância vital para a retenção da atenção, para a motivação, para a construção de hábitos de carácter visual, como alargamento dos horizontes da vida real ou imaginada e como elemento de aferição de conhecimento.

Eis a imagem, palavra-chave de toda a estrutura organizativa e sequencial que dá origem ao objecto de estudo desta Tese: **o vídeo**. O vídeo é hoje um prolongamento da existência do homem, sendo impossível imaginar o mundo actual sem a utilização desse maravilhoso recurso que se tornou imprescindível mas que continua a suscitar fascínio e deslumbramento.

Perante tal proliferação e banalização da imagem, mentes mais mundanas são levadas a acreditar que nada mais há a descobrir ou a pesquisar e que pouco há, ainda, a inventar, pensar ou organizar. No entanto, muito existe por fazer em todas as áreas da tecnologia audiovisual. Se, por um lado se assistiu, ao longo do séc. XX, a uma proliferação da informação audiovisual nos mais variados serviços e sistemas, por outro, assistiu-se também, ao crescendo das dificuldades técnicas associadas à sua utilização maciça.

1.1 Contexto e Motivação

O aumento vertiginoso da produção de conteúdos audiovisuais tem levado os operadores de televisão e outros produtores de conteúdos audiovisuais a demonstrar interesse na criação de bibliotecas digitais que permitam o arquivo de conteúdos multimédia para posterior reutilização ou para a sua disponibilização *on-line* para serem usados por outras companhias ou pelo público em geral. Neste contexto, a anotação de conteúdos e a indexação de vídeo digital são problemas que assumem alguma importância, uma vez que a quantidade e a necessidade de aceder de forma eficiente a este tipo de informação continua a aumentar. Para satisfazer esta necessidade são necessários sistemas capazes de tratar a informação audiovisual em termos de armazenamento, transmissão, procura e visualização. Em muitas bases de dados de vídeo, a anotação é feita manualmente por humanos: este processo consome demasiado tempo e é demasiado caro. Assim, o desenvolvimento de sistemas, automáticos e eficientes, de anotação, indexação e procura de conteúdos multimédia torna-se importante e, por isso, vários métodos têm vindo a ser desenvolvidos ao longo dos últimos anos. Em muitos destes métodos, privilegia-se a informação textual existente nas imagens e nos vídeos que, por ser uma fonte de informação com elevado valor semântico, torna possível a indexação e procura de conteúdos de forma acessível e intuitiva a produtores e consumidores. Muitas vezes, o texto existente nos vídeos relata o seu propósito ou resume o seu conteúdo: por exemplo, o texto de rodapé nas notícias, os títulos dos vídeos, o nome de oradores, etc. Assim, é razoável usar este texto para construir palavras chave e indexar o vídeo.

A existência de uma norma é absolutamente essencial para a difusão, em larga escala, de qualquer tecnologia onde a noção de interoperabilidade seja importante. Só através do ‘acordo’ consignado pela norma se pode conseguir a interoperabilidade entre sistemas, bem como a convergência de definições e critérios que permitem o avanço tecnológico nos seus mais variados domínios. A tecnologia audiovisual e as aplicações e serviços a ela associados não são, a esse nível, uma excepção. A existência de uma norma mundial para a descrição de conteúdos audiovisuais, qualquer que seja o seu suporte, permite um nível de interoperabilidade entre aplicações e utentes impossível de alcançar de outro modo.

Com o objectivo de contribuir para a solução do problema da descrição de informação audiovisual e, consequentemente para a explosão das aplicações associadas, o grupo *Motion Picture Experts Group* (MPEG) da *International Standards Organization* (ISO), decidiu lançar, em 1996, um projecto denominado *Multimedia Content Description Interface*, mais conhecido como MPEG-7 [Manjunath02]. Este projecto propunha a especificação de um conjunto de ferramentas e métodos que permitissem a descrição de vários tipos de informação como, por exemplo, imagens estáticas, vídeo ou áudio. A descrição da informação deveria ser completamente independente do seu formato – digital ou analógico, ou forma de armazenamento – papel, filme ou cassete. O desenvolvimento desta norma visava permitir que a grande quantidade de conteúdos audiovisuais, presentemente disponíveis, pudesse ser pesquisado, filtrado, gerido e consumido de forma criteriosa, flexível, rápida e eficiente. Por outro lado, a existência de uma norma internacional permitiria que mais conteúdo fosse descrito com o mesmo formato e que surgissem aplicações cada vez mais potentes para esse formato largamente usado.

A norma MPEG-7, como todas as outras normas da família MPEG (MPEG-1, -2, -4), especifica métodos de representação da informação audiovisual de forma a satisfazer um conjunto relevante de requisitos. No MPEG-7, estes requisitos estão relacionados com a identificação e descrição (procura, filtragem, etc.) de conteúdo audiovisual [MPEG7-Req02]. Neste contexto, a informação textual associada à informação audiovisual assume um papel importante na descrição do conteúdo. Alguns exemplos, através dos quais se pode constatar esta importância, prendem-se com a informação associada aos nomes de locais e pessoas, publicidade em eventos e também à anotação individual dos conteúdos. No entanto, as descrições textuais trazem também questões delicadas, nomeadamente associadas à dependência linguística, à sua subjectividade e ao enorme esforço associado à anotação manual de todo o conteúdo que hoje é produzido [MPEG7-Req02]. Por outro lado, como todas as ‘boas’ normas, a norma MPEG-7 especifica apenas o ‘essencial’ ou seja neste caso o formato das descrições e não a forma como estas são criadas ou consumidas já que essas fases do processamento, tão importantes para o desempenho dos sistemas, não requerem especificação normativa em termos de interoperabilidade.

1.2 Objectivos

Infelizmente, a tecnologia actual disponibilizada no mercado e utilizada pelos sistemas de reconhecimento óptico de caracteres, em Inglês *Optical Character Recognition* (OCR), apresenta dificuldades no reconhecimento do texto existente nos vídeos, em virtude de este, tipicamente, possuir baixa resolução e surgir sobre fundos complexos. Para além disso, o texto pode aparecer em cada trama em vários locais, orientações, fontes, tamanhos e cores.

De forma a superar as dificuldades evidenciadas pelos sistemas OCR mais tipicamente disponíveis para reconhecimento de texto em sequências de vídeo e imagens a cores, têm sido desenvolvidos, nos últimos anos, vários métodos capazes de efectuar a extracção de texto em condições menos específicas e limitadas, de modo a tornar o processamento deste tipo de conteúdo mais acessível aos seus produtores e consumidores.

Os métodos desenvolvidos resolvem, em certas condições, o problema da extracção de texto em sequências de vídeo ou imagens, mas pode haver ainda algumas limitações. As maiores dificuldades advêm da existência de:

- Caracteres com diferentes tamanhos, orientações e perspectivas;
- Caracteres com diferentes cores na mesma linha ou palavra;
- Diferente espaçamento entre os caracteres na mesma linha, o que dificulta o seu agrupamento em palavras;
- Fraco contraste em relação ao fundo, especialmente quando este é de textura variada;
- Caracteres correspondentes a vários alfabetos.

Depois de diagnosticadas as principais limitações e dificuldades na extracção de texto em imagens e vídeo, torna-se fundamental o desenvolvimento de técnicas de processamento capazes de as superar, se não na totalidade pelo menos em parte. Neste contexto, os objectivos a atingir com a presente Tese são:

1. Revisão bibliográfica das principais técnicas disponíveis para a extracção automática de texto em imagens e vídeo, nomeadamente técnicas de segmentação, classificação, seguimento e reconhecimento;
2. Desenvolvimento de um mecanismo de extracção automática de texto em imagens sem limitações significativas em termos de conteúdo e logo superando algumas das limitações atrás identificadas;
3. Extensão do mecanismo desenvolvido para imagens para a extracção automática de texto em sequências de vídeo;
4. Implementação em *software* dos mecanismos desenvolvidos para a extracção de texto em imagens e sequências de vídeo;
5. Avaliação do desempenho, para vários tipos de conteúdo, dos mecanismos desenvolvidos e implementados para a extracção de texto em imagens e sequências de vídeo.

O texto extraído pode ser utilizado para os mais diversos fins como se poderá constatar mais adiante, nesta Tese.

1.3 Conceitos e Terminologia Relevantes

Para facilitar a leitura desta Tese e evitar inconsistências, utilizar-se-á sempre que possível a terminologia usada pela norma MPEG-7 em termos de análise e descrição de informação audiovisual [MPEG7-Req02]. A extracção e descrição do texto existente na informação audiovisual envolve os seguintes conceitos principais (entre parênteses indica-se o termo usado pelo MPEG-7):

- **Informação audiovisual (*data*)** – Conteúdo audiovisual que vai ser sujeito à extracção de texto, independentemente da sua forma de armazenamento, codificação, visualização e transmissão; por exemplo, um fluxo binário MPEG-4, uma imagem JPEG ou um filme numa cassete de vídeo;

- **Região** – Conjunto conexo de todos os *pixels* numa imagem com a mesma etiqueta de identificação; exemplos de regiões podem ser visualizados na Figura 1.1;
- **Objecto** – Região ou conjunto de regiões com um significado especial, nomeadamente semântico, exemplos de objectos simples constituídos por uma única região e de objectos complexos constituídos por várias regiões podem ser visualizados na Figura 1.1;

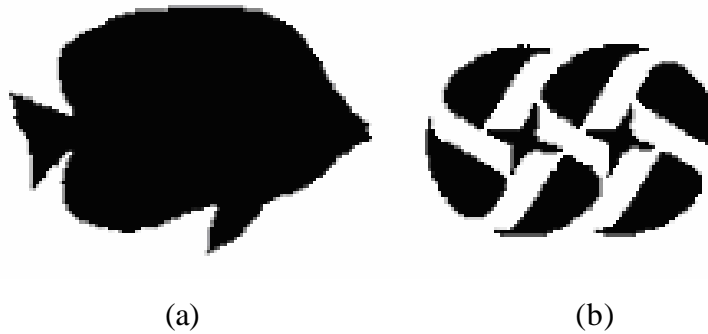


Figura 1.1 - Exemplos de objectos: (a) objecto simples constituído por uma única região; (b) objecto complexo constituído por mais do que uma região [MPEG7-Visual01].

- **Contorno** – Conjunto de todos os *pixels* que pertencem a uma região ou objecto e que têm como vizinhos, segundo um dado tipo de vizinhança, pelo menos um *pixel* não pertencente a essa região ou objecto; o contorno define a forma de uma região ou objecto. Um exemplo de contorno pode ser visualizado na Figura 1.2 (b);

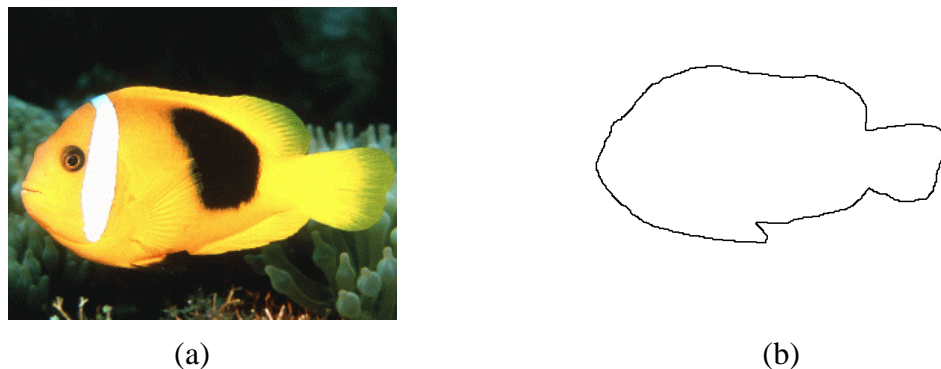


Figura 1.2 – Exemplo de contorno: (a) imagem com um objecto principal; (b) contorno do objecto principal em (a) [MPEG7-Visual01].

- **Shapel** – Elemento básico do suporte de um objecto ou região correspondendo aos *pixels* onde a textura do objecto tem valor não nulo;
- **Bounding box** – Representação grosseira da forma do objecto através do menor rectângulo que engloba completamente o objecto; o conceito de *bounding box* é ilustrado na Figura 1.3;

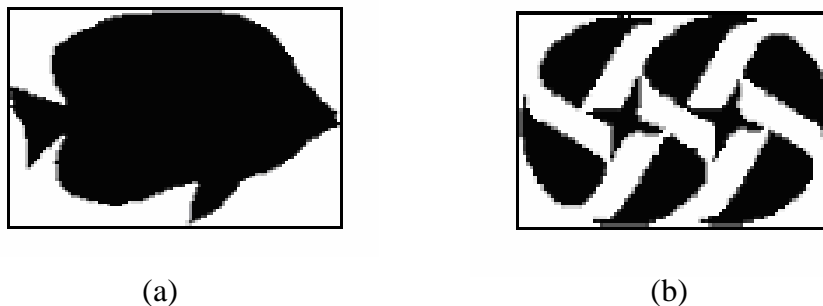


Figura 1.3 – Exemplos de *bounding boxes*: (a) forma de objecto simples e *bounding box* correspondente; (b) forma de objecto complexo e *bounding box* correspondente [MPEG7-Visual01].

- **Partição** – Conjunto de regiões disjuntas e que combinadas compõem a imagem na sua totalidade; tipicamente, a partição de uma imagem é criada unicamente com base nas características espaciais dessa imagem; em termos de vídeo, uma partição é criada tendo em conta tanto as características espaciais como as suas características temporais;
- **Característica (*feature*)** – Qualquer propriedade relevante da informação audiovisual e que pode significar algo para alguém; por exemplo, o título de um filme, o nome de um actor, a cor ou a textura de uma imagem, o timbre da voz;
- **Parâmetro** – Representação de uma dada característica usando uma técnica de análise; um parâmetro define apenas a semântica da representação da característica, por exemplo, a área de uma região, a circularidade de um contorno, o histograma de cor de uma imagem;
- **Descritor (*descriptor*)** – Representação completa de uma dada característica; um descritor define a sintaxe e a semântica da representação de cada característica, i.e. acrescenta ao parâmetro a definição da sintaxe da representação como, por exemplo, os descritores normalizados pela norma MPEG-7 [MPEG7-Visual01];
- **Valor do descritor** – Instanciação de cada um dos campos de um descritor que pode ser composto por um ou mais campos sintácticos para representar a característica audiovisual em questão;
- **Esquema de descrição (*description scheme*)** – Especifica a estrutura semântica das relações existentes entre os seus componentes que podem ser tanto descritores como esquemas de descrição; por exemplo, um filme temporalmente estruturado em cenas com descritores associados ao filme, tais como o nome do filme e do realizador e descritores de cor, movimento e áudio associados a cada cena;
- **Descrição (*description*)** – Consiste num esquema de descrição (que pode ser a combinação de vários outros) e no conjunto de valores dos descritores associados que descrevem a informação audiovisual em questão.

Ao longo desta Tese falar-se-á, inúmeras vezes, em vídeo e imagens, bem como em texto gráfico e texto de cena. Com o intuito de facilitar a compreensão desta Tese, torna-se conveniente definir estes conceitos e por vezes salientar diferenças. Da mesma forma, é importante a definição de alguns termos que serão, ao longo da mesma, muitas vezes referenciados e que, por não serem utilizados no seu mais comum significado ou não existir para eles qualquer definição de consenso geral, merecem algum tipo de esclarecimento.

Neste contexto, começar-se-á por salientar as diferenças existentes no processamento de vídeo e imagens com vista à extracção de texto. Assim, pode afirmar-se que as imagens não são mais do que um caso particular de vídeo, resumindo-se a sua diferença à inexistência de redundância temporal nas primeiras. A redundância temporal existente no vídeo releva-se no facto de cada linha de texto aparecer na mesma posição ou em posição próxima, ao longo de várias tramas sucessivas. Esta redundância temporal pode ser explorada para:

- Aumentar a probabilidade de detecção de texto, desde que este surja dentro de determinadas condições de trama para trama;
- Remover detecções falsas em tramas individuais, se as detecções não se mantiverem consistentes ao longo do tempo;
- Fazer a interpolação de linhas de texto que não foram ‘acidentalmente’ detectadas em tramas individuais.

Desta forma, todas as técnicas desenvolvidas para extracção de texto em imagens podem ser aplicadas ao vídeo, encarado como uma sequência de imagens; o contrário, como é óbvio, não se verifica necessariamente uma vez que as imagens são tramas isoladas.

A informação textual existente nas imagens e nos vídeos é uma fonte de informação com um elevado nível semântico em termos de pesquisa, desde que possa estar disponível como texto. Na verdade, em termos de extracção automática, o texto é, com toda certeza, o tipo de informação com maior valor semântico. As outras características automaticamente extraíveis como a cor, o movimento, a textura, etc. possuem muito menor valor semântico e, quando o possuem, este é, por vezes, bastante difícil de extrair. O texto existente nas tramas de vídeo ou nas imagens pode ser classificado como texto gráfico ou como texto de cena:

- **Texto gráfico** – Texto que é adicionado, automaticamente ou sinteticamente às tramas ou imagens, normalmente através de computador, para lhes juntar informação com o objectivo de complementar o conteúdo das mesmas. Este tipo de texto é, usualmente, mais estruturado e apresenta melhor contraste em relação ao restante conteúdo, uma vez que é adicionado de forma controlada;
- **Texto de cena** – Texto que é directamente capturado pelas câmaras de filmar e que faz parte das próprias cenas filmadas. Exemplos de texto de cena são os nomes das ruas nas placas, texto escrito em placares publicitários, nos carros e nas camisolas em eventos desportivos.

Na Figura 1.4 podem ser visualizados exemplos de texto de cena na imagem (a) e de texto gráfico na imagem (b).



Figura 1.4 – Exemplos dos dois tipos de texto: (a) texto de cena; (b) texto gráfico.

O texto de cena é, na maioria dos casos, mais difícil de detectar e extrair uma vez que pode possuir um número praticamente ilimitado de perspectivas, tamanhos, formas, cores e posições dentro da imagem ou trama de vídeo. É, no entanto, muito importante em aplicações como navegação, vigilância, classificação de vídeo ou análise de eventos desportivos.

1.4 Organização da Tese

Em qualquer documento, a existência de uma sequência lógica de evolução do texto é fundamental. A importância a dar ao rigor científico dos conceitos só tem paralelo na importância a atribuir à forma como os mesmos aparecem ao longo de todo o texto. Assim, esta Tese está organizada em 5 capítulos com o seguinte conteúdo:

- **Capítulo 1: Introdução** – Neste capítulo é feita uma contextualização e motivação do tema a tratar – extracção automática de texto em sequências de vídeo –, são definidos os objectivos a atingir, são definidos alguns conceitos e terminologia mais relevantes e, por último, é apresentada a organização da presente Tese;
- **Capítulo 2: Extracção de texto em imagens e sequências de vídeo: revisão bibliográfica** – Neste capítulo propõe-se uma arquitectura básica para a extracção de texto em imagens e sequências de vídeo. Para além disso, é feita uma revisão bibliográfica das várias técnicas utilizadas na extracção de texto em imagens e sequências de vídeo, nomeadamente técnicas de segmentação, classificação, seguimento e reconhecimento. São também descritos alguns dos sistemas de extracção de texto mais representativos e eficientes de entre aqueles actualmente disponíveis;
- **Capítulo 3: Extracção de texto em imagens** – Neste capítulo descreve-se, de forma pormenorizada, o algoritmo proposto no âmbito desta Tese para a extracção de texto em imagens, as quais podem conter tanto texto gráfico como texto de cena. Neste capítulo, a atenção centrou-se sobretudo no desenvolvimento de um algoritmo robusto capaz de detectar também texto inclinado, tanto de cena como gráfico. Quer o texto

gráfico inclinado, quer o texto de cena escrito em qualquer direcção, têm sido objecto de pouco investimento por parte de outros investigadores. Assim, neste capítulo foram aperfeiçoadas técnicas já existentes de segmentação e análise de contraste para as tornar mais eficazes na extracção de texto em imagens onde o texto é pouco contrastado e as suas fronteiras mal definidas, i.e. texto tipicamente de cena. Foram também desenvolvidas técnicas que possibilitam agrupar os vários caracteres, detectados em palavras escritas em qualquer direcção, com um número de falsos agrupamentos baixo. A avaliação de desempenho foi efectuada, quer usando um sistema OCR comercial, quer usando um OCR desenvolvido por Lienhart [Lienhart95];

- **Capítulo 4: Extracção de texto em sequências de vídeo** – Neste capítulo descreve-se, de forma pormenorizada, o algoritmo proposto no âmbito desta Tese para a extracção de texto em sequências de vídeo. Assim, adiciona-se neste capítulo ao mecanismo desenvolvido para imagens a componente temporal. Para tal, foram desenvolvidas técnicas de seguimento que utilizam duas tramas de cada vez para efectuar o seguimento do texto, i.e. relacionam o resultado da extracção para a trama anterior com o resultado para a trama actual. Esta relação é conseguida através da definição de uma assinatura para o texto detectado, formada pela suas características tais como cor, tamanho, deslocamento, etc. Foram, também, desenvolvidas técnicas de recuperação de texto sobre o qual a detecção falhou em tramas individuais recorrendo para tal à interpolação do texto em falta. De forma semelhante ao que foi efectuado no capítulo anterior, também aqui foi avaliado o desempenho do algoritmo proposto com dois sistemas OCR;
- **Capítulo 5: Comentários finais** – Para finalizar, tecem-se neste capítulo considerações finais e conclusões sobre os desafios inerentes à extracção de texto em imagens e sequências de vídeo. Com base nas considerações e conclusões finais, identificar-se-ão tópicos de interesse com vista a possível trabalho futuro.

Na presente Tese foi dado ênfase às áreas menos desenvolvidas por outros investigadores, tais como a extracção de texto de cena e a extracção de texto escrito em qualquer direcção. Os algoritmos resultantes permitem uma análise bastante robusta do texto de uma imagem ou vídeo contribuindo assim de forma importante para a descrição com elevado valor semântico deste tipo de dados.

Capítulo 2

Extracção de Texto em Imagens e Vídeo: Revisão Bibliográfica

A maior facilidade em adquirir, processar, armazenar e transmitir informação audiovisual veio acentuar a necessidade de desenvolver ferramentas para fazer o processamento dessa informação, com vários objectivos. Para além disso, o facto dos conteúdos criados terem começado a ser enriquecidos com componentes multimédia mais complexos – imagens, vídeo e áudio – originou o aumento substancial da capacidade das bibliotecas para os armazenar. Assim, surge a necessidade da existência de sistemas, automáticos e eficientes, de descrição, indexação e procura de conteúdos multimédia. Neste contexto, a informação textual existente nas imagens e nos vídeos é uma fonte de informação com um elevado nível semântico em termos de pesquisa, desde que esse mesmo texto esteja disponível como texto. Para isso, o texto deve ser detectado, segmentado e reconhecido automaticamente de modo a que possa ser utilizado para indexação e procura nas bibliotecas de imagem e vídeo como, por exemplo, na localização de cenas, procura de eventos, nomes de produtos, nomes de oradores, anúncios, etc.

Assim, de modo a usar a informação textual existente nos vídeos, são necessários sistemas capazes de detectar e segmentar de forma automática o texto aí existente antes de se proceder ao reconhecimento dos caracteres. Na Figura 2.1 é ilustrada uma possível aplicação para a extracção de texto em imagens ou vídeos.

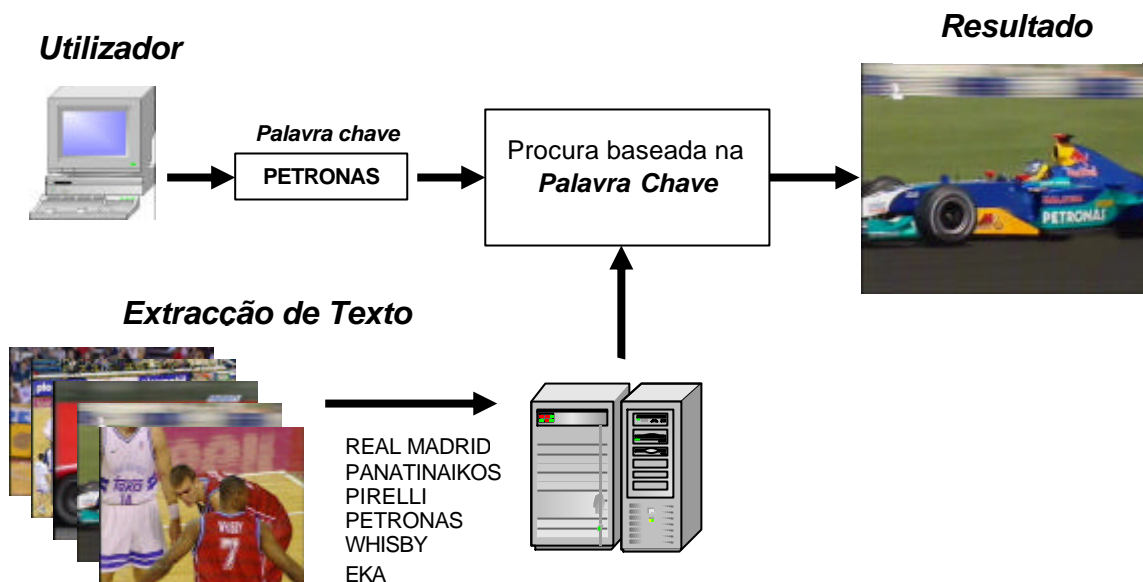


Figura 2.1 – Exemplo de procura baseada na extracção de texto em imagens ou vídeos.

Antes de iniciar a revisão bibliográfica das principais técnicas relevantes para a extracção de texto em imagem e vídeo, é fundamental estabelecer uma arquitectura básica para este tipo de processamento. Por arquitectura básica, entende-se o conjunto das várias fases de processamento e das suas interligações através das quais qualquer imagem ou sequência de vídeo deve passar com o objectivo de se extrair de forma automática o texto nelas contido.

2.1 Arquitectura Básica

No contexto desta Tese, sugere-se como arquitectura básica para um sistema de extracção de texto em sequências de vídeo aquela que está apresentada na Figura 2.2. A extracção de texto em imagens não é mais do que um caso particular da extracção de texto em vídeo, i.e. a extracção de texto numa imagem é equivalente à extracção de texto numa trama individual de vídeo em que não é explorada a redundância temporal entre tramas.

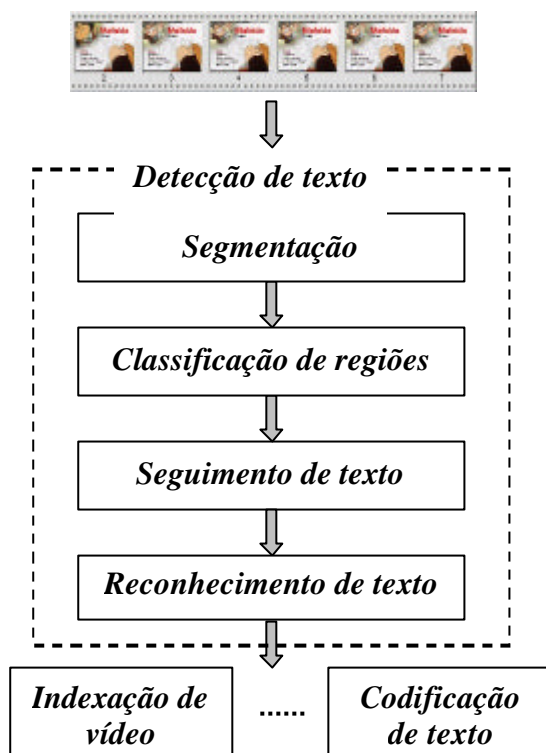


Figura 2.2 – Arquitectura básica para a extracção de texto em sequências de vídeo.

Na Figura 2.2 pode observar-se que a extracção de texto em vídeo decorre em duas fases principais distintas, nomeadamente detecção e reconhecimento do texto:

- **Detecção de texto** – Esta fase visa a detecção do texto existente nas imagens ou tramas de vídeo e pode ser dividida em três passos distintos: segmentação da imagem ou tramas de vídeo, classificação das regiões segmentadas como texto ou não texto e seguimento ao longo do tempo das regiões classificadas como texto:
 - ♦ **Segmentação** – Esta fase visa a divisão completa das tramas de vídeo ou imagens em regiões homogéneas segundo um dado critério: a cada uma destas regiões podem corresponder ou não caracteres textuais. O processo de segmentação pode começar com o processamento das tramas de vídeo ou imagens com o intuito de diminuir a influência de alguns efeitos indesejáveis, tais como diferentes gradientes de luminosidade, ruído ou elevado número de cores. Posteriormente, é efectuada a segmentação propriamente dita de modo a criar regiões homogéneas, segundo um ou mais critérios. Para isso, podem ser utilizadas uma ou mais técnicas de segmentação, das muitas actualmente disponíveis. Exemplos de técnicas de segmentação de sequências de vídeo ou imagens para posterior extracção de texto podem ser encontrados em [Zhong95, Messelodi99, Lienhart00];
 - ♦ **Classificação de regiões** – Esta fase da detecção de texto visa a classificação de cada uma das regiões provenientes da fase de segmentação como texto ou não texto; para além disso, visa o agrupamento do primeiro tipo de regiões de modo a formar

palavras e linhas. Para tal, é efectuada a classificação de cada uma das regiões provenientes da fase anterior, para assim se determinar quais as regiões que correspondem a caracteres de texto. Para este efeito, as técnicas mais utilizadas, por se revelarem as mais eficazes, são as que se baseiam na análise geométrica das regiões [Zhong95, Messelodi99, Lienhart00] e as que utilizam redes neuronais [Li00, Li02, Lienhart02]. As regiões que forem classificadas como não texto são descartadas. Seguidamente, as várias regiões classificadas como prováveis caracteres são agrupadas de modo a formarem conjuntos de regiões que representam palavras ou mesmo linhas de texto;

- ♦ **Seguimento de texto** – Nesta fase, é efectuada o refinamento da classificação efectuada na fase anterior. Para tal, é efectuada o seguimento das regiões classificadas anteriormente como texto, explorando assim a redundância temporal existente no vídeo. A exploração da redundância temporal permite aumentar a probabilidade da detecção de texto, efectuar a remoção de falsas detecções em tramas individuais (eliminação de falsos positivos), bem como determinar com exactidão o início e o fim da ocorrência de cada sequência de texto no vídeo. Os métodos desenvolvidos para a detecção de texto em imagens não incluem esta fase uma vez que nas imagens não existe redundância temporal pois as imagens são tramas isoladas. Exemplos de métodos que exploram a redundância temporal para refinar a detecção do texto existente no vídeo podem ser encontrados em [Li00, Lienhart00, Li02, Lienhart02, Wolf02].
- **Reconhecimento de texto** – Nesta fase, é efectuada o reconhecimento do texto existente nas imagens ou tramas de vídeo, usando as regiões candidatas determinadas na fase anterior. Para esse efeito, utiliza-se um sistema OCR, que tanto pode ser um sistema comercial [Zhong95, Wu99, Li02, Lienhart02, Wolf02], como uma implementação desenvolvida para o caso específico em questão [Lienhart95, Sato99].

A Figura 2.3 ilustra, de forma simplificada, o resultado de cada uma das fases do processo de extracção de texto para uma trama de vídeo ou imagem.



(a)



(b)



(c)



(d)

SCHNITTASSISTENZ
SABINE BROSE
SPEZIALEFFEKTE
MICHAEL BOUTERWECK
ALAN STUART
KOSTUMASSISTENZ

(e)

Figura 2.3 – Exemplo da extracção de texto para uma trama de vídeo: (a) imagem original; (b) imagem segmentada; (c) imagem com as regiões classificadas como texto; (d) imagem depois do refinamento da detecção através da exploração da redundância temporal; (e) texto resultante da aplicação de um sistema OCR à imagem resultante da fase de seguimento [Lienhart00].

O texto resultante do processo de extracção pode ter as mais variadas aplicações, nomeadamente acrescentar uma componente semântica à descrição do vídeo correspondente usando, eventualmente, os descritores adequados da norma MPEG-7. A componente semântica acrescentada pode revelar-se útil, por exemplo, para:

- **Navegação automática** – A componente semântica detectada, por exemplo, no processamento de imagens com sinalização e toponímia, pode revelar-se útil ao pretender validar-se determinado percurso rodoviário, através da sua comparação com a informação existente numa carta electrónica;
- **Vigilância** – A componente semântica detectada pode revelar-se fundamental quando do seguimento da trajectória de determinada viatura através da extracção e verificação da sua matrícula;
- **Indexação** – A componente semântica detectada pode ser utilizada como palavra chave na indexação e classificação de vídeos, na análise de eventos, etc.;
- **Codificação** – A componente semântica detectada pode ainda ser utilizada para fazer a codificação eficiente do texto como um objecto textual independente; este tipo de codificação baseada em objectos foi adoptada pela norma MPEG-4 [Pereira02].

Nas secções seguintes será efectuada uma revisão bibliográfica das principais técnicas relevantes para a extracção automática de texto em imagens e vídeo, nomeadamente técnicas de segmentação, classificação, seguimento e reconhecimento. Para além disso, serão apresentados alguns sistemas completos de extracção de texto em imagens e sequências de vídeo, escolhidos devido à sua relevância.

2.2 Técnicas de Segmentação de Imagem e Vídeo

A segmentação é o primeiro e um dos mais importantes objectivos na extracção de texto em imagens e sequências de vídeo. Contudo, não existe uma teoria completa ou uma solução final para a segmentação que cubra todos os casos, estando o desenvolvimento das várias técnicas

existentes fortemente dependente das propriedades associadas às regiões desejadas. Pavlidis refere sobre a segmentação de vídeo que *“o problema é basicamente de percepção psicofísica não sendo, por isso, susceptível de uma solução puramente analítica. Quaisquer algoritmos matemáticos necessitam de ser complementados por heurísticas, normalmente envolvendo semânticas sobre a classe de imagens em questão”* [Pavlidis77]. Já segundo Haralick e Shapiro, a segmentação de uma imagem pode ser definida como *“o processo que tipicamente divide o domínio espacial de uma imagem em subconjuntos mutuamente exclusivos, denominados regiões, sendo cada uma delas uniforme e homogénea em relação a uma determinada propriedade como, por exemplo, tom, cor, contraste ou textura e cujo valor dessa propriedade difere significativamente entre regiões vizinhas”* [Haralick94]. Para estender esta definição à extracção de texto em sequências de vídeo, torna-se necessário ter em conta a dimensão temporal. A análise temporal é introduzida através da estimação do movimento existente entre tramas consecutivas, obtendo-se assim informação importante para a formação de regiões que não são espacialmente homogéneas mas que podem pertencer a um mesmo objecto.

Dependendo dos objectivos e critérios em causa, um grande número de técnicas para segmentação automática de vídeo tem sido proposto na literatura. Bons exemplos dessas técnicas podem ser encontrados em [Aach93, Cortez95, Marques96, Raghu96, Pavlidis90, Moghaddamzadeh97, Salembier99, Fan01]. Estas técnicas podem ser agrupadas em três grandes categorias em função dos critérios de homogeneidade adoptados para as regiões pretendidas [Correia02]:

- **Segmentação espacial** – As regiões pretendidas deverão ser homogéneas em termos das suas características espaciais. Os critérios de homogeneidade podem estar associados ao contraste, média ou direcionalidade da luminância e crominâncias. Vários tipos de segmentação espacial podem ser considerados dependendo da aplicação em causa. Assim, as técnicas de segmentação espacial podem ser divididas nas seguintes classes:
 - ♦ **Baseadas na amplitude** – Técnicas simples que identificam as várias regiões com base na análise dos histogramas da luminância e crominâncias [Haralick92];
 - ♦ **Baseadas na textura** – Técnicas que segmentam as regiões com base nas suas características em termos de textura; estas técnicas são, geralmente, bastante eficientes na detecção de regiões com uma grande variedade de texturas [Nunes95];
 - ♦ **Baseadas em fronteiras** – Técnicas que detectam primeiramente as fronteiras existentes na imagem e depois processam o resultado de forma a identificar as várias regiões [Jain89, Pratt91];
 - ♦ **Baseadas em regiões** – Técnicas que detectam regiões homogéneas na imagem, separadas por fronteiras bem definidas [Haralick92, Cortez95].
- **Segmentação temporal** – As regiões pretendidas deverão ser homogéneas em termos das suas características de movimento na sequência de vídeo. Estas técnicas operam, usualmente, sobre vectores de movimento estimados, conseguindo-se assim produzir regiões coerentes no tempo; no entanto, têm o inconveniente de não conseguir identificar objectos fixos. Dois tipos de técnicas podem ser considerados, dependendo das aplicações:
 - ♦ **Detecção de alterações** – Técnicas que se baseiam na identificação das áreas que se alteram (ou não) ao longo das sucessivas tramas do vídeo [Hötter88, Musmann89];

- ♦ **Segmentação de movimento** – Técnicas que se baseiam em critérios de homogeneidade do movimento como, por exemplo, a direcção e velocidade dos campos de vectores de movimento; neste caso, objectos com movimentos diferentes podem ser identificados ainda que possuam uma textura semelhante [Wu93, Wang94].
- **Combinação de segmentação espacial e temporal** – As regiões pretendidas deverão ser homogéneas em ambas as dimensões: espacial (textura) e temporal (movimento). Existem vários tipos de técnicas de segmentação combinando a informação espacial e temporal, dependendo da forma como essa informação espacial e temporal é processada. Três tipos de técnicas podem ser consideradas, dependendo das aplicações:
 - ♦ **Temporal depois da espacial** – Técnicas em que a segmentação temporal é efectuada tendo em conta, também, a informação espacial já disponível; por exemplo, um objecto pode ser formado por várias regiões (espacialmente detectadas) se estas partilharem as mesmas características de movimento [Choi97];
 - ♦ **Espacial depois da temporal** – Técnicas que aperfeiçoam o resultado da segmentação temporal, utilizando a segmentação espacial como, por exemplo, para obter fronteiras com maior precisão [Mech98, Kim99];
 - ♦ **Temporal e espacial em simultâneo** – Técnicas em que a segmentação é efectuada considerando, simultaneamente, a informação temporal e espacial [Salembier94].

As várias classes de técnicas de segmentação de vídeo aqui identificadas serão discutidas nas secções seguintes. Como é evidente, a segmentação de imagens considera apenas técnicas de segmentação espacial.

2.2.1 Segmentação Espacial

Na segmentação espacial, cada imagem é considerada isoladamente mesmo que faça parte de uma sequência de vídeo. A partição da imagem, ou seja, o conjunto de regiões disjuntas que compõem (completamente) a imagem é criada unicamente com base nas características espaciais de cada imagem. Consequentemente, a informação temporal existente no vídeo não é considerada, significando isso a impossibilidade de gerar regiões coerentes no tempo, no caso de se utilizar apenas este tipo de técnicas.

2.2.1.1 Segmentação Espacial Baseada na Amplitude

As técnicas de segmentação baseadas na amplitude encontram-se entre as mais importantes técnicas de segmentação espacial. Na segmentação baseada na amplitude procura definir-se um conjunto de níveis de separação ou limiares que permitam identificar, no histograma de amplitudes de uma imagem, zonas com diferentes propriedades. Assim, os *pixels* da imagem são agrupados em regiões em função dos valores determinados para os níveis de separação.

A segmentação de amplitude é uma técnica com grande utilização pois oferece uma abordagem muito simples ao problema da segmentação e produz bons resultados, sobretudo

para imagens em tons de cinzento onde a principal característica é o nível de cinzento (ou luminância) dos *pixels*.

A abordagem mais básica consiste na segmentação da imagem com base num limiar para o valor da luminância. Esta técnica, quando aplicada a imagens simples, depois de cuidadosamente seleccionado o valor de limiar, resulta numa boa separação dos objectos com um brilho elevado e uniforme em relação ao fundo. A Figura 2.4 ilustra um exemplo da segmentação baseada na amplitude de uma imagem em tons de cinzento, utilizando um único limiar.

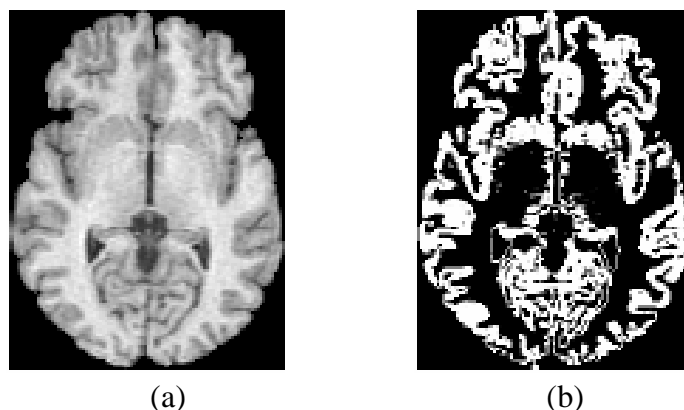


Figura 2.4 – Exemplo de segmentação baseada na amplitude: (a) imagem original; (b) segmentação com um único limiar da imagem em (a) [Pham02].

O conceito de segmentação por limiar pode ser generalizado para vários limiares, visando a detecção de regiões que se diferenciam através dos seus níveis de luminância. Este tipo de segmentação usando múltiplos limiares apresenta um maior grau de dificuldade de implementação do que as técnicas de um único limiar. A razão desta dificuldade prende-se com a necessidade de estabelecer vários (bons) limiares para detectar as regiões pretendidas.

Quando as imagens são a cores, a sua segmentação é uma extensão multi-modo do conceito de segmentação baseada na amplitude. A aplicação desta técnica a imagens a cores obriga à criação de um histograma para cada componente de cor, bem como à sua análise multi-espectral através do exame das várias componentes espectrais. Os resultados obtidos para cada componente espectral, para uma dada área da imagem, podem ser combinados utilizando técnicas de segmentação baseadas em *clustering*. Isto significa fazer a análise das várias componentes/modos para cada *pixel*, agrupando-os numa mesma região de acordo com os seus valores de luminância e cromaticidade[Gonzalez93, Liu94].

As técnicas baseadas na amplitude apresentam como principais vantagens o seu baixo custo computacional e a sua eficácia em segmentar objectos que sejam distintos dos restantes em termos de alguma das suas componentes espectrais. Como principais desvantagens, há a salientar o número elevado de pequenas regiões, e tipicamente desconhecido, que normalmente resultam da segmentação de imagens texturadas (fenómeno denominado por sobresegmentação), bem como o facto de não ser explorada a relação espacial existente entre *pixels* vizinhos.

2.2.1.2 Segmentação Espacial Baseada na Textura

As técnicas de segmentação baseadas na textura detectam regiões com características homogêneas em termos de textura, sendo de salientar a sua eficácia na detecção de regiões com uma diversidade de texturas elevada ainda que tenham a mesma luminância e crominâncias médias.

A noção de textura, apesar de poder ser identificada em praticamente todos os tipos de imagens e, em particular, em imagens naturais, não tem uma definição precisa, universalmente aceite pela comunidade científica. A dificuldade em elaborar uma definição de textura suficientemente genérica resulta, em parte, do elevado número de atributos que seria necessário incluir numa definição desse tipo.

Uma definição de textura foi proposta por Gagalowicz e Ma “*Se se mover uma janela sobre uma textura e se efectuarem medidas texturais nessa janela, os resultados dessas medidas devem ser invariantes*” [Gagalowicz85]. Esta definição remete para uma outra questão importante, ou seja, a noção de resolução da textura que pode ser definida como o tamanho mínimo da janela que permite obter medidas texturais (conjunto de estatísticas locais ou outras propriedades locais) invariantes. Exemplos de texturas e da aplicação da segmentação de textura são ilustrados na Figura 2.5.

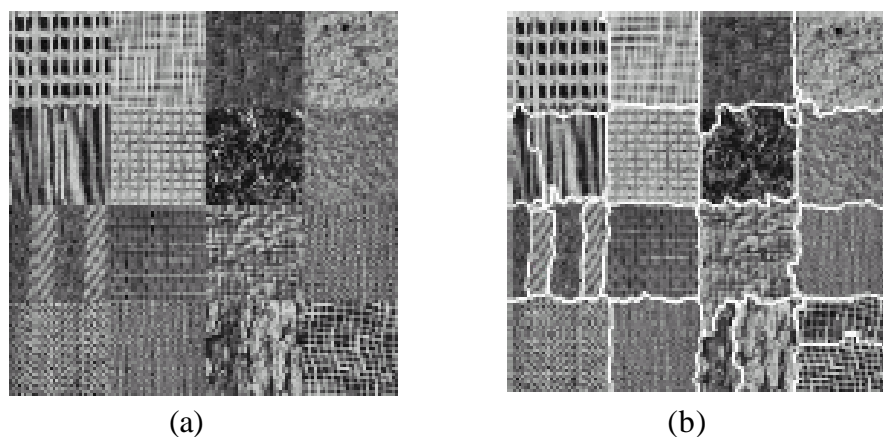


Figura 2.5 – Exemplo de segmentação espacial baseada na textura: (a) imagem original constituída por vários tipos de textura; (b) regiões correspondentes à segmentação da imagem em (a) [Liu02].

Dois tipos principais de texturas podem ser consideradas [Nunes95]:

- **Texturas aleatórias** – Texturas típicas de algumas imagens de superfícies naturais. De um modo geral, não apresentam descontinuidades bem definidas, antes pelo contrário apresentam um aspecto desordenado mas homogêneo. Na Figura 2.6 (a), pode observar-se um exemplo de uma textura aleatória;
- **Texturas determinísticas** – Texturas que se caracterizam por uma estrutura onde é possível identificar padrões elementares que se repetem no espaço da imagem em várias direcções, de um modo mais ou menos regular. Um exemplo de uma textura determinística pode ser observado na Figura 2.6 (b).

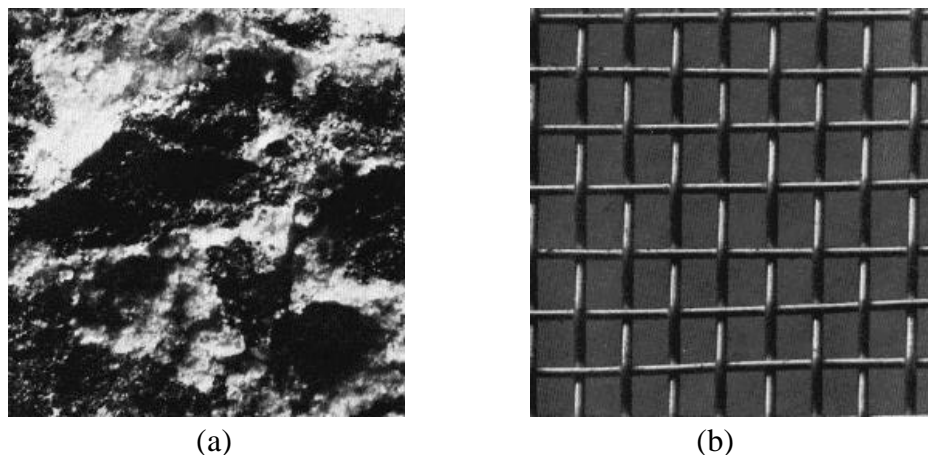


Figura 2.6 – Exemplos de texturas: (a) textura aleatória; (b) textura determinística [MPEG7-Visual01].

Esta classificação não implica que todas as texturas sejam puramente aleatórias ou puramente determinísticas. Conforme o tipo de textura, existem abordagens mais ou menos adequadas para as descrever, nomeadamente [Nunes95]:

- **Empíricas** – As descrições empíricas da textura fazem uso de um conjunto de características relevantes para o sistema visual humano como sejam a uniformidade, regularidade, suavidade, densidade, grossura, granularidade, aspereza, contraste e direccionalidade;
- **Estatísticas** – As descrições da textura em termos estatísticos são particularmente apropriadas para texturas aleatórias. Nesta abordagem, os métodos mais simples baseiam-se na estatística dos níveis de luminância, utilizando como medida os momentos do histograma de luminância da imagem que são medidas estatísticas dos níveis de luminância da imagem ou das suas regiões. No entanto, estas medidas apresentam limitações uma vez que não têm em conta a correlação espacial existente entre os *pixels*. Para superar esta limitação, é frequente usar as medidas das matrizes de co-ocorrência espacial dos níveis de luminância em vez dos momentos do histograma dos níveis de luminância. A descrição das texturas em termos estatísticos inclui medidas tais como:
 - ♦ Medidas do histograma de primeira e de segunda ordem;
 - ♦ Medidas da função de auto-correlação;
 - ♦ Medidas de transformadas das texturas.
- **Estruturais** – As descrições da textura em termos estruturais são particularmente apropriadas para texturas determinísticas. Neste tipo de abordagem, a textura é descrita em termos de primitivas simples e de regras que determinam os possíveis arranjos dessas primitivas. Na abordagem estrutural são usadas primitivas simples para formar padrões mais complexos através da aplicação de regras que condicionam os possíveis arranjos das mesmas. As primitivas texturais podem, por exemplo, ser descritas em termos de:
 - ♦ Nível de luminância;

- ◆ Forma;
- ◆ Homogeneidade de propriedades locais como tamanho, orientação e histograma de segunda ordem, i.e. co-ocorrência de primitivas.

As regras de disposição determinística podem ser definidas em termos de:

- ◆ Adjacência;
- ◆ Proximidade;
- ◆ Periodicidade.

As regras de disposição aleatória podem ser definidas em termos de:

- ◆ Densidade de arestas;
 - ◆ Densidade de extremos relativos.
- **Espectrais** – As descrições da textura em termos espectrais permitem a descrição de texturas com periodicidade e direccionalidade através da exploração das propriedades da densidade espectral de potência das imagens. Na abordagem espectral são usadas características do espectro de Fourier para detectar a periodicidade global. Este serve para descrever a direccionalidade de padrões periódicos ou quase periódicos, uma vez que esta característica está associada a picos de energia no espectro;
 - **Modelos generativos** – As descrições da textura com base em modelos generativos consistem na utilização de modelos de textura com determinados valores dos seus parâmetros para sintetizar texturas que estejam de acordo com os referidos modelos e com os parâmetros estimados.

Para utilizar computacionalmente o conceito de textura, é necessário caracterizá-lo matematicamente o que requer a identificação de atributos ou características apropriadas para as diferentes texturas da imagem a segmentar. Estas técnicas de segmentação podem ser baseadas em métodos de optimização ou modelos probabilísticos como sejam os campos aleatórios de Markov e a estimação Bayesiana para classificação [Nunes95].

A segmentação baseada na textura apresenta como grande vantagem a sua capacidade para detectar homogeneidades mais sofisticadas, ainda que as regiões da imagem possuam texturas com elevada variedade; esta capacidade não existe associada a outros tipos de técnicas. Como desvantagem, este tipo de técnicas apresenta normalmente um elevado custo computacional.

2.2.1.3 Segmentação Espacial Baseada em Fronteiras

Pode definir-se ‘fronteira’ como sendo uma zona onde ocorre uma ou mais variações nas características da imagem. As técnicas de segmentação baseadas na detecção de fronteiras assumem que o valor de pelo menos uma propriedade dos *pixels* varia rapidamente na fronteira entre duas regiões. Assim, estas técnicas procuram localizar variações abruptas nos valores de alguma propriedade dos *pixels*, tais como o nível de cinzento, cor, contraste ou alguma outra medida local que permita identificar uma fronteira entre duas regiões.

O processo de segmentação baseada na detecção de fronteiras pode ser dividido em três etapas principais [Correia02]:

1ª Detecção das fronteiras

Esta etapa consiste tipicamente na aplicação de operadores para detecção de fronteiras. Estes operadores podem basear-se em uma de duas aproximações [Pratt91]:

- **Detecção das diferenças espaciais na imagem** – Neste tipo de abordagem, a imagem é processada de forma a acentuar as variações espaciais de amplitude, i.e. as zonas onde se localizam as fronteiras. Para tal, são normalmente empregues dois tipos de técnicas [Gonzalez93]:
 - ♦ **Cálculo da derivada de primeira ordem** – Nestas técnicas, as zonas da imagem sobre as quais o cálculo da derivada de primeira ordem produz valores elevados correspondem a descontinuidades, i.e. fronteiras. A primeira derivada pode ser estimada através do cálculo dos gradientes (na vertical e horizontal) na vizinhança de cada *pixel*;
 - ♦ **Cálculo da derivada de segunda ordem** – Nestas técnicas, a passagem por zero na segunda derivada indica a presença de fronteiras na imagem, uma vez que este zero corresponde ao ponto central de uma transição na imagem.

Este tipo de detecção é usualmente implementado com base nos designados detectores de fronteira, tais como operadores de Sobel, Roberts, Prewitt, Laplacian e Canny. Exemplos de imagens processadas com estes tipos de detectores de fronteiras podem ser observados em [Canny86, Jain89, Pratt91, Haralick92, Gonzalez93]. Na Figura 2.7 ilustram-se exemplos da detecção de fronteiras utilizando vários operadores.

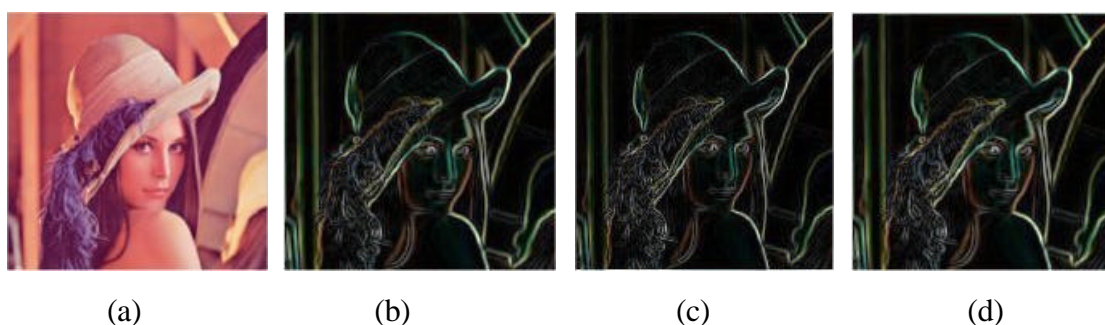


Figura 2.7 – Exemplos da detecção de fronteiras: (a) imagem original; (b), (c) e (d) resultado da detecção de fronteiras utilizando os operadores de Prewitt, Roberts e Robison, respectivamente, para imagem em (a).

- **Adaptação a um dado modelo de fronteira** – Neste tipo de abordagem, os valores dos *pixels* (luminância ou luminância e crominâncias) correspondentes a uma determinada zona da imagem são comparados com um modelo de fronteiras. Esta técnica pressupõe o conhecimento *a priori* do tipo de fronteiras esperado para a imagem. Por exemplo, as fronteiras podem ser detectadas através do seguimento de modelos paramétricos, tais como linhas rectas, círculos ou elipses. Neste caso, técnicas especiais como a

transformada de Hough têm de ser aplicadas para fazer a identificação das fronteiras [Pratt91].

As fronteiras resultantes da aplicação destes operadores são normalmente descontínuas. Para além disso, nas imagens onde as fronteiras não são muito contrastadas podem surgir falsas detecções, localizadas onde não existem realmente limites de regiões, ou então, as fronteiras podem ser omitidas onde os limites das regiões realmente existem.

2ª Selecção das fronteiras

A segunda etapa do processo de segmentação baseado na detecção de fronteiras consiste na selecção dos segmentos de fronteiras mais relevantes detectados no passo anterior. Esta selecção pode ser feita com base nas seguintes técnicas [Jain89]:

- **Limiar de fronteira** – Neste caso, a selecção de fronteiras na imagem faz-se através da utilização de um valor limiar que permita remover as fronteiras detectadas com um valor de gradiente inferior a esse limiar;
- **Relaxação de fronteira** – Neste caso, uma medida da qualidade de fronteira é calculada para cada fronteira, decidindo-se assim quais as fronteiras que devem, ou não, ser descartadas. Para tal, é analisada a magnitude da fronteira, i.e. o valor do seu gradiente, bem como o contexto onde a fronteira existe de modo a avaliar a qualidade de cada fronteira. O critério mais usado para descartar uma fronteira é baseado num valor de limiar determinado em função dos valores de qualidade pretendidos.

3ª Identificação das regiões

Na terceira e última etapa, as fronteiras seleccionadas na etapa anterior são combinadas em cadeias de forma a definirem os limites das várias regiões. Após a conclusão desta etapa, os *pixels* que não estiverem separados por uma fronteira são considerados como fazendo parte da mesma região.

A identificação dos limites das regiões pode ser efectuada utilizando as seguintes técnicas [Gonzalez93, Jain89]:

- **Ligação de fronteiras** – As fronteiras podem ser ligadas entre si, se estiverem próximas umas das outras. Assim, se uma fronteira estiver próxima de outra e se o ângulo entre as suas tangentes for relativamente pequeno, estas podem ser ligadas;
- **Transformada de Hough** – Se os limites das regiões procuradas seguirem um modelo paramétrico conhecido, por exemplo se a forma do objecto for conhecida, pode utilizar-se a transformada de Hough para localizar esses limites a partir das fronteiras anteriormente detectadas na imagem;
- **Procura em grafos** – Faz-se uma representação das fronteiras detectadas usando um grafo onde os limites das regiões correspondem a caminhos nesse grafo. Como informação inicial, apenas são necessários os pontos de início e fim do limite da região. Desta forma, uma cadeia de fronteiras representativas do caminho óptimo, para esse limite, pode ser determinada usando uma função de avaliação de caminhos;
- **Programação dinâmica** – Utiliza-se o princípio de optimização de Bellman's que diz o seguinte “*o caminho óptimo entre dois pontos é igualmente óptimo entre quaisquer dois*”

pontos situados no mesmo caminho” [Jain89]. Este princípio pode ser aplicado ao problema da determinação das fronteiras das regiões, se for definida uma noção de ‘boa fronteira’. Esta técnica pode ser utilizada para seleccionar a melhor fronteira de entre as várias cadeias de fronteiras existentes entre um ponto de início e um de fim.

A segmentação baseada na detecção de fronteiras apresenta como principal vantagem um custo computacional razoável associado aos detectores de fronteira. Como ponto fraco, há que referenciar a sua sensibilidade ao ruído, especialmente quando se usam janelas muito pequenas como máscaras/filtros de detecção. O facto destas técnicas se basearem unicamente na informação espacial leva a que possam produzir um número elevado de pequenas regiões, sobretudo para imagens muito texturadas.

2.2.1.4 Segmentação Espacial Baseada em Regiões

Na segmentação baseada em regiões parte-se do pressuposto que *pixels* adjacentes e pertencentes a uma mesma região têm características visuais semelhantes, por exemplo em termos de níveis de cinzento, cor ou textura. Dependendo do tipo de aplicação que a segmentação irá ter, da imagem a segmentar e dos resultados pretendidos, a selecção dos critérios de homogeneidade pode ser mais ou menos sofisticada. Entre os critérios mais relevantes encontram-se a gama dinâmica, a média, o valor médio e a variância, aplicadas às componentes de uma imagem, tais como luminância e crominâncias ou componentes RGB.

Duas condições básicas têm que ser respeitadas pelas regiões obtidas através da utilização de algoritmos de segmentação baseados em regiões [Correia02]:

- Cada uma das regiões tem obrigatoriamente que verificar o critério de homogeneidade seleccionado;
- A união de duas regiões adjacentes não pode verificar o critério de homogeneidade seleccionado.

As várias técnicas utilizadas para efectuar a segmentação baseada em regiões podem ser organizadas nas seguintes classes:

- **Region-splitting** – Nesta classe de técnicas, os algoritmos partem duma região do tamanho da imagem ou de um grupo pré-definido de regiões, dividindo-a(s) progressivamente em regiões mais pequenas de acordo com o critério de homogeneidade seleccionado. Quando as duas condições definidas anteriormente forem cumpridas, obtém-se a partição pretendida;
- **Region-merging** – Nesta técnica, contrariamente à técnica anterior, parte-se de uma imagem segmentada em pequenas regiões (no limite todos os *pixels* da imagem), proveniente duma fase de pré-processamento e fazem-se crescer em área essas regiões através da fusão com regiões vizinhas, aplicando critérios de semelhança. A descrição completa de uma técnica deste tipo pode ser encontrada em [Salembier97];
- **Split-and-merge** – Esta técnica resulta da combinação de uma técnica de *region-splitting* com uma técnica de *region-merging*. Nesta abordagem, considera-se normalmente uma representação piramidal da imagem onde as regiões correspondem à divisão da imagem em quadrados que, consoante a sua dimensão, correspondem aos vários níveis da

pirâmide. O processo de segmentação começa com a divisão da imagem em regiões quadradas com uma determinada dimensão. Caso estas regiões não sejam homogéneas segundo o critério seleccionado, são novamente divididas (usualmente em quatro sub-regiões), repetindo-se este processo até que as sub-regiões sejam homogéneas. No processo de agrupamento (*merging*), quando um dado conjunto de regiões adjacentes de um dado nível da pirâmide é homogéneo, as regiões são agrupadas numa única região, num nível superior da pirâmide. Exemplos de técnicas deste tipo podem ser encontradas em [Horowitz72, Haralick85, Cortez95]. A Figura 2.8 mostra um exemplo da utilização desta técnica de segmentação, podendo ver-se a imagem original (Lena), a segmentação no fim da fase de *splitting* e a segmentação no fim da fase de *merging*;



Figura 2.8 – Exemplos de segmentação usando *split-and-merge*: (a) imagem original; (b) fim da fase de *splitting* da imagem em (a); (c) fim da fase de *merging* aplicada à imagem resultante do *splitting*.

- **Region-growing** – Nesta técnica, a segmentação é feita de forma semelhante à abordagem *region-merging* uma vez que regiões vizinhas com propriedades semelhantes são também agrupadas. A grande diferença entre as duas abordagens é que no caso *region-growing* o ponto de partida não é uma imagem completamente segmentada como nas técnicas de *region-merging* mas sim um conjunto de sementes (um *pixel* ou conjunto de *pixels*) que é sabido fazerem parte das várias regiões que se pretendem obter [Zucker76, Moghaddamzadeh97, Fan01].

As técnicas utilizadas na segmentação baseada em regiões apresentam, como principal vantagem, a sua eficiência na identificação de regiões homogéneas em termos das características espaciais seleccionadas, bem como a sua exactidão na localização das fronteiras. Como maior desvantagem, deve considerar-se o elevado número de regiões que tipicamente surgem como resultado da segmentação.

2.2.2 Segmentação Temporal

Os algoritmos utilizados na segmentação temporal efectuem a segmentação através da avaliação da homogeneidade existente na dimensão temporal, i.e. a homogeneidade existente em termos do movimento dos *pixels* entre tramas consecutivas. Para se efectuar este tipo de segmentação do vídeo (não se pode fazer para imagens por não haver dimensão temporal), começa-se, usualmente, por estimar um vector de movimento para cada parte da imagem, por

exemplo um *pixel* ou um bloco de *pixels*, para assim se poderem identificar as regiões com movimentos coerentes. Um aspecto importante da segmentação temporal é a possibilidade de efectuar o seguimento no tempo das regiões identificadas ao longo da sequência e consequentemente dos objectos constituídos por conjuntos de regiões; esta capacidade assegura a coerência temporal entre imagens segmentadas consecutivas.

As duas principais técnicas de segmentação baseadas na dimensão temporal são a segmentação baseada na detecção de alterações e a segmentação baseada em movimento [Correia02].

2.2.2.1 Segmentação Temporal Baseada na Detecção de Alterações

Na segmentação baseada na detecção de alterações é feita a divisão da imagem entre os *pixels* que alteram (significativamente) a sua amplitude e os que se mantêm (essencialmente) inalterados ao longo do tempo ou seja há apenas duas regiões (eventualmente não conexas) na imagem. Esta separação é basicamente efectuada através da comparação da imagem actual com a imagem anterior (ver exemplo na Figura 2.9).

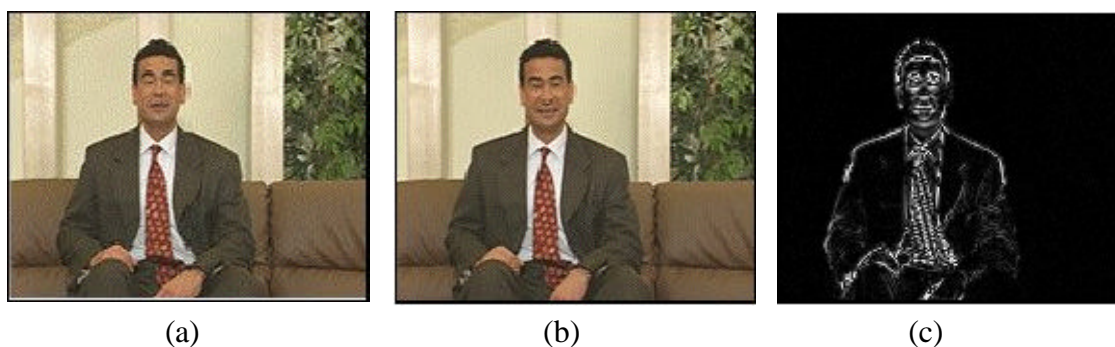


Figura 2.9 – Exemplo do 1º passo da segmentação temporal baseada na detecção de alterações: a diferença entre as componentes de luminância das imagens (a) e (b) é apresentada em (c) [Correia02].

Neste tipo de segmentação, é criada uma partição binária para a imagem actual através da classificação das zonas da mesma como alteradas ou não alteradas em relação à imagem anterior. Dependendo das características da aplicação e do conteúdo que está a ser analisado, este resultado pode ser visto como o resultado final da segmentação ou como uma base para posterior refinamento da segmentação baseada na informação espacial.

A segmentação baseada na detecção de alterações divide-se, basicamente, em três passos:

- 1º Determinação das diferenças entre imagens consecutivas** – Neste passo, as diferenças existentes entre imagens sucessivas são calculadas, depois da compensação de movimento global para evitar que os movimentos provenientes da câmara de filmar influenciem demasiado o resultado da segmentação;
- 2º Definição do valor de limiar para a imagem das diferenças** – Neste passo, define-se o valor do limiar a aplicar à imagem das diferenças com vista à obtenção de uma imagem

segmentada binária e logo das 2 regiões muitas vezes designadas por *background* e *foreground*. O valor do limiar usado pode ser pré-definido ou calculado de forma dinâmica com base em propriedades da imagem como, por exemplo, a variação do ruído da câmara de filmar [Thoma89];

3º Pós-processamento do resultado do passo anterior – Neste passo, o resultado do cálculo das diferenças pode ser optimizado através do pós-processamento da imagem binária resultante do passo anterior, por exemplo com vista a eliminar regiões demasiado pequenas. Exemplos de pós-processamento podem ser encontrados em [Hötter88, Musmann89, Thoma89, Mech98].

Uma limitação da segmentação baseada na detecção de alterações, bem como da generalidade dos algoritmos baseados em segmentação temporal, prende-se com a dificuldade, se não mesmo impossibilidade, de detectar o movimento de objectos com cor uniforme levando assim a erros de segmentação. No caso particular da detecção de alterações, algumas das regiões (uniformes) constituintes de um objecto com movimento podem fazer parte de zonas detectadas sem alteração o que provoca a criação de buracos no objecto em questão.

2.2.2.2 Segmentação Temporal Baseada em Movimento

A segmentação baseada em movimento permite fazer a distinção entre objectos com movimentos diferentes. Estas técnicas têm por base métodos que estimam a informação de velocidade 2D para assim tentarem identificar regiões homogéneas em termos das suas características de movimento. Este tipo de técnicas procura identificar a presença de um conjunto de objectos com movimento através da análise de uma estimativa da informação de movimento associada à imagem, por exemplo o campo de vectores de movimento.

Várias técnicas têm sido propostas na literatura para implementar segmentação baseada em movimento, depois de efectuada uma estimativa do mesmo. A seguinte classificação para estas técnicas foi proposta em [Correia02]:

- **Técnicas de *clustering*** – Estas técnicas procuram localizar conjuntos de *pixels* com vectores de movimento com propriedades semelhantes. Podem ser baseadas em:
 - ♦ **Modelos de movimento paramétrico** – Estes modelos assumem um conjunto inicial de regiões com movimento para as quais são calculados modelos de movimento paramétrico. O número inicial de regiões consideradas deve ser superior ao número de regiões que se pretende segmentar. Seguidamente, cada *pixel* é transferido para um conjunto que corresponde a uma região com movimento, de acordo com o vector de movimento do modelo paramétrico que melhor o represente. Um exemplo da aplicação desta técnica pode ser visto em [Wang94];
 - ♦ **Transformadas de Hough** – Estes modelos utilizam a transformada de Hough para agrupar os vectores de movimento em regiões. A utilização da transformada de Hough parte do princípio que um dado modelo paramétrico pode ser utilizado para descrever as regiões com movimento. Assim, a transformada de Hough é utilizada para localizar os *pixels* de cada região e procurar a melhor correspondência com o modelo de movimento. Um exemplo é apresentado em [Kruse96];

- ♦ **Modelos combinados** – Estes modelos fazem o agrupamento dos vectores de movimento em regiões através da combinação das suas características, utilizando para tal uma classificação Bayesiana. Estas características podem ser as coordenadas dos *pixels* e as componentes dos seus vectores de movimento; todavia, outras características podem ser utilizadas para efectuar o processo de agrupamento. Um exemplo da aplicação desta técnica pode ser encontrado em [Chalom95].
- **Técnicas hierárquicas** – Estas técnicas fazem a segmentação do movimento através da aplicação sucessiva de algoritmos que detectam o movimento dominante. Em cada iteração, o objecto com mais movimento é identificado. Este objecto é então removido da imagem que vai ser processada na próxima iteração, continuando o ciclo até que toda a imagem esteja segmentada. Um exemplo da aplicação desta técnica pode ser visto em [Wu93];
- **Técnicas baseadas em campos aleatórios de Markov¹** – Estas técnicas formulam o problema da segmentação do movimento como uma estimativa probabilística de um campo que é modelado por um campo aleatório de Markov. A modelação de Markov é uma formalização da informação contextual, tornando-a assim apropriada para os problemas de análise de imagem onde a maioria das medidas pode ser feita sobre as vizinhanças dos *pixels*. A função de energia global resultante da fase de modelação de Markov pode ser minimizada utilizando para tal técnicas Bayesianas. Um exemplo da aplicação de campos de Markov à segmentação baseada em movimento pode ser visto em [Murray87].

Como maiores vantagens da utilização da segmentação temporal de vídeo, encontram-se a sua eficácia na detecção de regiões que sejam homogéneas em termos de movimento, bem como a possibilidade de utilizar a informação temporal para fazer o seguimento de objectos ao longo da sequência de imagens. Como desvantagens, devem apontar-se os factos de as técnicas baseadas na segmentação temporal serem incapazes de detectar objectos estáticos no tempo, apresentarem pouca precisão na localização das fronteiras das regiões e ainda, nalguns casos, um elevado custo computacional.

2.2.3 Combinação da Segmentação Espacial e Temporal

Para um grande número de aplicações, a combinação da utilização de ambos os tipos de técnicas – espacial e temporal – apresenta-se como a melhor solução, uma vez que os dois tipos de técnicas se complementam, compensando mutuamente as limitações inerentes a cada um dos tipos [Correia02]. Como se verá de seguida, as técnicas espaciais e temporais podem ser combinadas de três formas distintas.

2.2.3.1 Segmentação Temporal Depois da Espacial

Neste tipo de técnicas, primeiro é efectuada uma segmentação espacial, que é posteriormente complementada/melhorada com informação proveniente de uma segmentação temporal.

¹ *Markov Random Fields* (MRF)

Usualmente, os resultados da segmentação espacial contêm várias regiões pertencentes a um mesmo objecto (ou seja há sobresegmentação) que vão poder ser agrupadas utilizando a informação resultante da segmentação temporal que indica, por exemplo, que várias regiões espaciais têm o mesmo tipo de movimento. Esta informação temporal adicional pode ser utilizada para manter a coerência temporal dos objectos quando é efectuada a segmentação de sequências de vídeo. Um exemplo dum algoritmo que utiliza esta técnica pode ser encontrado em [Choi97].

2.2.3.2 Segmentação Espacial Depois da Temporal

Este tipo de técnicas começam por efectuar uma segmentação temporal que é seguida de uma segmentação espacial, com o objectivo de melhorar os resultados da primeira segmentação. Dependendo das ferramentas de estimação de movimento usadas e das inexactidões por elas originadas, alguns erros podem surgir na segmentação temporal, sobretudo na localização dos contornos das regiões. Por exemplo, os objectos com movimento semelhante são agrupados através da segmentação temporal, enquanto que os objectos sem movimento ou partes estáticas dos objectos com movimento, não podem ser detectados unicamente com recurso a este tipo de segmentação. Assim, o resultado da segmentação temporal é melhorado através da aplicação de um passo de segmentação espacial. Este passo pode simplesmente consistir na correcção da localização dos contornos das regiões ou pode incluir a detecção de regiões que não foram detectadas no processo temporal. A escolha do tipo de combinação mais adequada depende do tipo de aplicação em causa e do tipo de resultados pretendidos [Correia02].

2.2.3.3 Segmentação Temporal e Espacial em Simultâneo

Esta abordagem consiste em utilizar simultaneamente ambos os tipos de informação, espacial e temporal, durante o processo de segmentação que não é constituído por duas fases claramente sucessivas. É nesta classe de técnicas que se encontram aquelas que são mais completas e que apresentam melhores resultados na segmentação de sequências de imagens. Vários algoritmos têm sido desenvolvidos utilizando este tipo de estratégia, tais como os descritos em [Salembier94, Choi97, Moscheni96].

A combinação da segmentação espacial e temporal apresenta como principais vantagens a sua eficácia na identificação de regiões que são homogéneas em termos de características espaciais e/ou temporais, bem como uma boa capacidade para efectuar o seguimento de objectos ao longo das sequências de imagens. Como desvantagem, apresentam um elevado custo computacional devido à combinação de diferentes técnicas.

2.3 Técnicas de Classificação das Regiões

Depois da segmentação de uma imagem ou trama em regiões através de métodos como os analisados na secção 2.2, o resultado dessa segmentação, i.e. as regiões provenientes da segmentação, necessita de ser representado de forma eficiente para que possa mais facilmente ser processado e classificado por computador. Lembre-se que, no contexto da extracção de texto em imagens e sequências de vídeo, pretende-se classificar cada região como texto ou não texto.

Basicamente, a descrição de uma região pode ser efectuada com base nas suas características internas ou externas [Gonzalez93]:

- A região é descrita com base nas suas características externas, i.e. do seu contorno; normalmente, uma descrição baseada no contorno é escolhida quando as características da forma do objecto são as mais importantes;
- A região é descrita com base nas suas características internas, i.e. dos *shapels* que a constituem; normalmente, uma descrição deste tipo é escolhida quando as características de reflectividade do objecto tais como a cor e a textura são as mais importantes.

Em qualquer dos casos, as características escolhidas para descrever uma região devem ser tão insensíveis quanto possível a variações como mudanças de escala, rotações e translações.

2.3.1 Ferramentas de Descrição de Regiões

O conteúdo do vídeo pode ser descrito utilizando vários tipos de descritores e esquemas de descrição. As ferramentas de descrição podem ser aplicadas individualmente ou em conjunto a sequências de imagens ou a imagens individuais. Características do vídeo que são tipicamente consideradas relevantes são a forma, a cor, a textura, o movimento e a localização dos objectos existentes na cena representada; descritores associados a estas características, por exemplo os definidos pela norma MPEG-7, podem ser utilizados para efectuar a descrição do vídeo ou seja das regiões que o compõem.

2.3.1.1 Descrição da Forma

Idealmente, a descrição da forma de uma região deve ser invariante a variações como mudanças de escala, rotações e translações. Duas classes principais de descritores de forma são consideradas em [Zibreira00]:

- **Descritores de forma baseados no contorno** – Os descritores baseados no contorno descrevem uma região conexa tendo em conta os seus *shapels* mais exteriores ou seja o contorno fechado da mesma. A Figura 2.10 mostra um exemplo de um contorno fechado a ser descrito por este tipo de parâmetros.



(a)



(b)

Figura 2.10 – *Bream* (trama 1): (a) Imagem com um objecto; (b) contorno do objecto em (a).

Os principais parâmetros de forma baseados no contorno disponíveis na literatura podem ser organizados segundo as suas propriedades do seguinte modo [Zibreira00]:

- ♦ **Parâmetros geométricos** – Parâmetros que representam a forma de um objecto simples (ou seja com uma única região) usando propriedades geométricas do seu contorno tais como o perímetro, a corda máxima, a circularidade, a convexidade e a excentricidade [Russ95];
- ♦ **Parâmetros baseados em transformadas** – Parâmetros que representam a forma de um objecto simples utilizando coeficientes calculados a partir de uma dada transformada; exemplos são a transformada de Fourier e as *wavelets* [Jain89, Antonini94, Bimbo99, Muller99];
- ♦ **Parâmetros baseados em momentos** – Parâmetros que representam a forma de um objecto simples utilizando um conjunto de valores estatísticos; exemplos são os momentos geométricos, também denominados como momentos invariantes [Muller99];
- ♦ **Parâmetros baseados em contornos normalizados** – Parâmetros que representam a forma de um objecto simples utilizando o seu contorno normalizado; o contorno normalizado é insensível a transformações geométricas e ao número de pontos que o definem [Tabatabai99];
- ♦ **Parâmetros baseados nos ângulos de curvatura da forma** – Parâmetros que representam a forma de um objecto simples através de um conjunto de ângulos de curvatura extraídos a partir do seu contorno [IBMRe99, Niblack95];
- ♦ **Parâmetros baseados numa imagem *Curvature Scale Space (CSS)*** – Parâmetros que representam a forma de um objecto simples com vários níveis de detalhe de acordo com os pontos de inflexão (convexos ou côncavos) do contorno fechado [Bober99, Mokhtarian99].
- **Descritores de forma baseados em regiões** – Os descritores baseados em regiões descrevem uma região tendo em conta todos os seus *shapels*. Os parâmetros de forma baseados em regiões descrevem formas simples mas também formas mais complexas, por exemplo a forma de um objecto formado por várias regiões não conexas [MPEG7-Visual01]. A Figura 2.11 apresenta alguns objectos que poderão ser descritos por parâmetros baseados em regiões.



Figura 2.11 – Exemplos de objectos simples e complexos, com as respectivas regiões e buracos [MPEG7-Visual01]

Os principais parâmetros de forma baseados em regiões disponíveis na literatura podem ser organizados segundo as suas propriedades do seguinte modo [Zibreira00]:

- ♦ **Parâmetros geométricos** – Parâmetros que representam a forma de um objecto simples ou complexo usando as propriedades geométricas da região ou regiões que lhe correspondem; exemplos são a *bounding box*, área, centróide, projecções: altura e largura, diâmetro circular equivalente, solidez e compactação [Russ95];
- ♦ **Parâmetros baseados em transformadas** – Parâmetros que representam a forma de um objecto simples ou complexo utilizando coeficientes calculados a partir de uma dada transformada; exemplos são a Transformada Angular-Radial (*Angular-Radial Transform, ART*) e a transformada de Fourier [Kim00a, Kim00b];
- ♦ **Parâmetros baseados em momentos** – Parâmetros que representam a forma de um objecto simples ou complexo utilizando um conjunto de valores estatísticos associados a um dado tipo de momento; exemplos são os momentos geométricos, os momentos de Legendre, os momentos de Zernike, os momentos rotacionais e os momentos complexos [Teh88, Khotanzad90, Kim99a, Kim99b];
- ♦ **Parâmetros baseados em vectores próprios multi-nível (*Multi Layer Eigen Vectors, MLEV*)** – Parâmetros que representam a forma de um objecto simples ou complexo com vários níveis de detalhe, subdividindo sucessivamente o objecto segundo as direcções dos vectores próprios dos eixos principais, calculadas na iteração anterior [Kim99c, Kim99d].

2.3.1.2 Descrição da Cor

A utilização da característica cor para a descrição das regiões assenta em dois tipos de informação: as características das cores descritivas de cada região e a distribuição espacial das mesmas. Por exemplo, uma descrição MPEG-7 da cor pode basear-se em quatro aspectos essenciais: cor dominante, histogramas de cor, selecção do espaço de cor e estrutura da cor [MPEG7-Visual01].

Os descritores MPEG-7 que exprimem características das cores correspondentes a cada região são:

- **Cor dominante** – Este descritor permite fazer uma descrição das cores mais representativas de uma região ou conjunto de regiões de uma imagem. Tipicamente este descritor inclui a especificação do espaço de cor seleccionado, da cor ou cores dominantes e da percentagem de *pixels* para cada uma das cores dominantes [MPEG7-Visual01];
- **Estrutura da cor** – Este descritor descreve uma imagem ou uma região de uma imagem em termos do seu conteúdo de cor e do tipo de estrutura ou distribuição espacial da cor. A estrutura de cor de uma imagem reflecte o grau de agregação dos *pixels* em termos de cor [MPEG7-Visual01]. Na Figura 2.12 apresenta-se um exemplo de uma imagem com cor altamente estruturada e um outro exemplo com cor altamente não-estruturada. No segundo exemplo, os *pixels* de cor vermelha estão “espalhados” por toda a imagem enquanto que no primeiro exemplo estão agregados numa zona rectangular. Como o número de *pixels* desta cor é igual nas duas imagens e todos os outros *pixels* têm a mesma

cor, um histograma de cor não permitiria diferenciar estas duas imagens; no entanto, a utilização do descritor estrutura de cor permite diferenciá-las;

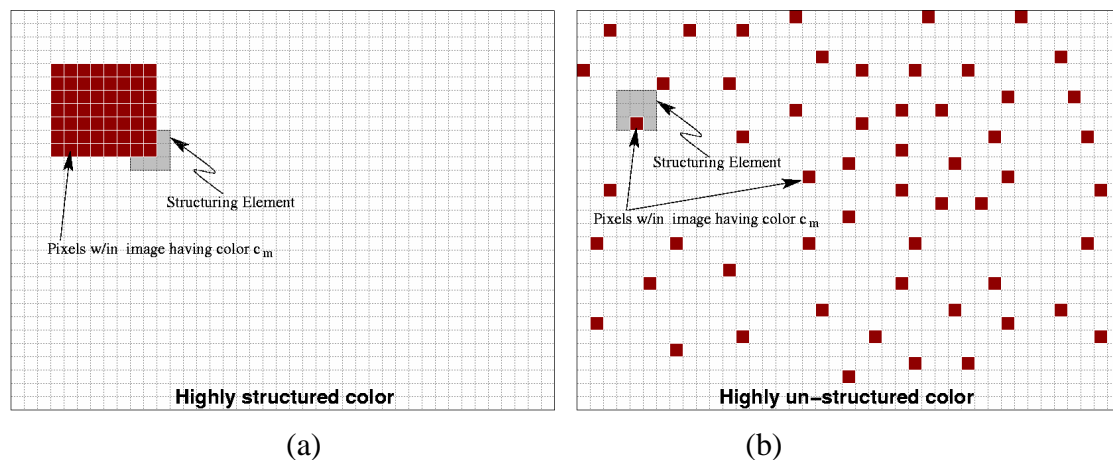


Figura 2.12 – Exemplos de imagens com: (a) cor altamente estruturada; (b) cor altamente não-estruturada [MPEG7-Visual01].

- **Histograma de cor** – Este descritor descreve a distribuição estatística da cor numa imagem ou região de uma imagem através do seu histograma. O número de ocorrências de cada cor ou gama de cores pode ser representado utilizando para tal um histograma de cor. Tipicamente, os histogramas de cada componente de cor são calculados separadamente e os seus níveis de representação binária (*bins*) podem ser utilizados na descrição da cor. Exemplos da utilização de histogramas de cor na descrição de vídeo podem ser encontrados em [MPEG7-Visual01];
- **Seleção do espaço de cor** – Este descritor permite seleccionar o espaço de cor a ser utilizado na descrição. É possível utilizar qualquer um dos seis espaços de cor especificados pela norma MPEG-7: YCbCr, Transformação Linear do Espaço RGB, HSV, HMMD e Monocromático para descrever o espaço de cor[MPEG7-Visual01].

2.3.1.3 Descrição da Textura

A textura pode ser descrita utilizando vários tipos de descritores e esquemas de descrição. Alguns exemplos são apresentados em [Correia02]:

- **Matrizes de co-ocorrência** – As matrizes de co-ocorrência registam a frequência relativa com que um par de *pixels* com um dado valor e separados por uma determinada distância ocorrem numa imagem. Várias matrizes para várias orientações de co-ocorrências são normalmente consideradas. Este tipo de medidas reflecte as características da textura da área em análise. Exemplos deste tipo de descritor podem ser encontrados em [Haralick92];
- **Transformadas** – Várias transformadas podem ser utilizadas para representar a imagem e assim descrever a sua textura. Alguns exemplos são a transformada *wavelet*, a

transformada Gabor e a transformada de Fourier. Exemplos deste tipo de descritor podem ser encontrados em [Rui98];

- **Campos aleatórios de Markov** – Os campos aleatórios de Markov permitem classificar as regiões da imagem como pertencendo a uma dada classe de textura com base na informação textual calculada numa vizinhança local. Exemplos deste descritor podem ser encontrados em [Nunes95].

Outros descritores podem, todavia, ser considerados para efectuar a descrição da textura, tais como: regularidade, rugosidade, homogeneidade, direccionalidade e contraste. Exemplos destes descritores podem ser encontrados em [MPEG7-Visual01].

2.3.1.4 Descrição do Movimento

Para efectuar a descrição de sequências de vídeo deve ser tida em conta a dimensão temporal e logo o movimento,. O movimento pode ser descrito utilizando os seguintes descritores [MPEG7-Visual01]:

- **Movimento da câmara de filmar** – Este descritor descreve os parâmetros de movimento 3D da câmara de filmar. Uma estimativa do movimento da câmara de filmar fornece dois tipos de informação para a descrição: qualitativa (p.e. *zoom-in*) e quantitativa (p.e. conjunto dos valores paramétricos para o modelo de movimento usado).

Este descritor permite caracterizar as seguintes operações de câmara de filmar (ver Figura 2.13): fixa, rotação horizontal (*panning*), movimento transversal vertical (*booming*), *zooming* (mudança de comprimento focal), translação sobre o eixo óptico (*dollying*) e rotação em torno do eixo óptico (*rolling*).

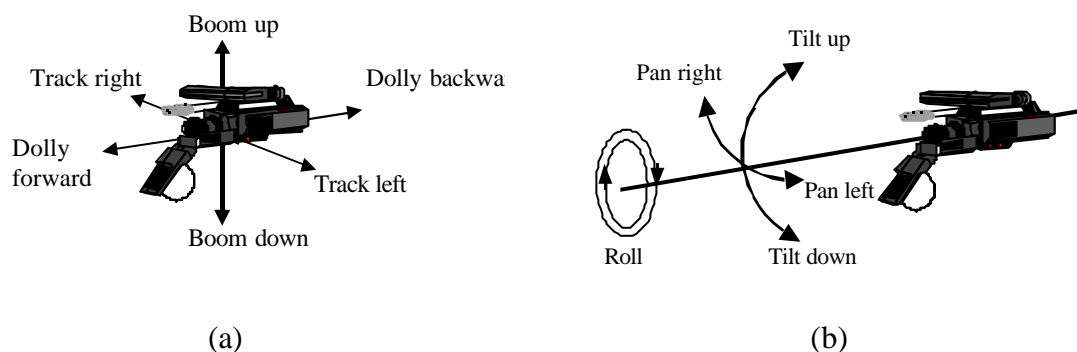


Figura 2.13 – Exemplos de movimentos de câmara de filmar: (a) *tracking*, *booming* e *dollying*; (b) *panning*, *tilting* e *rolling* [MPEG7-Visual01].

- **Movimento do objecto e trajectória** – Este descritor descreve os parâmetros do movimento de cada objecto que é considerado importante numa cena; este movimento pode ser determinado quando a segmentação da cena está disponível e pode ser descrito utilizando um modelo paramétrico. Através do seguimento dos objectos no tempo, podem descrever-se as suas trajectórias;

- **Actividade do movimento** – Este descritor captura a noção intuitiva de intensidade de acção ou ritmo de acção num segmento de vídeo. A descrição é efectuada utilizando atributos tais como a intensidade do movimento, a direcção do movimento, a localização espacial do movimento e ainda as suas distribuições espacial e temporal.

2.3.1.5 Descrição da Localização

As características de localização referem-se ao posicionamento espacial dos objectos ou regiões dentro da imagem ou ao posicionamento espacial-temporal dos objectos dentro de uma sequência de vídeo. Como descritores de localização podem ter-se [MPEG7-Visual01]:

- **Localizador de objecto** – Este descritor dá a localização de um objecto ou de uma região dentro de uma imagem utilizando uma *bounding box* ou outra representação poligonal que confine a posição do objecto ou região. Assim, um conjunto de coordenadas espaciais permite localizar a posição da *bounding box* na imagem bem como a sua orientação e dimensão;
- **Localizador espacial-temporal** – Este descritor efectua a descrição da localização de uma região num determinado instante; para efectuar esta localização, é necessário definir a trajectória da região ou objecto a descrever.

Uma vez efectuada a descrição das regiões, estas podem ser comparadas com descrições/valores previamente definidos/classificados com vista a efectuar a classificação de cada uma das regiões como texto ou não. Para tal, é necessário dispor de medidas de semelhança que meçam, eficazmente, a semelhança existente entre os valores utilizados na descrição de cada uma das regiões detectadas e valores previamente definidos típicos de regiões de texto.

2.3.2 Métodos Utilizados na Classificação das Regiões

Depois de efectuada a descrição das várias regiões segmentadas, utilizando para tal um ou mais dos parâmetros/descriptores anteriormente apresentados, torna-se necessário classificar cada região segmentada como sendo texto ou não. Para efectuar essa classificação podem ser encontrados na literatura vários tipos de métodos [Ohya94, Lienhart95, Zhong95, Jain98, Hasan00, Hase01, Li98, Messelodi99, Lienhart00, Chen01, Liu02, Lienhart02, Wolf02, Zhang02]. De entre estes métodos, os dois tipicamente mais utilizados são:

- **Análise geométrica das regiões** – Este tipo de método efectua comparações entre os valores dos parâmetros que descrevem as regiões segmentadas na imagem ou trama e determinados valores previamente definidos que caracterizam a presença de texto em termos dos parâmetros em questão. As regiões que não verificarem os critérios de filtragem que caracterizam o texto são classificadas como não texto e posteriormente descartadas. Alguns exemplos da aplicação de filtros baseados em heurísticas podem ser encontrados em [Lienhart95, Zhong95, Jain98, Messelodi99, Lienhart00, Chen01]. Estes métodos apresentam dificuldades na classificação, sobretudo quando o texto possui caracteres de vários tamanhos e quando estes se tocam;

- **Redes neuronais** – Este tipo de método utiliza redes neuronais para efectuar a classificação de cada região como texto ou não. Os valores dos parâmetros que descrevem cada região segmentada servem de entrada para a rede neuronal. A resposta da rede é comparada com um limiar pré-definido característico da presença de texto para assim efectuar a classificação de cada região como texto ou não. Alguns exemplos da aplicação de métodos de classificação baseados em redes neuronais podem ser encontrados em [Li98, Li00, Li02, Lienhart02]. A eficácia deste tipo de métodos depende muito da qualidade do treino feito à rede neuronal. Atendendo a que o texto existente no vídeo possui vários tamanhos, fontes, estilos, etc., o treino de um classificador neuronal genérico torna-se particularmente difícil;
- **Outros métodos** – Outros métodos existem, ainda que menos difundidos, para classificar como texto ou não as várias regiões constituintes das imagens ou tramas de vídeo, por exemplo métodos baseados em operadores morfológicos [Hasan00], métodos baseados em *stochastic relaxation* [Ohya94 e Hase01], métodos baseados nos momentos de Zernike [Zhang02] e métodos baseados na detecção de pontos salientes [Bertini01].

Na prática, os sistemas mais relevantes propostos na literatura para a extracção de texto em sequências de vídeo ou imagens utilizam, normalmente, mais do que um dos métodos apresentados para efectuar a classificação das regiões segmentadas como texto ou não, tentando, assim, aumentar o desempenho global dos sistemas de classificação de texto [Lienhart02, Wolf02].

2.4 Técnicas de Seguimento

O seguimento de um objecto numa sequência de vídeo consiste em detectar e localizar a sua presença em várias tramas sucessivas de forma a manter a coerência da segmentação ao longo do tempo ou seja, conhecer a correspondência entre as regiões segmentadas em tramas sucessivas. Esta característica é essencial no contexto da detecção e extracção de texto em sequências de vídeo, pois permite a exploração da redundância temporal existente no vídeo para melhorar a capacidade de detecção dos algoritmos.

As técnicas de seguimento existentes na literatura podem ser agrupadas em duas classes principais:

- 1ª **Comparação de tramas sucessivas** – A primeira classe de técnicas faz o seguimento de uma partição previamente obtida, focando-se sobre duas tramas de cada vez, i.e. relacionam o resultado do momento anterior com o do momento actual. Exemplos deste tipo de técnicas podem ser observados em [Adiv85, Garduno94, Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Este tipo de algoritmos é frequentemente mais simples, ainda que possa apresentar algumas limitações quando tem que lidar com a oclusão temporária de objectos. Enquanto algumas técnicas fazem o seguimento de um conjunto de características como, por exemplo, os pontos das fronteiras ou as características de uma região, tais como cor, tamanho, forma, etc. [Lienhart00, Wolf02] outras trabalham directamente sobre os valores da imagem nas regiões identificadas, por exemplo executando uma projecção usando o movimento das regiões no momento anterior e procurando a sua posição no momento seguinte através da minimização da soma das diferenças quadradas, em inglês *Sum of Squared Differences (SSD)* [Li00, Li02, Lienhart02];

2ª Filtragem temporal recursiva – A segunda classe de técnicas faz o seguimento de uma partição previamente obtida recorrendo a um filtro temporal recursivo e considerando mais do que dois instantes de tempo. Os filtros temporais recursivos procuram minimizar a desigualdade entre a partição que está a ser seguida no momento actual e a sua predição por exemplo minimizando o *Mean Square Error* (MSE). Neste tipo de técnicas quando o objecto que está a ser seguido está parcialmente ou totalmente oculto, são utilizados filtros (por exemplo, filtros de Kalman) para efectuar a predição do deslocamento do objecto e assim estimar o seu movimento e a sua posição durante a oclusão. Exemplos deste tipo de técnicas podem ser encontradas em [Broida86, Moscheni96].

A solução específica aplicável em cada caso depende das características das regiões que se pretendem seguir, bem como do movimento esperado para essas regiões. No caso particular de um objecto composto por várias regiões que não são homogéneas nas suas características de movimento, o seu seguimento não deve ser feito usando o mesmo modelo de movimento para todas as regiões que o compõem [Marques97, Zhong98].

Quando é necessário efectuar o seguimento de regiões ou objectos que se ocultam um ao outro, o problema do seguimento pode ser resolvido utilizando um filtro temporal recursivo; em particular, os filtros de Kalman fornecem bons resultados [Gil96, Kruse99].

Para efectuar o seguimento de texto ao longo do tempo em sequências de vídeo nos sistemas estudados foram usualmente utilizadas as técnicas da primeira classe. A escolha deste tipo de técnicas para efectuar o seguimento do texto deve-se essencialmente ao facto deste, para que possa ser lido, estar visível em todas as tramas e surgir sempre em primeiro plano, i.e. o texto normalmente não fica oculto por outros objectos; isto é praticamente sempre para o texto gráfico e também quase sempre verdade para o texto de cena semanticamente relevante. Exemplos da sua aplicação no seguimento de texto podem ser encontrados em [Li00, Lienhart00, Li02, Lienhart02, Wolf02].

2.5 Técnicas de Reconhecimento de Texto

O reconhecimento de texto consiste em atribuir a cada região anteriormente classificada como texto um dado carácter ou seja atribuir a cada região uma classe de um dado alfabeto. As classes correspondem às várias formas que um dado carácter pode assumir dentro de um alfabeto, por exemplo, *a*, *A*, *ā*, *À*, etc. As técnicas que permitem fazer o reconhecimento das regiões classificadas como texto podem dividir-se em duas grandes categorias: técnicas baseadas em métodos de decisão teórica e técnicas baseadas em métodos estruturais [Gonzalez93]. Estas duas categorias de técnicas de reconhecimento serão seguidamente descritas.

2.5.1 Métodos de Decisão Teórica

Estes métodos são utilizados quando a descrição das regiões classificadas como texto é feita numericamente através de um vector de características. Os principais métodos de decisão teórica são brevemente descritos de seguida.

2.5.1.1 Métodos Baseados na Comparação de Características

Estas técnicas baseiam-se em medidas de semelhança associadas a certas características. Estas medidas fazem o cálculo da distância entre o vector de características que descreve a região extraída e classificada como texto e a descrição ou vector de características de cada classe de caracteres. Deste modo, para cada região a reconhecer é determinado um vector de características. Esse vector de características é normalizado e comparado com os vectores correspondentes aos caracteres da base de dados previamente treinada para os vários tipos de fontes. Podem ser utilizadas diferentes medidas de semelhança mas a mais utilizada é a distância Euclidiana [Gonzalez93].

Este tipo de classificador é especificado pelo vector de características de cada classe que é obtido através de um processo de treino. Durante este processo de treino, são utilizados para cada classe conjuntos de padrões de treino conhecidos.

2.5.1.2 Classificadores Estatísticos

A classificação estatística usa uma abordagem probabilística para efectuar o reconhecimento dos caracteres. A ideia é utilizar um esquema de classificação que seja ideal em termos de média. A sua utilização visa minimizar a probabilidade de efectuar falsas classificações [Gonzalez93].

Deste modo, para minimizar a probabilidade de efectuar falsas classificações, é utilizado um classificador de Bayes. Este classificador utiliza as funções de densidade de probabilidade para cada classe de caracteres bem como a probabilidade de ocorrência de cada classe. Assim, dada uma região R anteriormente classificada como texto, descrita pelo seu vector de características, a probabilidade dessa região, R , pertencer a uma dada classe, C , é calculada para todas as classes $C=1, \dots, N$. A região, R , em questão é reconhecida como o carácter correspondente à classe com maior probabilidade.

Para que este método possa ser optimizado, é necessário que as funções de densidade de probabilidade para cada classe de caracteres sejam conhecidas bem como a probabilidade de ocorrência de cada classe. As probabilidades de ocorrência de cada classe são normalmente obtidas assumindo que todas as classes são igualmente prováveis. Assume-se normalmente também que a função densidade de probabilidade para cada classe tem uma distribuição normal. Estes pressupostos são, na realidade, uma aproximação ao classificador de Bayes.

O classificador de Bayes para classes Gaussianas é especificado pelo vector de características e pela matriz de co-variância de cada classe. Estes parâmetros utilizados para especificar o classificador são obtidos através de um processo de treino. Durante o processo de treino para calcular os parâmetros e os descritores são utilizados, para cada classe, conjuntos de padrões de treino.

2.5.1.3 Redes Neurais

Para efectuar o reconhecimento de caracteres utilizando redes neuronais, os vectores de características correspondentes às regiões a ser classificadas são usados como entradas para as redes. Estas redes são previamente treinadas. O peso de cada nó da rede é ajustado durante o

seu treino de acordo com os valores/caracteres de saída desejados. Assim, depois de treinada a rede, efectua-se o reconhecimento dos caracteres, utilizando para tal os parâmetros também usados durante a fase de treino. A vantagem da utilização de redes neuronais em sistemas OCR resulta da sua natureza adaptativa, i.e. da possibilidade de treinar a rede à medida que esta é utilizada [Gonzalez93].

2.5.2 Métodos Estruturais

Os métodos estruturais de reconhecimento são utilizados quando a descrição das regiões classificadas como texto é feita através de parâmetros descritores da forma. Existem essencialmente dois tipos de métodos: emparelhamento de contornos (*matching shape numbers*) e métodos sintácticos [Gonzalez93]. Todavia, no contexto do reconhecimento de texto, os métodos de emparelhamento de contornos são pouco utilizados [Eikvil93]. Os métodos estruturais são brevemente descritos de seguida.

2.5.2.1 Emparelhamento de Contornos

Este método consiste na comparação de regiões cuja forma é descrita com base no seu contorno utilizando para tal o método de cadeias de códigos onde o contorno é representado por uma sequência de linhas rectas de determinado tamanho e direcção; tipicamente são utilizados 4 ou 8 tipos de linhas. A direcção desses segmentos é codificada utilizando um esquema numérico idêntico ao apresentado na Figura 2.14 .

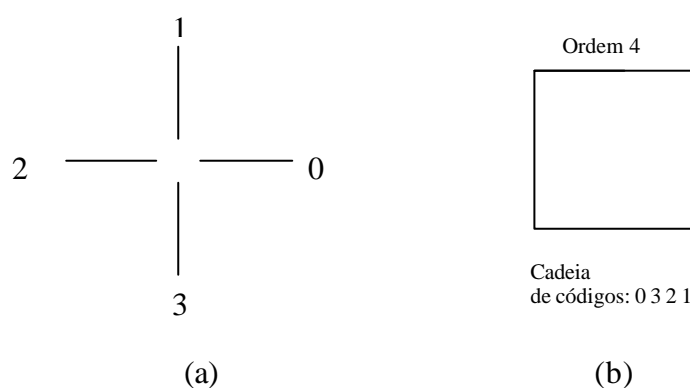


Figura 2.14 – Exemplo de uma cadeia de quatro códigos: (a) 4 linhas de código direccionais e (b) representação de uma região por uma cadeia de códigos [Gonzalez93].

O grau de semelhança, k , entre duas regiões A e B , descritas com base no seu contorno por uma cadeia de códigos, é tanto maior quanto maior for o número de códigos iguais que descrevem os seus contornos. A distância entre dois contornos A e B é definida como o inverso do seu grau de semelhança [Gonzalez93]:

$$D(A, B) = \frac{1}{k}$$

Deste modo, k pode variar entre 0 e infinito correspondendo, respectivamente, à não existência de coincidência ou à coincidência perfeita entre os contornos.

2.5.2.2 Métodos Sintácticos

Estes métodos exigem a definição de medidas de semelhança, através de conceitos matemáticos, baseadas nas relações existentes entre os parâmetros geométricos das regiões. A ideia é cada classe possuir uma definição gramatical da composição de cada carácter. A definição gramatical pode ser representada por cadeias ou árvores de decisão. Para se atribuir uma classe a um carácter desconhecido, são extraídos os seus parâmetros geométricos e é efectuada a sua comparação com as definições gramaticais de cada uma das classes. A classe que apresentar maior semelhança com o carácter/região a reconhecer é a classe escolhida, determinando-se o carácter reconhecido[Gonzalez93].

2.6 Sistemas de Extracção de Texto mais Relevantes

Nos últimos anos, vários têm sido os sistemas apresentados com o intuito de resolver o problema da extracção automática de texto em imagens e vídeos digitais. Bons exemplos destes sistemas podem ser encontrados em [Fletcher88, Ohya94, Lienhart95, Zhong95, Etemad97, Jain98, Li98, Messelodi99, Wu99, Li00, Lienhart00, Zhong00, Crandall01, Gu01 Hase01, Lienhart02, Zhang02].

Nas secções seguintes, será efectuada a descrição de quatro sistemas de extracção automática de texto, quer em imagens, quer em sequências de vídeo digital. A sua apresentação será efectuada por ordem crescente de complexidade, iniciando-se com a descrição dos sistemas que só são aplicáveis a imagens, para terminar com os sistemas que são aplicáveis quer a imagens, quer a vídeo. Estes sistemas foram escolhidos por serem relativamente recentes e por apresentarem bons desempenhos. Para além disso, são, também, representativos dos vários tipos de abordagens técnicas à extracção de texto em imagens ou sequências de vídeo. O primeiro e o segundo sistemas apresentados ilustram abordagens onde as imagens ou tramas são segmentadas em regiões conexas e a classificação dessas regiões como texto e não texto é feita com recurso à análise geométrica; o terceiro e o quarto sistemas ilustram a utilização de redes neuronais para efectuar a detecção do texto..

2.6.1 Extracção de Texto Gráfico e de Cena em Imagens

Em [Messelodi99] é apresentado um método para a extracção automática de linhas de texto em imagens baseado no pressuposto que o texto numa imagem surge como um conjunto de regiões que podem ser agrupadas e alinhadas. A escolha deste sistema de extracção de texto em imagens deveu-se ao facto de nele se considerar também o texto inclinado no processo de extracção de texto. O método proposto pode extrair texto gráfico e/ou de cena com as seguintes características:

- Caracteres de diferentes tamanhos, fontes e estilos;
- Caracteres orientados em qualquer direcção;

- Caracteres que façam parte da mesma palavra devem possuir todos a mesma cor;
- Caracteres com limitações em termos de tamanho: um caracter não pode ser maior nem menor que um dado número de *pixels* pré-definido.

O sistema de extracção de texto em imagens descrito nesta secção consiste em quatro etapas [Messelodi99]: segmentação, classificação de regiões, selecção de linhas de texto e reconhecimento de texto, como se ilustra na Figura 2.15.

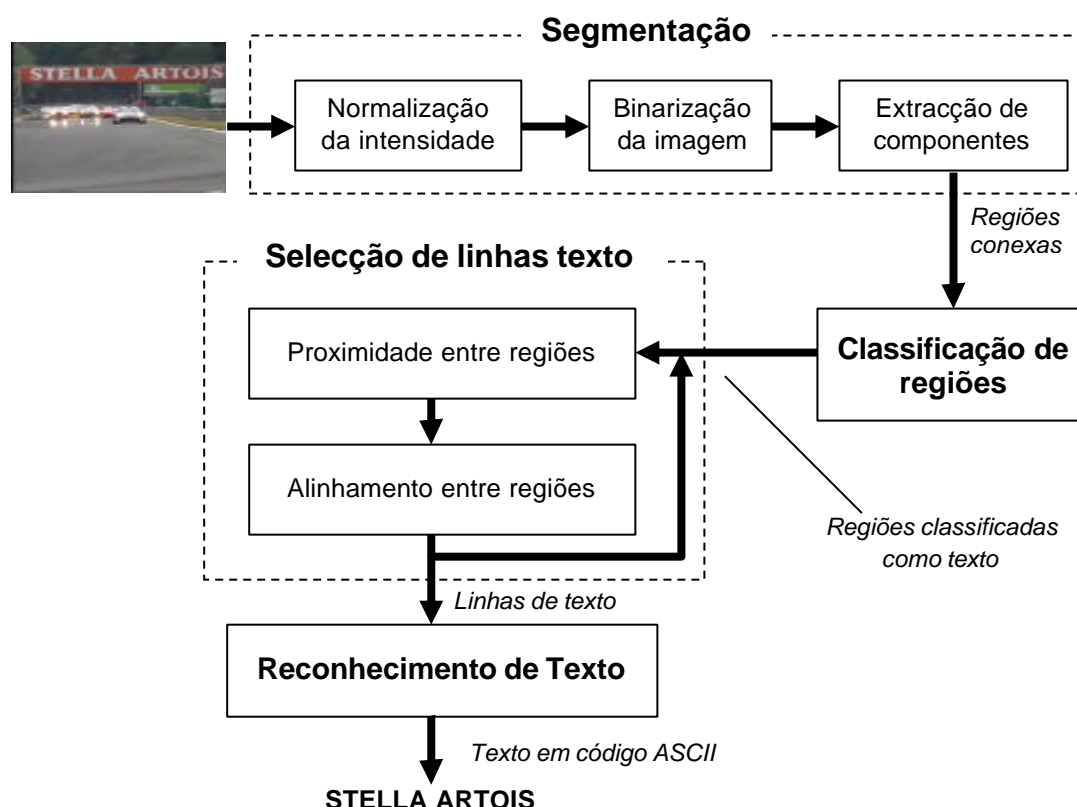


Figura 2.15 – Arquitectura do sistema de extracção de texto em imagens proposto em [Messelodi99].

As quatro fases pelas quais passa a extracção de texto proposta são descritas nas secções seguintes.

2.6.1.1 Segmentação

Nesta etapa, a imagem inicial é segmentada de forma a extrair regiões conexas. Estas regiões são definidas como componentes conexas, homogéneos em termos da sua luminância, extraídos de uma ou mais imagens resultantes do pré-processamento apropriado da imagem inicial. A fase de pré-processamento é concebida de forma a que os caracteres surjam como regiões homogéneas. Este processo decorre em três fases:

- **Normalização da intensidade da luminância** – A normalização da intensidade da luminância é necessária para reduzir os efeitos indesejáveis na imagem provocados pela existência de eventuais gradientes de iluminação ou de ruído. Esta normalização é conseguida dividindo o valor da intensidade da luminância de cada *pixel* pela média da intensidade calculada para os *pixels* na sua vizinhança. A dimensão, w , da janela sobre a qual se calcula a média da intensidade da luminância é um parâmetro importante, uma vez que o desempenho da normalização da intensidade da luminância depende da relação entre w e a espessura dos caracteres. O melhor valor para w corresponde ao valor da espessura dos caracteres. Todavia, na presença de caracteres com espessuras diferentes na mesma imagem, o desempenho da normalização de intensidade diminui. Na Figura 2.16 ilustra-se o efeito da normalização da intensidade da luminância;

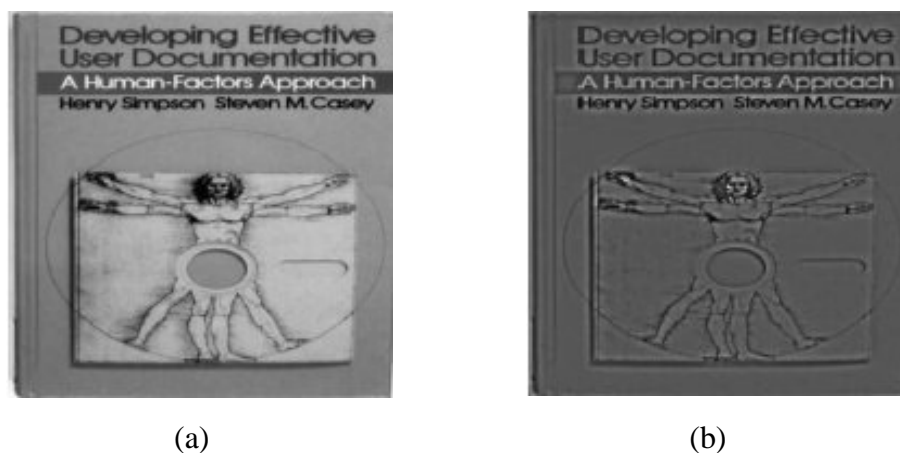


Figura 2.16 – Exemplo do efeito da normalização da intensidade da luminância: (a) imagem original e (b) resultado da normalização utilizando uma janela com um tamanho de 13 *pixels* da imagem em (a) [Messelodi99].

- **Binarização da imagem** – Ao contrário do que ocorre usualmente nos documentos onde os caracteres são impressos a preto sobre um fundo branco, nas imagens o texto pode surgir com contraste positivo ou negativo em relação ao fundo, i.e. texto normal ou inverso consoante a variação relativa da luminância entre o texto e o fundo. Nesta fase, são gerados dois mapas de bits obtidos através da aplicação à imagem com a intensidade normalizada de dois limiares globais: com o primeiro limiar capturam-se os *pixels* com valores de luminância elevados e que correspondem ao texto normal enquanto que com o segundo limiar se capturam os *pixels* com valores de luminância baixos e que correspondem ao texto inverso. Na Figura 2.17 é apresentado o resultado da binarização aplicada à imagem depois de efectuada a sua normalização da intensidade da luminância;

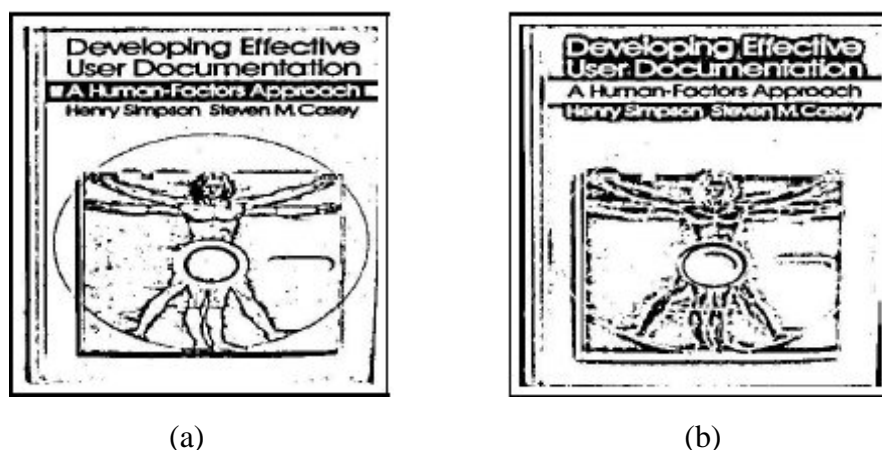


Figura 2.17 – Resultado da binarização aplicada à imagem anteriormente normalizada, Figura 2.16: (a) imagem com texto normal e (b) imagem com texto inverso [Messelodi99].

- **Extracção dos componentes conexos** – Nesta fase, são identificados os componentes conexos das duas imagens binárias geradas na fase anterior, uma correspondente ao texto normal e a outra ao texto inverso. Os componentes conexos das duas imagens binárias constituem as regiões para as fases seguintes do algoritmo.

Os componentes conexos gerados para cada uma das duas imagens binárias na fase anterior (ver Figura 2.17) são processados independentemente nas fases seguintes, classificação de regiões e selecção das linhas de texto. Isto equivale a assumir que cada linha de texto é escrita com um único tipo de contraste, positivo ou negativo.

2.6.1.2 Classificação de Regiões

Nesta etapa, cada componente conexo representa uma região caracterizada por um contraste acentuado relativamente às regiões que lhe são vizinhas. Considera-se que o conjunto das regiões contém todos os componentes de texto misturados com aqueles que não são texto. A filtragem das regiões de texto é conseguida através da aplicação de regras heurísticas que permitem classificar cada uma delas como texto ou não texto. Estas regras baseiam-se em características das regiões conexas tais como:

- **Área** – As regiões com uma área inferior a 12 *pixels* são descartados, uma vez que são considerados ruído;
- **Altura** – Regiões com uma altura da *bounding box* superior a 50% da altura da imagem são descartadas;
- **Largura** – Regiões com uma largura da *bounding box* superior a 50% da largura da imagem são descartadas;
- **Proximidade** – A proximidade é definida como a distância mínima entre o limite da região em análise e a borda da imagem. A proximidade pretende identificar regiões originadas essencialmente por ruído, existentes ao longo da borda da imagem. Assim,

regiões com uma proximidade em relação à borda da imagem inferior a 0.02 da maior dimensão da imagem (altura e largura) são eliminadas;

- **Excentricidade** – Regiões com uma excentricidade² inferior a 0,09 são descartadas;
- **Solidez** – Regiões com uma solidez³ inferior a 0,2 são descartadas;
- **Contraste** – Por razões de legibilidade, o texto é frequentemente caracterizado por elevado contraste; isto sugere a filtragem de regiões caracterizadas por baixo contraste. Assim, as regiões com um valor de contraste inferior a 10 são descartadas (a escala para o contraste varia de 0 a 255).

O resultado da aplicação dos filtros heurísticos aplicados às duas imagens binárias resultantes da segmentação (Figura 2.17) é ilustrado na Figura 2.18.

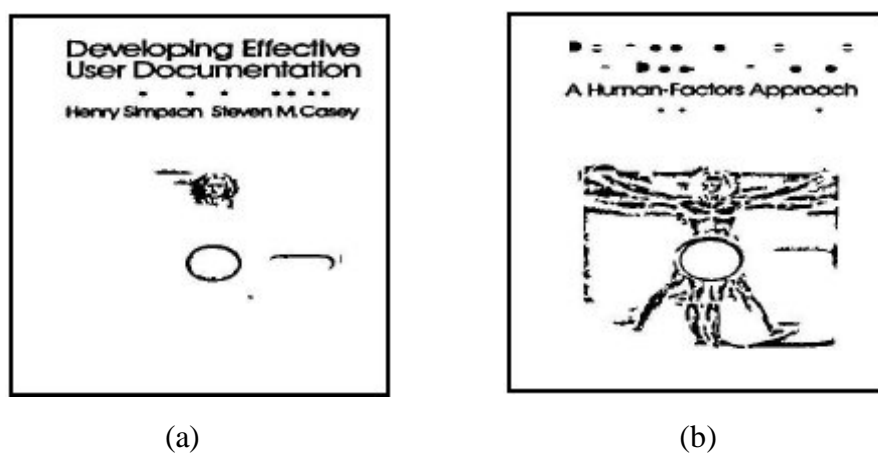


Figura 2.18 – Resultado da aplicação dos filtros heurísticos às imagens binárias resultantes da segmentação (Figura 2.17): (a) imagem com texto normal e (b) imagem com texto inverso [Messelodi99].

As regiões que não verificam simultaneamente todas as condições supracitadas são classificados como não texto e descartadas.

² A excentricidade de uma região é definida como a relação entre o seu raio máximo, R_{max} , e o seu raio mínimo, R_{min} : $E = \frac{R_{max}}{R_{min}}$.

³ A solidez de uma região exprime a semelhança entre a área da forma convexa da região e a área da mesma região ou seja $S = \frac{A}{A_{cv}}$.

2.6.1.3 Selecção de Linhas de Texto

Nesta etapa pretende-se efectuar o agrupamento das regiões que passaram na filtragem da fase anterior de modo a formar grupos de regiões candidatos a linhas de texto. No processo de agrupamento das regiões classificadas como texto, é utilizado um método que consiste na divisão sucessiva do conjunto formado por todas as regiões que passaram na filtragem em subconjuntos de regiões que cumpram os critérios para a formação de linhas de texto. O processo de divisão termina quando for atingido um critério de paragem, i.e. o subconjunto possui uma única linha de texto ou não possui texto.

Depois de cada divisão e antes da divisão seguinte, é determinado o estado de cada subconjunto. O estado de cada subconjunto determina se a sua divisão terminou ou continua. Para tal, a cada subconjunto são aplicados critérios que fazem uso de condições heurísticas. Estas condições permitem ao sistema determinar o estado de cada subconjunto:

- *Mais_linhas* – O subconjunto contém mais do que uma linha de texto e logo deve continuar a ser dividido, uma vez que não satisfaz nenhum dos critérios de paragem;
- *Uma_linha* – O subconjunto contém uma única linha de texto. A divisão do subconjunto termina uma vez que o mesmo satisfaz o primeiro critério de paragem;
- *Não_texto* – O subconjunto não contém texto e logo a divisão do subconjunto termina uma vez que o mesmo satisfaz o segundo critério de paragem.

Para determinar o estado do subconjunto, são utilizadas as seguintes regras heurísticas:

- **Número de regiões no subconjunto** – É assumido que as palavras ou linhas de texto possuem mais de dois caracteres; assim, se o subconjunto possuir uma única região é marcado como *não_texto*;
- **Distância entre regiões** – A distância entre duas regiões contíguas tem de ser inferior a um determinado valor de limiar Th_{dist} . Se esta distância for superior a Th_{dist} , o subconjunto é marcado com *mais_linhas*. Neste sistema foi utilizado $Th_{dist}=30\text{ pixels}$;
- **Altura do subconjunto** – É assumido que os caracteres devem possuir um tamanho mínimo para que sejam legíveis. Assim, são definidos dois limiares para a altura do subconjunto, Th_{h_min} e Th_{h_max} . Se a altura de todas as regiões do subconjunto for inferior a Th_{h_min} , o subconjunto é marcado como *não_texto*. Se for superior a Th_{h_max} , a probabilidade de o conjunto possuir mais do que uma linha de texto é elevada; como tal, é marcado com *mais_linhas*. Na avaliação de desempenho, foi utilizado $Th_{h_min}=10\text{ pixels}$ e $Th_{h_max}=15$;
- **Relação entre as alturas das regiões e a altura média do subconjunto** – Se a maioria da altura das regiões for semelhante à altura média do subconjunto, este é marcado como *uma_linha*; caso contrário, é marcado como *mais_linhas*;
- **Alinhamento** – Se todas as regiões que formam o subconjunto estiverem alinhadas segundo uma dada direcção, este é marcado como *uma_linha*; caso contrário, assume-se que o subconjunto possui mais do que uma linha de texto e logo é marcado como *mais_linhas*.

O processo de divisão decorre em dois passos, correspondendo cada passo à aplicação de uma condição de divisão. Estas são aplicadas segundo uma ordem pré-definida: proximidade e alinhamento.

- 1º **Proximidade** – Neste passo é aplicada a primeira condição para a formação de linhas, ou seja, é verificada a proximidade entre regiões. O objectivo é dividir o conjunto de regiões inicial em subconjuntos que cumpram o critério de proximidade entre as suas regiões. A proximidade entre as regiões de um subconjunto tem que ser inferior a um determinado valor de limiar, Th_{dist} , previamente definido. Neste sistema foi utilizado um valor de $Th_{dist}=30\ pixels$. Este valor foi calculado empiricamente com base no tamanho médio dos caracteres existentes no conjunto de imagens de teste. O resultado da divisão do conjunto inicial em subconjuntos através da condição de proximidade, com $Th_{dist}=30\ pixels$, é ilustrado na Figura 2.19. Na imagem (b) são ilustrados os três subconjuntos, realçados a cores diferentes, resultantes da divisão da imagem baseada na proximidade.

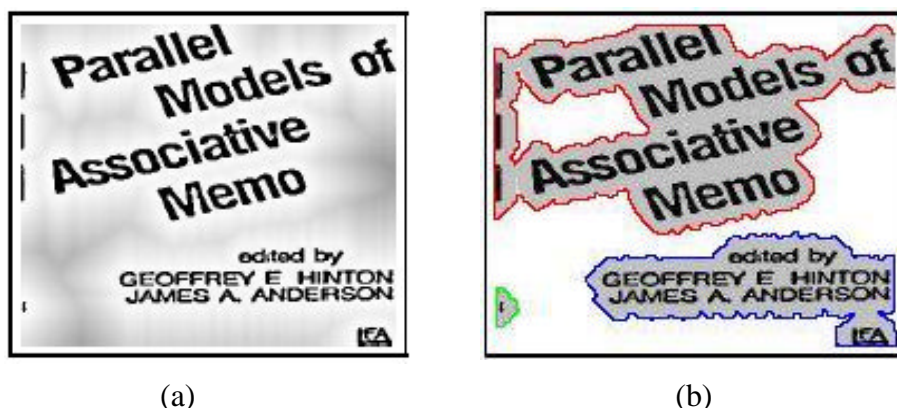


Figura 2.19 – Exemplo do efeito da formação de linhas utilizando a primeira condição, i.e. a proximidade entre regiões, com $Th_{dist}=30\ pixels$: (a) imagem com o conjunto inicial de regiões classificadas como texto e (b) imagem com os três subconjuntos resultantes da divisão da imagem em (a) realçados a cores diferentes [Messelodi99].

De seguida, para cada um dos subconjuntos resultantes da divisão do conjunto inicial é determinado o seu estado através da aplicação das regras heurísticas descritas anteriormente. Os subconjuntos cujo estado for *mais_linhas* e *uma_linha* passam ao passo seguinte;

- 2º **Alinhamento** – Neste passo é aplicada a segunda condição para a formação de linhas, ou seja, é verificado o alinhamento entre regiões. Cada subconjunto marcado com *mais_linhas* no passo anterior é dividido em novos subconjuntos cujas regiões se encontrem alinhadas segundo uma dada direcção. Aos subconjuntos marcados com *uma_linha* é simplesmente determinada a sua orientação. Uma vez concluída a divisão, determina-se o estado de cada um dos novos subconjuntos resultantes.

Para aplicar a condição de alinhamento entre as regiões, a ideia é dividir os caracteres do subconjunto através de linhas rectas com um declive semelhante à direcção do subconjunto, sem que estas intersectem qualquer caracter. Assim, conseguem-se formar partições que dependem da direcção do subconjunto. Para tal, são utilizados os centros

dos caracteres os quais correspondem aos centros das *bounding boxes* desses caracteres. O cálculo da direcção do subconjunto é conseguido em 4 etapas:

- 1^a **Etapa** – Nesta etapa é determinado um conjunto de direcções candidatas através do cálculo do histograma da luminância segundo os vários declives associados às rectas que passam nos centros de todos os pares de caracteres, ver Figura 2.20 (c);
- 2^a **Etapa** – Nesta etapa é calculado um valor de limiar $Th_{dir}=f \times max_hist$, onde f é um factor previamente definido (foi utilizado $f=0.9$) e max_hist é o valor máximo do histograma calculado na fase anterior;
- 3^a **Etapa** – Nesta etapa são seleccionadas as regiões candidatas. Como regiões candidatas são seleccionadas aquelas que possuem um declive ao qual corresponde um valor no histograma superior ao valor de limiar Th_{dir} calculado na etapa anterior;
- 4^a **Etapa** – Nesta etapa é efectuada a projecção do subconjunto de caracteres segundo cada direcção candidata. A direcção do subconjunto corresponde à direcção sobre a qual existem mais ocorrências de caracteres, ver Figura 2.20 (b).

Na Figura 2.20 ilustra-se o processo de determinação da direcção do subconjunto de caracteres: na Figura 2.20 (a) está representado o subconjunto de caracteres (cinzento) e os seus centros (pontos pretos); na Figura 2.20 (b) é representado o histograma calculado segundo os vários declives associados às rectas que passam nos centros de todos os pares de caracteres; por último, na Figura 2.20 (c) é representada a projecção do histograma, sendo possível observar que no caso ilustrado o número máximo de ocorrências de caracteres dá-se segundo a direcção da recta com um declive de 105° .

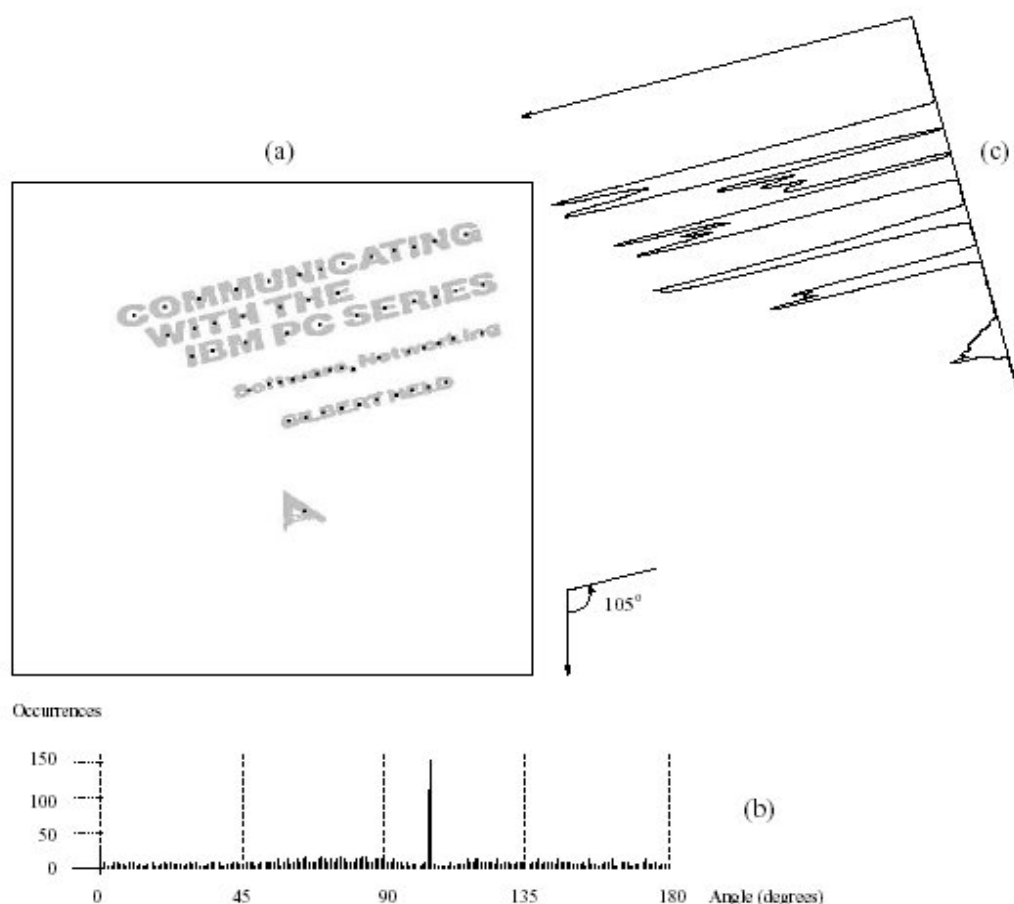


Figura 2.20 – Exemplo do cálculo da direcção de subconjunto: (a) subconjunto de caracteres (cinzento) e os seus centros (pontos pretos); (b) histograma calculado segundo o declive dos segmentos de recta formados entre cada par de centros e (c) projecção do histograma [Messelodi99].

Uma vez concluída a divisão para cada um dos novos subconjuntos resultantes, é determinado o seu estado.

Se no final dos dois passos acima descritos ainda persistirem subconjuntos classificados com o estado *mais_linhas*, o processo repete-se sucessivamente para cada um destes subconjuntos até que não restem subconjuntos marcados com o estado *mais_linhas*.

No final do processo de divisão, isto é quando não restarem mais subconjuntos com possibilidade de serem divididos, os subconjuntos marcados com *uma_linha* identificam o conjunto de prováveis linhas de texto; os restantes subconjuntos são eliminados.

2.6.1.4 Reconhecimento de Texto

Depois de terminada a formação das linhas de texto e das regiões classificadas como não texto terem sido descartadas, é aplicado ao mapa de bits que contém os subconjuntos marcados com

uma_linha um sistema OCR comercial que reconhece o texto existente nos mesmos e o converte para código ASCII.

2.6.1.5 Avaliação do Desempenho

O algoritmo anteriormente apresentado foi testado com um conjunto de 100 imagens monocromáticas com uma resolução de 512×512 *pixels*, obtidas a partir da digitalização de 100 capas de livros [Messelodi99]. Três tipos de avaliação foram efectuados:

1º Avaliação da classificação de regiões – A avaliação da classificação das regiões pretende avaliar o desempenho da fase de classificação das regiões efectuada através da sua filtragem usando condições heurísticas. O algoritmo extraiu 180327 regiões conexas das 100 imagens, das quais apenas 8% (14384) foram classificadas como texto através da filtragem com condições heurísticas. O desempenho de cada condição heurística é apresentado na Tabela 2.1.

Tabela 2.1 – Desempenho em termos da classificação das regiões para as várias condições heurísticas.

Condição Heurística	Número de Regiões Filtradas (%)	Número de Regiões Erradamente Eliminadas
Área	145129 (87.44%)	2
Excentricidade	569 (0.34%)	8
Solidez	118 (0.7%)	1
Tamanho relativo (altura e largura)	265 (0.16%)	0
Proximidade	3492 (2.1%)	4
Contraste	16406 (9.88%)	140
Total	165979 (100.0%)	155

A maioria das regiões conexas foram removidas devido ao seu pequeno tamanho ou devido ao seu baixo contraste em relação ao fundo da imagem na sua vizinhança. Para determinar a precisão da classificação das regiões, todas as regiões que foram eliminadas foram analisadas manualmente através da observação visual para determinar se correspondiam a caracteres falsamente classificados ou não. Foram eliminados erradamente 155 caracteres o que representa menos de 1 carácter por cada 1000;

2º Avaliação da determinação do ângulo de inclinação – A avaliação da determinação do ângulo de inclinação pretende avaliar o desempenho do algoritmo na determinação do ângulo de inclinação das linhas de texto. Para tal, foram seleccionadas 321 linhas de texto das 100 imagens; estas linhas foram rodadas manualmente para inclinações compreendidas entre $[-45^\circ, 45^\circ]$. O erro médio obtido para a inclinação das linhas de texto foi de 0.37° e o erro máximo foi de 2.8° . Os maiores erros derivam essencialmente das linhas de texto curtas ou das linhas que terminam com um carácter maior ou menor do que a média da altura da linha;

3º Avaliação da selecção de linhas – A avaliação da selecção de linhas pretende avaliar o desempenho da fase de selecção das linhas de texto. De modo a avaliar este desempenho, foram extraídas manualmente todas as linhas de texto das 100 imagens, num total de 432 linhas. Os resultados obtidos para a avaliação da selecção de linhas de texto são apresentados na Tabela 2.2. São consideradas linhas correctamente detectadas aquelas onde faltam poucos caracteres ou foram adicionadas poucas regiões falsamente classificadas como texto, 7 e 28 por linha, respectivamente.

Tabela 2.2 – Desempenho em termos da selecção de linhas.

	Detectadas	Falhas de Detecção	Total
Linhas de texto	394	38	432
Linhas de texto falsamente detectadas	335	-	335
Total	729	38	767

Segundo os autores, foram detectadas 91% das linhas de texto; a precisão da detecção, i.e. a relação entre o número de linhas de texto correctamente detectadas e o número de linhas de texto detectadas, foi da ordem dos 54%. O autor justifica este resultado em termos de precisão com a estratégia utilizada na fase de classificação das regiões como texto e não texto, na qual foi dada preferência à detecção de todo o texto ainda que para tal sejam classificadas erradamente como texto muitas regiões que não correspondem a texto.

Em [Messelodi99] não foram apresentados resultados respeitantes ao reconhecimento por parte de sistemas OCR.

O sistema proposto em [Messelodi99] e aqui descrito apresenta como ponto forte a sua capacidade para identificar texto constituído por linhas orientadas em qualquer direcção. O texto pode ser de cena ou gráfico e pode ainda caracterizar-se por diferentes tamanhos e fontes. Este sistema só funciona para sequências de vídeo tomadas como uma sequência de imagens independentes não explorando a redundância temporal existente no vídeo.

2.6.2 Extracção Automática de Texto Gráfico para Indexação de Vídeo

Em [Lienhart00] é apresentado um método para extracção automática de texto em sequências de vídeo digital, incluindo o seu seguimento no tempo. O texto detectado é directamente passado a um sistema OCR comercial que é responsável pelo seu reconhecimento e conversão para código ASCII. Este texto é utilizado para efectuar a indexação e a pesquisa de vídeo.

Este sistema foi escolhido por ser um bom exemplo da extracção de texto em vídeo baseado em componentes conexos, i.e. detecta as regiões de texto através da extracção de regiões homogéneas em termos de cor ou níveis de cinzento; estas regiões obedecem a determinadas características, por exemplo em termos de tamanho, forma e alinhamento espacial. Para além disso, este sistema é um bom exemplo da utilização da extracção automática de texto para efectuar a indexação de vídeo.

Este método de extracção destina-se a texto gráfico ao qual são, todavia, impostas as seguintes restrições:

- Os caracteres devem encontrar-se em primeiro plano e nunca parcialmente ocultos;
- Os caracteres pertencentes a uma palavra devem possuir todos a mesma cor;
- Os caracteres devem estar alinhados segundo a direcção horizontal;
- Os caracteres não devem alterar a sua forma, tamanho, cor e orientação de trama para trama;
- Os caracteres têm limitações em termos de tamanho: um caracter não pode ser nem maior nem menor que um dado número de *pixels* pré-definido;
- Os caracteres devem ser estacionários ou ter movimentos lineares, i.e. movimentos rectilíneos com velocidades constantes; os movimentos lineares devem ser horizontais ou verticais;
- Os caracteres devem permanecer na imagem durante várias tramas consecutivas.

O método proposto em [Lienhart00] engloba quatro fases principais: detecção, seguimento, reconhecimento do texto e indexação/recuperação como ilustrado na Figura 2.21.

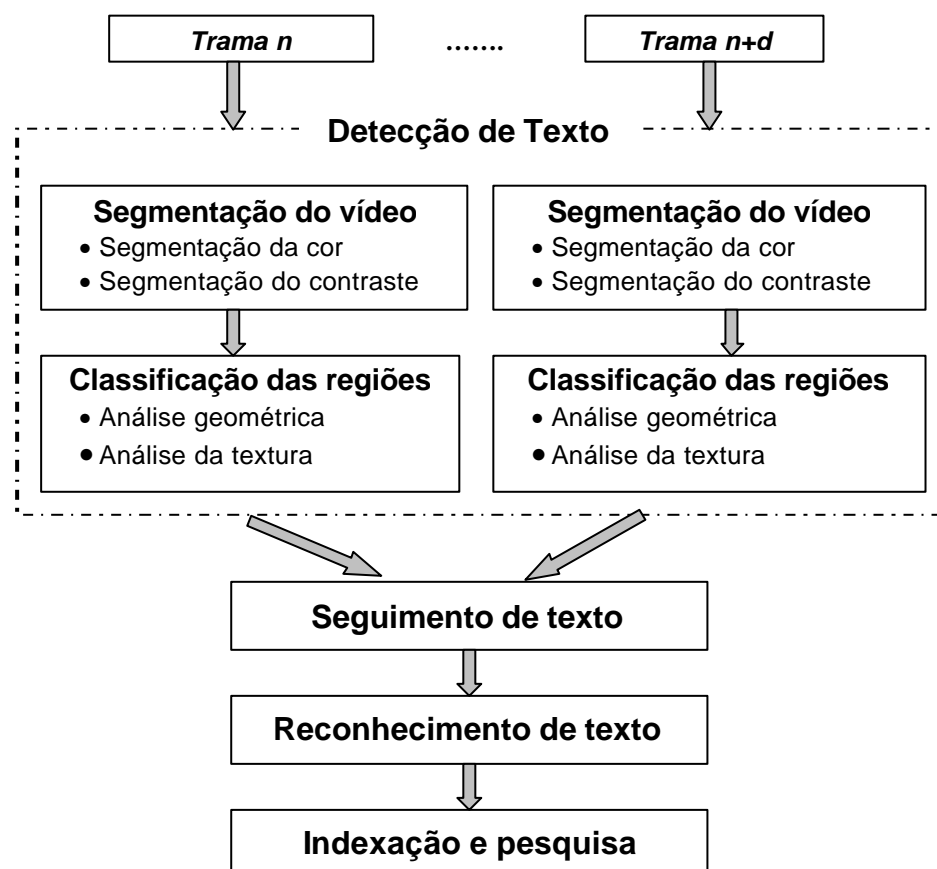


Figura 2.21 – Arquitectura do sistema de extracção de texto gráfico em sequências de vídeo proposto em [Lienhart00].

As quatro fases pelas quais passa a extracção de texto para o algoritmo aqui descrito são apresentadas nas secções seguintes.

2.6.2.1 Detecção de Texto

A detecção do texto existente em cada trama de vídeo decorre em duas fases: segmentação da trama e classificação das regiões.

1ª Fase – Segmentação da trama

A primeira acção a tomar sobre uma trama de vídeo consiste na sua segmentação em regiões apropriadas. Esta segmentação é levada a cabo em duas etapas, como ilustrado no esquema da Figura 2.21: segmentação da cor e segmentação do contraste:

- **Segmentação da cor** – Nesta etapa, as características de cromaticidade dos caracteres são utilizadas para definir os critérios de separação das várias regiões. Começa-se por efectuar uma sobre-segmentação (ou seja uma segmentação com demasiadas regiões) para cada trama da sequência de vídeo, utilizando-se um algoritmo baseado na técnica *region-growing* [Zucker76]. O valor do limiar de decisão para a distância da cor é seleccionado de forma a impedir que os caracteres se agrupem facilmente com os seus vizinhos. Seguidamente, as várias regiões são agrupadas segundo um critério de homogeneidade de forma a remover a sobre-segmentação e ao mesmo tempo a evitar a sub-segmentação (ou seja uma segmentação com poucas regiões). De modo a localizar e, eliminar pequenas regiões situadas sobre as zonas de fronteira entre as regiões segmentadas, é combinada a detecção de fronteiras com a orientação local da variação do contraste. Para localizar as fronteiras foi utilizado o detector de Canny [Canny86] e para calcular a orientação da variação local do contraste foi utilizado um tensor de inércia [Jähne97]. Todavia, o autor não descreve como é feita esta combinação de modo a eliminar as pequenas regiões entre as fronteiras em [Lienhart00]. O processo de segmentação da cor termina com o fim do agrupamento das regiões com cores semelhantes, ver Figura 2.22 (b);



(a)



(b)

Figura 2.22 – Exemplo da segmentação da cor utilizando a técnica *region-growing*: (a) trama original; (b) resultado da segmentação da cor [Lienhart00].

- **Segmentação do contraste** – Nesta etapa, procura-se tirar benefício do elevado contraste tipicamente existente entre o texto gráfico e os *pixels* que o circundam. Para cada trama de vídeo é gerada uma imagem binária com base no contraste, calculando o contraste local para cada posição $I(x,y)$ através da expressão (2.1):

$$Cont_{local}(x, y) = \sum_{k=-r}^r \sum_{l=-r}^r G_{k,l} \cdot |I_{x,y} - I_{x-k,y-l}| \quad (2.1)$$

Onde $G_{k,l}$ é um filtro 2D de Gauss, r é o tamanho da vizinhança local usada para o cálculo e $| |$ representa o valor absoluto da diferença entre os valores das luminâncias para o *pixel* em análise, antes e depois de filtrado.

O contraste local é, tipicamente, representado através de uma imagem binária onde as regiões com contraste elevado são representadas a branco e as regiões com baixo contraste são representadas a preto. De seguida, a imagem binária de contraste é dilatada com o objectivo de garantir com elevada probabilidade que inclui todas as regiões de elevado contraste existentes na imagem, ver Figura 2.23.

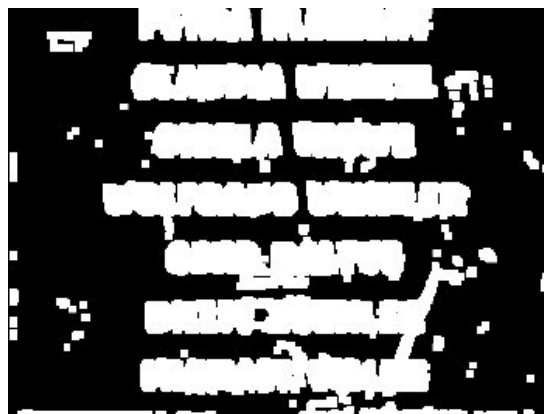


Figura 2.23 – Exemplo da imagem binária de contraste dilatada [Lienhart00].

Finalmente, as regiões na imagem de cor segmentada na etapa anterior que não sejam sobrepostas em mais de 80% da sua área pela parte branca da imagem binária do contraste dilatada são descartadas por se considerar não corresponderem a caracteres. O valor de 80% foi determinado empiricamente com base em conteúdos com texto gráfico onde a espessura dos caracteres não é muito grande. A segmentação do contraste em conteúdos com texto de espessuras elevadas torna-se ineficaz.

2ª Fase – Classificação das regiões

Uma vez segmentado o vídeo, a fase seguinte consiste na classificação das várias regiões provenientes da segmentação. Essa classificação consiste em duas etapas, como ilustrado na Figura 2.21:

- **Análise geométrica** – Nesta etapa, as várias regiões resultantes da fase anterior são filtradas com base nas seguintes restrições geométricas:

- ♦ Altura $\in [4, 90]$ pixels;
- ♦ Comprimento $\in [1, 120]$ pixels;
- ♦ Relação comprimento/altura $\in [0.4, 7]$;
- ♦ Solidez $\in [0.15, 1]$.

Os valores específicos nestas restrições dependem do tamanho mínimo e máximo dos caracteres que se pretendem detectar. As regiões que não cumpram as restrições geométricas supracitadas são descartadas. O resultado da segmentação depois de aplicada a análise geométrica é ilustrado na Figura 2.24.



Figura 2.24 – Segmentação depois da análise geométrica (246 regiões) [Lienhart00].

- **Análise da textura** – Nesta etapa, pequenas regiões de não texto, mas ainda no processo por não terem sido eliminadas devido ao seu tamanho ou forma serão eliminadas através da análise da textura. Esta análise procura explorar o facto de, ao longo de uma linha de texto ou de uma palavra, o contraste variar de uma forma periódica. Esta periodicidade é típica em palavras ou linhas de texto, podendo ser considerada como uma característica da textura do texto. Assim, sobre um fundo mais ou menos uniforme, podem ser observadas flutuações periódicas de contraste devidas ao texto. Na maioria dos casos, estas flutuações do contraste também podem ser observadas na presença de texto sobreposto sobre fundos complexos, desde que o texto esteja rodeado por uma aura. Esta aura é usualmente adicionada durante a produção do vídeo para melhorar a legibilidade do texto, ver Figura 2.25.



Figura 2.25 – Exemplo de texto rodeado por uma aura para melhorar a sua legibilidade [Lienhart00].

O método proposto em [Lienhart00] explora o facto do texto variar de forma periódica ao longo de uma palavra ou de uma linha de texto sendo então possível distinguir as fronteiras entre regiões de texto e não texto, usando o seguinte processo:

- ◆ Inicialmente, todas as regiões classificadas como caracteres pela análise geométrica, com o mesmo tamanho e determinada proximidade umas em relação às outras, são agrupadas para formar conjuntos de possíveis palavras;
- ◆ Seguidamente, é estimada a direcção das possíveis palavras através do cálculo da direcção do seu eixo principal. Esta direcção é definida através do ângulo θ entre o eixo dos xx e o eixo em torno do qual a palavra pode sofrer rotação com uma inércia mínima, ver Figura 2.26.



Figura 2.26 – Exemplo da direcção da escrita, correspondente à linha vermelha [Lienhart00].

De acordo com [Jähne97] esta direcção é dada por:

$$\theta = \frac{1}{2} \arctan \frac{2m_{1,1}}{m_{2,0} - m_{0,2}} \quad (2.2)$$

sendo os momentos dados por:

$$m_{p,q} = \sum_{x_1, x_2} (x_1, \mathbf{m}_{x_1})^p (x_2, \mathbf{m}_{x_2})^q \quad (2.3)$$

- ◆ Por último, a textura é analisada. Para extrair as características que permitam fazer a análise da textura das palavras, é necessário definir os seguintes parâmetros:
 - N_{\min} – Número de caracteres existentes na direcção da escrita, na vizinhança de cada caracter; no método proposto utilizou-se $N_{\min}=2$;
 - C_{\maxDist} – Comprimento máximo medido na horizontal que cada caracter pode ocupar; no método proposto utilizou-se $C_{\maxDist} = 1,5 \times$ (comprimento máxima estimado para o caracter).

No método proposto, considera-se que as flutuações de contraste são caracterizadas pela existência de fronteiras. Deste modo, através da contagem do número de fronteiras existentes na direcção da escrita consegue identificar-se a presença de caracteres ou não. Assim, assume-se que se está na presença de texto quando se têm pelo menos $2N_{\min}$ fronteiras num comprimento $2C_{\maxDist}$. Para detectar as fronteiras existentes bem como a sua orientação, utiliza-se o operador de *Canny* [Canny86].

O resultado da análise da textura para a imagem já usada na análise geométrica é ilustrado na Figura 2.27.



Figura 2.27 – Segmentação depois da análise de textura (242 regiões) [Lienhart00].

2.6.2.2 Seguimento de Texto

Nesta etapa, é explorada a redundância temporal existente no vídeo. Assume-se que o texto gráfico surge estático ou com movimentos lineares, uma vez que os movimentos não lineares são pouco prováveis e de difícil seguimento. O objectivo da análise do movimento é efectuar o seguimento ao longo de toda a sua permanência na sequência de vídeo das regiões que foram anteriormente classificadas como texto de modo a melhorar a sua classificação. Para tal, são identificadas as regiões que não podem ser seguidas ao longo das várias tramas ou que não têm um movimento linear para as reclassificar como regiões que não correspondem a caracteres e as descartar. Esta análise é feita em duas etapas:

1ª Formação das cadeias de caracteres

Na primeira etapa são formadas cadeias de caracteres definidas como colecções de regiões que representam cada caracter ao longo do tempo, i.e. ao longo das várias tramas onde ele existe. Note-se que cada trama só pode contribuir com uma região para a formação de uma dada cadeia de caracteres. Uma cadeia de caracteres C é descrita como uma estrutura tripla $(A, [a, e], v)$: em A são colocados os valores das características das regiões que originaram a cadeia de caracteres e que são utilizadas para efectuar a comparação com novas regiões; $[a, e]$ armazena o intervalo temporal onde a cadeia existe no vídeo; em v é colocada a direcção do movimento da região e a velocidade estimada para esta entre duas tramas contíguas.

Assim, a região R_i na trama N é comparada com todas as cadeias de caracteres C_j , $j \in \{1, \dots, J\}$ formadas nas tramas 1 a $N-1$. Se a região R_i for suficientemente semelhante à região que representa a cadeia de caracteres comparada, ou seja a região que a originou, é-lhe adicionada a nova região R_i (ou seja prolongando a cadeia de caracteres no tempo). Para verificar se a região R_i é suficientemente semelhante à região que representa a cadeia de caracteres em teste, são comparadas a cor, o tamanho (número de *pixels*) e a posição da região R_i com as características correspondentes das cadeias de caracteres C_j , $j \in \{1, \dots, J\}$.

No final do processamento da trama N , são eliminados todas as cadeias de caracteres que não reúnam as seguintes características:

- Todas as cadeias de caracteres que foram criadas na trama $N-1$ e não continuaram na trama N ;
- Todas as cadeias de caracteres com duração inferior a 8 tramas e que não continuaram durante as últimas 6 tramas em relação à trama N ou as cadeias cuja predição de posição na trama $N+1$ seja fora da trama. Esta predição é feita com base na informação de movimento v para cada cadeia de caracteres;
- Todas as cadeias de caracteres que se movam com uma velocidade superior a 9 *pixels*/trama.

Depois de processadas todas as tramas da sequência de vídeo, as cadeias de caracteres que possuírem uma duração temporal inferior a 8 tramas são eliminadas. As cadeias de caracteres válidas formam o conjunto dos caracteres da sequência de vídeo.

2ª Formação de palavras

Na segunda etapa, as cadeias de caracteres são agrupados em palavras e linhas de texto. Uma palavra é formada por, pelo menos, três caracteres que se encontrem próximos entre si e que cumpram os seguintes requisitos:

- Ocorram na mesma trama;
- Apresentem o mesmo movimento linear;
- Tenham a mesma cor;
- Formem uma linha recta;
- Sejam vizinhos, i.e. não distem mais do que um valor pré-determinado na direcção da escrita.

No início do processo de agrupamento visando a formação das palavras ou linhas de texto, todas as cadeias de caracteres contidas no conjunto de cadeias de caracteres são consideradas. Posteriormente, são formadas combinações de três cadeias de caracteres que representem uma palavra válida. Essas cadeias de caracteres são então movidas do conjunto de cadeias de caracteres para dentro da nova palavra. Seguidamente, todas as cadeias de caracteres restantes são analisadas e movidas ou não para a nova palavra em função do seu ajustamento à mesma. Este processo de procura da próxima palavra válida e adição das cadeias de caracteres continua até que não seja possível formar mais palavras válidas ou não existam mais cadeias de caracteres para adicionar às palavras.

Uma vez concluída a formação das palavras, poderá ocorrer a falta de alguns caracteres nalgumas tramas pertencentes ao intervalo de duração da palavra. A recuperação desses caracteres torna-se importante e é feita através da sua interpolação. Na Figura 2.28 ilustra-se o resultado depois da fase de formação de palavras para uma trama individual.



(a)

PETRA KLEINERT
 CLAUDIA WENZEL
 GISELA TROWE
 WOLFGANG WINKLER
 GERD BALTUS
 BILLIE ZOCKLER
 MICHAEL WALKE

(b)

Figura 2.28 – Exemplo da formação das palavras: (a) imagem original e (b) resultado da formação de palavras para a imagem em (a) [Lienhart00].

2.6.2.3 Reconhecimento do Texto

Depois de terminada a formação das cadeias de texto e das regiões classificadas como não texto terem sido descartadas, é aplicado a cada trama um sistema OCR comercial ou um sistema de reconhecimento especificamente desenvolvido que reconhece o texto existente em cada trama e o converte para código ASCII.

No sistema proposto em [Lienhart00] foi utilizado o *software* OCR *development kit* Recognita V3.0 para Windows 95, o qual foi incorporado no sistema de extracção de texto desenvolvido. Na Figura 2.29 é ilustrado o resultado do reconhecimento utilizando o OCR Recognita V3.0, aplicado a uma imagem depois de efectuada a formação de palavras.

WOLFGANG WINKLER
 WALKE
 PETRA KLEINERT
 BILLIE ZOCKLER
 MICHAEL
 GERD BALTUS
 CLAUDIA WENZEL
 GISELA TROWE

(a)

WOLFGANG WINKLER
 WALKE
 PETRA KLEINERT
 BILLIE ZOCKLER
 MICHAEL
 GERD BALTUS
 CLAUDIA WENZEL
 GISELA TROWE

(b)

Figura 2.29 – Exemplo do reconhecimento de texto utilizando o OCR Recognita V3.0: (a) imagem depois da formação de palavras; (b) resultado do reconhecimento do texto existente na imagem em (a) [Lienhart00].

2.6.2.4 Indexação e Pesquisa

Para efectuar a indexação e a pesquisa no vídeo é utilizado o resultado do reconhecimento do texto efectuado na fase anterior. A questão que se coloca nesta fase é qual a qualidade necessária para o reconhecimento do texto para que seja possível efectuar a sua indexação e pesquisa? Numerosos tipos de fontes, tamanhos e estilos são utilizados no texto artificial utilizado nos vídeos digitais. Assim, os erros de reconhecimento por parte dos sistemas OCR são comuns, o que dificulta a tarefa de indexação e pesquisa. De forma a tentar colmatar esta dificuldade, o sistema proposto em [Lienhart00] foi concebido para funcionar ainda que o reconhecimento do texto possa possuir baixa qualidade, i.e. possam existir muitos erros de reconhecimento.

Indexação

Na fase de indexação, o texto reconhecido em cada trama é armazenado depois de eliminadas todas as linhas de texto com menos de três caracteres. A necessidade desta eliminação deve-se ao facto de as linhas com menos de três caracteres serem produzidas essencialmente por ruído que foi classificado como texto, para além de possuírem pouco valor semântico.

Pesquisa Textual

A pesquisa textual em sequências de vídeo é efectuada com recurso à procura de determinada cadeia de caracteres (*string*). Neste sistema são utilizados dois modos de procura:

- 1º **Emparelhamento exacto** – Este modo de procura retorna todas as tramas onde a cadeia de caracteres reconhecida for rigorosamente idêntica à cadeia de caracteres chave utilizada para a procura;
- 2º **Emparelhamento aproximado** – Este modo de procura tolera um certo número de caracteres diferentes entre a cadeia de caracteres chave utilizada para a procura e o texto reconhecido. Neste modo de procura é utilizada a distância de Levenshtein $L(A,B)$ entre uma cadeia de caracteres A e uma cadeia de caracteres B . A distância de Levenshtein é definida como sendo o menor número de subtracções, eliminações e inserções de caracteres que é necessário efectuar na cadeia de caracteres A para a transformar na cadeia de caracteres B [Stephen94]. Assim, para cada trama é calculada a distância de Levenshtein mínima entre a cadeia de caracteres chave e as linhas de texto existentes na trama. Se esta distância mínima for menor do que um determinado limiar pré-definido, a existência da palavra chave na trama é assumida como verdadeira.

Deste modo, os algoritmos de detecção e reconhecimento propostos em [Lienhart00] podem ser utilizados para fazer pesquisa textual em vídeo.

2.6.2.5 Avaliação do Desempenho

Para efectuar a avaliação do desempenho do sistema proposto em [Lienhart00], foram utilizadas dez sequências de vídeo, num total de cerca de 22 minutos, digitalizadas a partir de emissões de TV, com uma resolução espacial de 384×288 *pixels* e uma resolução temporal de 25 tps. Os conteúdos incluem inícios e finais de filmes, comerciais e programas informativos. Três tipos de avaliação foram efectuados:

1º Avaliação da detecção – Na avaliação da detecção e antes de processar cada sequência de vídeo, determina-se manualmente através da observação do vídeo a trama de início e de fim de cada ocorrência de texto, bem como o texto nelas contido. Os resultados obtidos para a avaliação da detecção são apresentados na Tabela 2.3.

Tabela 2.3 – Desempenho em termos de detecção de texto.

	Início e finais de filmes	Comerciais	Programas informativos
Nº de tramas	2874	579	3147
Nº de caracteres	2715	264	80
Nº de caracteres correctamente detectados	2596 (96%)	173 (66%)	79 (99%)

O desempenho da detecção é elevado para os inícios e finais de filmes e programas de informação: 96% e 99%, respectivamente. O desempenho é mais elevado para as sequências de vídeo onde existe movimento do texto e/ou do fundo, uma vez que neste tipo de sequências a análise de movimento torna possível a eliminação de falsas detecções. Este desempenho é menor para texto e fundo estacionários, o que ocorre com mais frequência nos comerciais. Assim, o desempenho da detecção é menor para as sequências onde predominam os comerciais, ou seja da ordem dos 66%;

2º Avaliação do reconhecimento – Na avaliação do desempenho do algoritmo proposto em termos de reconhecimento do texto foi utilizado o sistema OCR Recognita V3.0. Para quantificar os valores do desempenho foram utilizadas duas métricas:

- ♦ **Rácio de Caracteres Reconhecidos (RCR)** – Relação entre o número de caracteres reconhecidos correctamente e o número total de caracteres a detectar;
- ♦ **Rácio de Caracteres Errados (RCE)** – Relação entre o número de caracteres reconhecidos erradamente e o número total de caracteres a detectar;
- ♦ **Rácio de Caracteres Errados por Trama (RCET)** – Relação entre o número de caracteres reconhecidos erradamente em tramas sem texto e o número de tramas sem texto.

Nas tramas onde não existe texto não é calculado o RCE. Nessas tramas são contados os caracteres falsamente reconhecidos, contribuindo assim para a métrica (RCET). Os resultados obtidos para a avaliação do reconhecimento são apresentados na Tabela 2.4.

Tabela 2.4 – Desempenho em termos de reconhecimento do texto.

	Início e finais de filmes	Comerciais	Programas informativos
RCR	0.76	0.65	0.41
RCE	0.09	0.14	0.46
RCET	0	25.44	16

Os valores obtidos para o reconhecimento variam entre 41% e 76% dependendo do conteúdo (ver Tabela 2.4). O valor de RCET é muitas vezes elevado para as tramas sem texto o que indica que ocorrem muitas detecções falsas, especialmente para comerciais e programas informativos, onde o texto e as cenas estacionárias são mais comuns;

3º Avaliação da pesquisa – A avaliação da pesquisa é feita através da capacidade do sistema detectar somente texto relevante, enquanto ignora o texto não relevante. Para tal, foram utilizadas duas métricas:

- ♦ **Recall** – Relação entre o número de ocorrências de texto relevante retornadas pelo sistema e o número total de ocorrências de texto relevante existentes no vídeo;
- ♦ **Precisão** – Relação entre o número de ocorrências de texto classificadas como relevantes pelo sistema e o número de ocorrências de texto retornadas pelo sistema.

Assume-se que texto relevante foi correctamente identificado quando é localizada pelo menos uma trama pertencente ao intervalo de tramas onde a cadeia de caracteres chave de procura existe. Para avaliar o desempenho do sistema foram utilizadas como cadeias de caracteres chave, palavras existentes nas sequências de vídeo de teste. A pesquisa foi efectuada utilizando quer o modo de emparelhamento exacto, quer o modo de emparelhamento aproximado. Na

Tabela 2.5 são apresentados os resultados obtidos para a avaliação da pesquisa textual.

Tabela 2.5 – Desempenho em termos de pesquisa textual

	Emparelhamento exacto		Emparelhamento aproximado	
	<i>Recall</i>	<i>Precisão</i>	<i>Recall</i>	<i>Precisão</i>
Início e finais de filmes	0.60	0.71	0.78	0.54
Comerciais	0.74	0.73	0.54	0.65
Programas informativos	0.64	0.95	0.82	0.60

O valor de *recall* é superior para a pesquisa por aproximação. Tal, deve-se ao facto de no emparelhamento aproximado necessita-se apenas que parte da cadeia de caracteres chave, seja coincidente com o texto existente no vídeo. Esta maior facilidade na pesquisa do texto origina o aumento de falsas detecções, levando a que o valor da precisão seja inferior para o emparelhamento aproximado quando comparada com o emparelhamento exacto.

Assim, os valores de *recall* para o emparelhamento aproximado são ordem dos 0.54 a 0.82, i.e. 54 a 82% do material relevante existente no vídeo é localizado. A precisão varia

entre 0.54 e 0.60, valores indicativos de que 54 a 60% do texto relevante existente no vídeo é correctamente localizado.

Para o emparelhamento exacto a *recall* tem valores que variam entre 0.60 e 0.64, o que indica que 60 a 64% do texto relevante é localizado. A sua precisão que varia entre 0.71 e 0.95, valores indicativos de que 71 a 95% do texto relevante é localizado correctamente.

O sistema proposto em [Lienhart00] e aqui descrito apresenta um sistema completo para a extracção de texto em sequências de vídeo. Este sistema propõe (apesar de não efectuar a sua descrição em detalhe) a combinação da detecção de fronteiras com a orientação local da variação do contraste para melhorar a segmentação das tramas em regiões conexas, bem como uma técnica de segmentação do contraste que explora o elevado contraste existente entre o fundo e o texto, nomeadamente em texto gráfico. Os princípios utilizados nestas técnicas foram explorados na presente Tese de modo a aplicá-los a conteúdos onde o texto é menos contrastado em relação ao fundo, característica típica do texto de cena. Para efectuar o seguimento do texto, é utilizada uma técnica que se baseia na comparação de tramas sucessivas. Cada vez que um caracter novo surge, é gerada uma assinatura para esse caracter. Esta assinatura é utilizada para efectuar a comparação ao longo das tramas. Todavia, este sistema foi concebido apenas para efectuar a extracção de texto gráfico (com um elevado contraste em relação ao fundo das tramas) e escrito na horizontal.

2.6.3 Extracção de Texto em Imagens e Vídeos

Em [Lienhart02] é apresentado um sistema para extracção de texto gráfico em imagens e vídeo que, para além da detecção e reconhecimento do texto, permite também efectuar o seguimento do texto no tempo (isto é claro para o caso do vídeo). A escolha deste sistema deveu-se ao facto de ser um sistema recente onde o autor obteve resultados muito bons tanto para a detecção de texto em imagens como em sequências de vídeo. Para além disso, ilustra a utilização de um classificador neuronal para classificar cada região como texto ou não texto, utilizando como dados de entrada as fronteiras existentes nas tramas de vídeo ou imagens.

Duas condições são impostas à caracterização do texto neste sistema:

- Somente texto horizontal é considerado, uma vez que este ocorre em mais de 99% dos casos de texto gráfico [Lienhart02];
- A altura do texto pode variar entre 8 *pixels* e metade da altura da imagem;
- As ocorrências de texto só são consideradas se existirem pelo menos dois caracteres seguidos;
- Só o texto cuja cor para uma mesma linha não se altera ao longo do tempo é considerado.

O sistema proposto em [Lienhart02] engloba três fases principais: detecção, segmentação e reconhecimento do texto, como ilustrado na Figura 2.30. Este sistema aborda tanto a extracção de texto em imagens, como a extracção de texto em sequências de vídeo. Estes dois tipos de abordagens diferem essencialmente ao nível da detecção de texto, uma vez que a detecção de texto no vídeo possui mais uma fase do que a detecção de texto em imagens, a

fase de seguimento do texto. Como tal, a detecção de texto será descrita de forma separada para as imagens e para o vídeo.

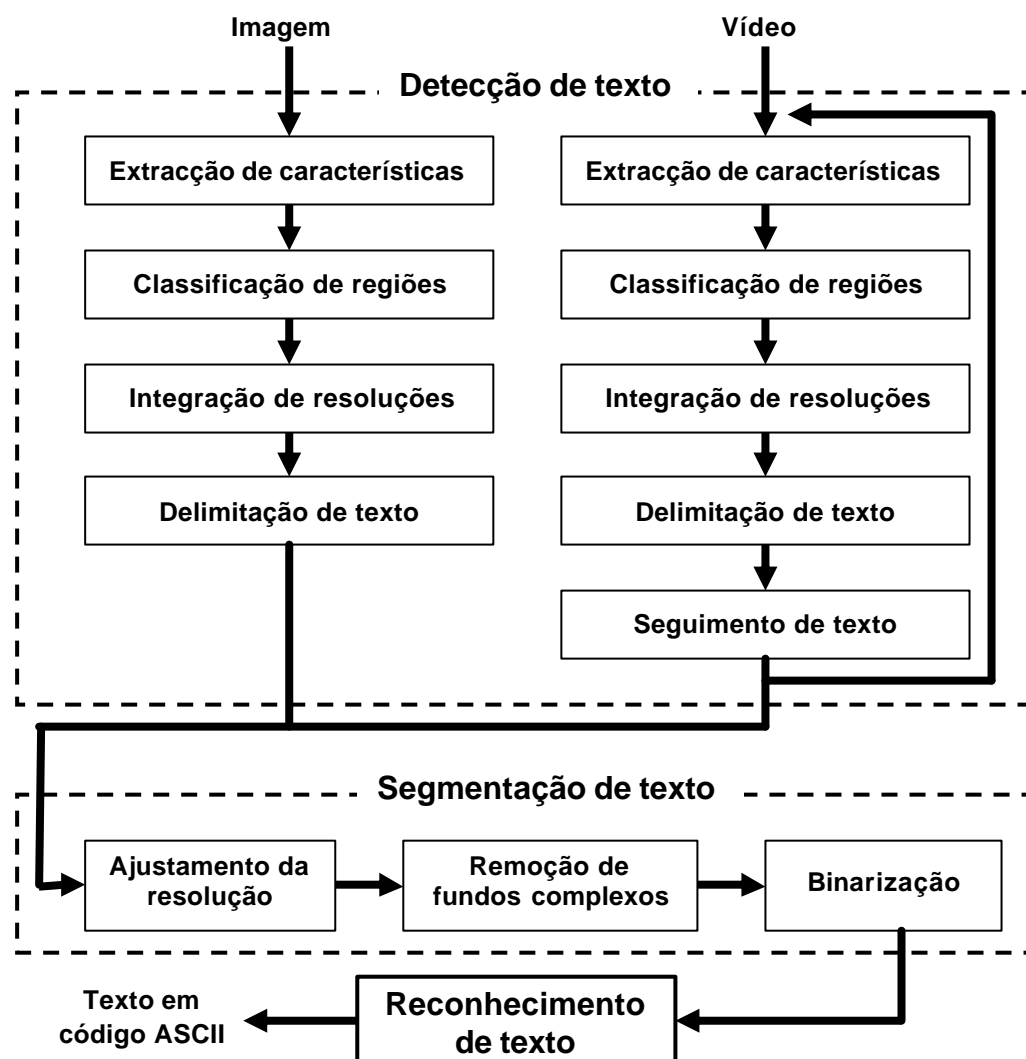


Figura 2.30 – Arquitectura do sistema de extracção de texto em imagens e vídeo proposto em [Lienhart02].

As três fases pelas quais passa a extracção de texto são descritas nas secções seguintes.

2.6.3.1 Detecção de Texto em Imagens

Nesta fase, o texto é detectado e circunscrito por *bounding boxes*. Cada uma delas circunscreve apenas uma palavra ou uma linha de texto. Como ilustrado na Figura 2.30, a detecção do texto decorre em quatro fases: extracção de características, classificação das regiões, integração das resoluções e delimitação do texto. A detecção de texto em imagens não possui a fase de seguimento. Estas fases são de seguida descritas de uma forma mais pormenorizada.

1ª Fase – Extracção de características

Para possibilitar a detecção de texto de vários tamanhos, é feita a decomposição da imagem original em diferentes resoluções, utilizando-se um factor de subamostragem para as resoluções em x e y de 3 e 2, respectivamente. Para cada uma das resoluções, é aproveitado o facto do texto gráfico se caracterizar, normalmente, por regiões de elevado contraste e componentes de alta frequência. Várias são as técnicas que podem ser utilizadas para amplificar estas características da imagem. No sistema aqui descrito, Lienhart e Wernicke [Lienhart02] utilizaram o gradiente RGB da imagem de entrada I , $I(x,y) = (I_r(x,y), I_g(x,y), I_b(x,y))$, para calcular a imagem de orientação das fronteiras, $E(x,y)$, definida como:

$$E(x, y) = \left(\sum_{c \in \{r, g, b\}} \left| \frac{I_c(x, y)}{dx} \right|, \sum_{c \in \{r, g, b\}} \left| \frac{I_c(x, y)}{dy} \right| \right) \quad (2.4)$$

Onde, dada a matriz $I(x,y)$ da imagem I , as matrizes $I_r(x,y)$, $I_g(x,y)$, $I_b(x,y)$ representam as componentes R,G e B do *pixel* (x,y) na imagem I , respectivamente.

Para cada resolução da imagem original, é calculada uma imagem de orientação de fronteiras $E(x,y)$ que representa o mapa de todas as fronteiras com orientação compreendida entre 0° e 90° permitindo assim efectuar a distinção entre fronteiras com orientação horizontal, diagonal e vertical. Como características para efectuar a localização do texto em cada resolução da imagem original são utilizadas as fronteiras detectadas.

2ª Fase – Classificação de regiões

A classificação das regiões propriamente dita é conseguida através da classificação de regiões de 20×10 *pixels* como sendo texto ou não, sendo que o número de linhas da janela determina o tamanho dos caracteres que podem ser detectados. Para tal, é necessária a utilização de um detector de texto que usa uma janela de 20×10 *pixels* para efectuar o varrimento das várias imagens de fronteiras E (imagem que contém as orientações das fronteiras para cada resolução da imagem original) e classificar cada janela como contendo texto ou não. Esta dimensão foi seleccionada por apresentar uma boa relação entre o desempenho do sistema e a sua complexidade computacional. Foi testada, também, a dimensão de 30×15 *pixels*; todavia não aumentou o desempenho do sistema mas apenas a sua complexidade computacional. Por outro lado, uma dimensão menor diminuiu o desempenho do sistema.

Como detector de texto foi utilizada uma rede neuronal, previamente treinada, constituída por 200 neurónios que são alimentados por uma janela de 20×10 da imagem de fronteiras E . A resposta da rede neuronal para cada imagem de fronteiras E é guardada numa imagem, denominada imagem de resposta. O enchimento da imagem de resposta com o valor de saída da rede neuronal efectua-se, se e só se, o referido valor for superior a um limiar de decisão Th_{rede} previamente determinado (foi utilizado $Th_{rede}=0$). Assim, valores positivos na saída da rede correspondem a regiões de texto. O autor não refere o intervalo de valores de resposta da rede; todavia estes valores estão sempre associados ao treino dado à rede.

3ª Fase – Integração de resoluções

Como o detector de texto é aplicado a todas as resoluções derivadas da imagem original, é necessário fazer a integração dos resultados da classificação para a resolução original com vista a identificar numa mesma resolução as regiões classificadas como texto. Na Figura 2.31 é ilustrado um exemplo da integração numa única imagem, da saída da rede neuronal para várias resoluções. As regiões que a rede neuronal classificou como texto são apresentadas a

branco e as classificadas como não texto a preto. Ao mapa resultante da integração das várias resoluções, chama-se mapa saliente S .

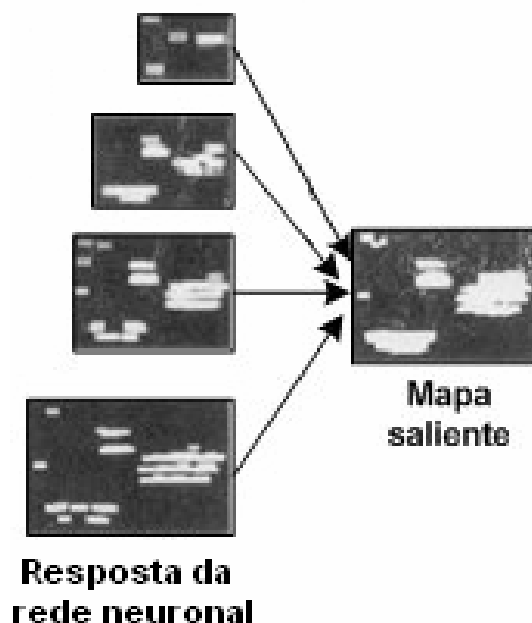


Figura 2.31 – Exemplo da integração dos resultados da classificação efectuada pela rede neuronal para as várias resoluções de modo a formar o mapa saliente [Lienhart02].

4ª Fase – Delimitação de texto

As palavras ou linhas de texto existentes na imagem são delimitadas por *bounding boxes*. Para determinar as *bounding boxes* delimitadoras do texto é utilizado o mapa saliente, i.e. a imagem com o resultado da integração das classificações efectuadas pela rede neuronal para as várias resoluções da imagem. Esta delimitação é conseguida em três passos:

- 1º **Criação de *bounding boxes*** – Neste passo são criadas *bounding boxes* que circunscrevem as palavras ou linhas de texto. Para tal, foi utilizado um algoritmo baseado na técnica *region-growing* para segmentar o mapa saliente. Uma posição de início é necessária para formar cada região de texto. Cada região cresce através da adição de linhas e colunas completas do rectângulo ao qual pertence o *pixel* que serviu de semente para a criação da região. A decisão de agrupar ou não novas linhas ou colunas é baseada no valor médio das linhas ou colunas candidatas do mapa saliente.

O algoritmo inicia-se com a procura do próximo *pixel* que ainda não tenha sido processado no mapa saliente com um valor superior a um determinado limiar Th_{core} previamente definido. Na escolha do limiar de decisão tem-se em conta que a rede neuronal classifica as regiões de não texto com valores mais baixos do que as regiões de texto. Neste sistema $Th_{core}=5$ funciona bem; todavia este valor pode ter de ser ajustado consoante o treino dado à rede. Sempre que é encontrado um *pixel* no mapa saliente onde $S(x,y) > Th_{core}$, este é considerado como a posição de início de uma nova região de texto

com uma *bounding box* de largura e altura igual a 1. Esta nova região de texto é expandida para cima de forma iterativa com base na média do valor dos *pixels* da linha (do mapa S) por cima da *bounding box*. Se o valor da média for superior a $Th_{região}=4.5$, a linha é adicionada à região. Seleccionou-se $Th_{região} < Th_{core}$ para que as regiões contenham todas as partes dos caracteres e não apenas as partes centrais. Este processo iterativo continua até que a *bounding box* pare de crescer. O mesmo critério é utilizado para expandir as *bounding boxes* para baixo, esquerda e direita.

As coordenadas das *bounding boxes* na imagem original correspondem às coordenadas destas no mapa saliente;

- 2º **Refinação de *bounding boxes*** – Neste passo é refinada a posição e as dimensões das *bounding boxes* criadas no passo anterior. Esta necessidade surge uma vez que enquanto a algumas *bounding boxes* podem faltar-lhe linhas ou colunas, outras delimitam mais do que uma linha de texto e a maioria delas possuem grandes quantidades de *pixels* de fundo. De modo a melhorar as *bounding boxes* utiliza-se a informação contida no perfil das projecções horizontal e vertical. A projecção horizontal e vertical são descritas por valores ao longo dos eixos dos xx e yy e, são definidas como o vector da soma dos valores da intensidade de cada *pixel* ao longo de cada coluna e linha, respectivamente. São ilustrados na Figura 2.32 (a) um exemplo da projecção horizontal e na Figura 2.32 (b) um exemplo da projecção vertical.

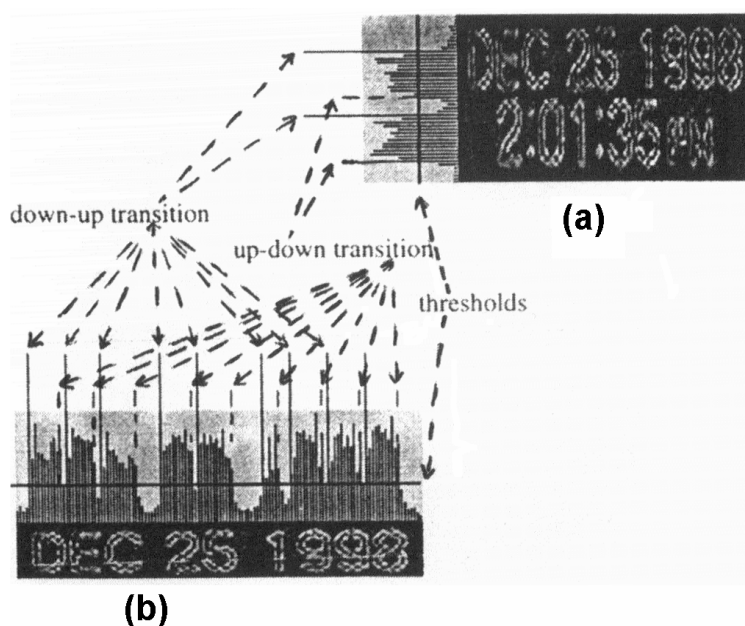


Figura 2.32 – Exemplo com o perfil das projecções utilizadas na formação de palavras ou linhas de texto: (a) projecção horizontal e (b) projecção vertical [Lienhart02].

Os limites superiores e inferiores das linhas de texto são assinaladas por valores elevados na projecção horizontal. De forma semelhante, os limites esquerdo e direito do texto são assinalados na projecção vertical por valores elevados. Deste modo, a projecção horizontal é utilizada para criar *bounding boxes* que contenham uma única linha de texto, enquanto que a projecção vertical é utilizada para dividir as linhas de texto em palavras. As variações nos valores das projecções podem ser utilizadas para localizar o texto desde

que seja definido um valor de limiar adaptativo Th_{texto} que permita separar as regiões de texto das de não texto.

Assim, a refinação das *bounding boxes* inicia-se com a aplicação da projecção horizontal para isolar as linhas de texto, seguida da projecção vertical para dividir as linhas em palavras. Para aplicar a projecção horizontal a cada *bounding box*, começa-se por expandir estas para cima e para baixo em 25% da sua altura. Esta expansão é necessária para assegurar que a totalidade dos caracteres é delimitada pela *bounding box*. De seguida é calculado um valor de limiar, Th_{texto} , que permite verificar se as linhas formadas pela projecção horizontal correspondem a linhas de texto ou não. Th_{texto} é dado pela expressão:

$$Th_{texto} = \min_{perfil} + (\max_{perfil} - \min_{perfil}) \times 0.175 \quad (2.5)$$

onde os valores \max_{perfil} e \min_{perfil} correspondem ao valor máximo e mínimo do perfil da projecção horizontal efectuada sobre a *bounding box* expandida. O valor 0.175 foi obtido empiricamente. Deste modo, as linhas cujo perfil horizontal exceda Th_{texto} são classificadas como contendo texto.

A aplicação da projecção vertical é feita de forma semelhante, mas unicamente às *bounding boxes* que contenham uma única linha de texto. Todavia, existem duas diferenças:

- 1ª O factor 0.175 da projecção horizontal é substituído pelo factor 0.25 na projecção vertical por se obterem melhores resultados com este valor, também, obtido empiricamente;
- 2ª Um parâmetro de abertura é definido para possibilitar a divisão da linha em palavras. Na definição do valor de abertura tem-se em conta que o espaçamento entre palavras é maior do que o espaçamento entre caracteres; deste modo, definiu-se um parâmetro de abertura adaptativo e que é igual à altura da *bounding box* em análise.

A aplicação da projecção horizontal seguida da projecção vertical provoca a divisão das *bounding boxes* em várias regiões, algumas das quais pequenas. Uma vez concluída a refinação das *bounding boxes* de texto, estas são filtradas em termos da sua altura, ou seja, *bounding boxes* com

$$altura_{caixa} < \min_{altura_texto} = 8pt \quad (2.6)$$

ou

$$altura_{caixa} > \max_{altura_texto} = \frac{altura_{imagem}}{2} \quad (2.7)$$

são classificadas como sendo regiões sem texto e, como tal, eliminadas. Como resultado, tem-se a imagem original com as palavras ou linhas de texto circunscritas por *bounding boxes*.

- 3º **Determinação da cor do texto e do fundo** – Neste passo são estimadas para cada *bounding box*, tanto a cor do texto como a cor do fundo de modo a verificar se o texto é normal ou inverso. Se o valor da luminância do texto é inferior ao do fundo, assume-se que o texto é normal; caso contrário, o texto é inverso. Todavia, quer nas imagens, quer

nos vídeos, o número de cores é normalmente elevado. Assim, para simplificar o processo de determinação da cor do texto e do fundo, o número de cores existente em cada *bounding box* é reduzido através da quantificação das quatro cores dominantes, utilizando para tal o algoritmo de quantificação de cor proposto por Wu [Wu96]. Dois histogramas são formados a partir da *bounding box* quantificada com as quatro cores:

- ◆ Um histograma de cor que cobre as quatro linhas do centro da *bounding box*;
- ◆ Um histograma de cor que cobre as duas linhas de cima e de baixo da *bounding box*.

No primeiro histograma, a cor predominante é a cor do texto, enquanto que no segundo histograma a cor dominante é a cor do fundo. Deste modo, consegue-se estimar a cor do texto e a cor do fundo de cada *bounding box*. Esta técnica falha quando o texto dentro da mesma *bounding box* possui mais do que uma cor.

2.6.3.2 Detecção de Texto em Vídeo

Na detecção de texto em vídeo é explorada a redundância temporal inerente ao vídeo. Para isso, as *bounding boxes* com o mesmo conteúdo, em tramas sucessivas, são definidas como um objecto de texto. O objecto de texto representa uma palavra ou uma linha de texto ao longo do tempo, através dos seus mapas de bits, tamanho e posição nas várias tramas em que estas ocorrem. Os objectos de texto são extraídos em duas fases: análise do vídeo e seguimento do texto, como é ilustrado no esquema apresentado pela Figura 2.33.

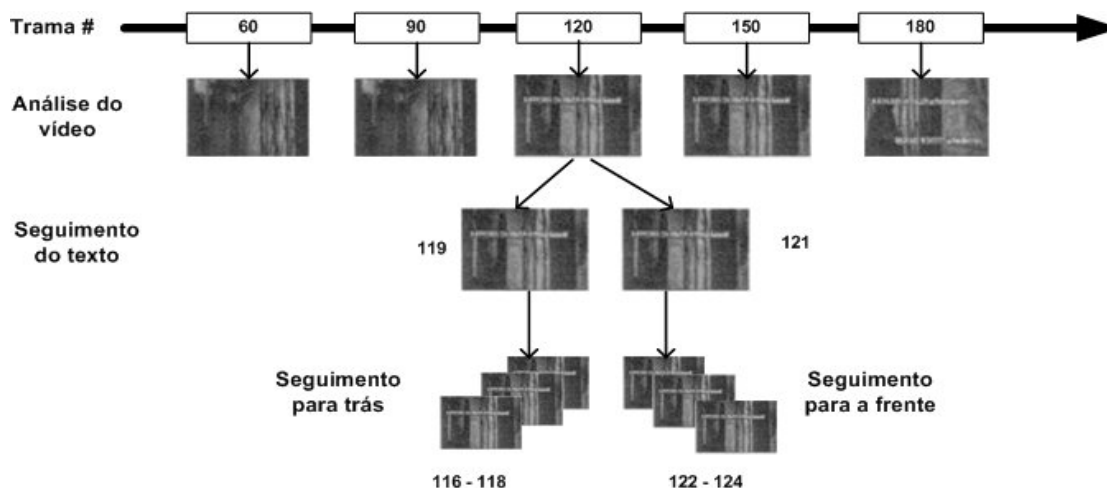


Figura 2.33 – Relação entre a fase de análise do vídeo e a fase de seguimento do texto [Lienhart02].

As duas fases pelas quais passa a extracção de texto em sequências de vídeo no sistema aqui descrito serão apresentadas de seguida.

1ª Fase – Análise periódica do vídeo

Na primeira fase, o vídeo é analisado ao longo do tempo de uma forma grosseira ou seja apenas algumas tramas são analisadas. Para tal, o método descrito anteriormente para detectar texto em imagens é aplicado, periodicamente, a uma em cada 30 tramas, como ilustrado na Figura 2.33. Se for detectado texto, a segunda etapa é iniciada.

Assim, a análise periódica do vídeo é efectuada, com baixa resolução temporal, de forma a detectar ocorrências de texto ao longo do tempo. Para alcançar esse objectivo, o detector de texto é aplicado apenas a um subconjunto das tramas, convenientemente espaçadas no tempo. Se o detector de texto não detectar qualquer linha de texto na trama t , o processo de análise continua na trama $t+30$. Se pelo menos uma linha de texto for detectada, o detector de texto é aplicado à trama $t-1$ e $t+1$. Assim, para cada linha de texto na trama t , o algoritmo procura a linha de texto correspondente nas tramas $t-1$ e $t+1$. A correspondência entre duas linhas de texto é considerada válida se as suas respectivas *bounding boxes* se sobrepuserem uma à outra em mais de 80% dos seus *pixels*. A percentagem de sobreposição é definida por:

$$\text{sobreposição} = \frac{A \cap B}{A} \quad (2.8)$$

onde A representa o conjunto de *pixels* que correspondem à *bounding box* detectada na trama n e B o conjunto dos *pixels* que correspondem à segunda *bounding box*, detectada na trama $n+1$ ou $n-1$. Se a correspondência entre as duas linhas de texto for válida, passa-se à fase de seguimento do texto.

2ª Fase – Seguimento de texto

Uma vez detectado texto numa trama de vídeo, este necessita de ser expandido no tempo para todas as tramas que contêm a linha de texto que o originou, quer para a frente, quer para trás, para formar um objecto de texto. Será descrito o seguimento para a frente, uma vez que o seguimento para trás é idêntico, excepto na direcção.

Para efectuar o seguimento na trama $n+1$ de uma palavra ou linha de texto detectada na trama n , é necessário calcular a sua assinatura ou seja o perfil resultante das suas projecções vertical e horizontal. Tal procedimento possibilita fazer a sua distinção em relação às outras palavras ou linhas de texto. Assim, procura-se na trama seguinte e na trama anterior, uma região da imagem com a mesma dimensão e com a melhor comparação em termos de assinatura. Esta procura exaustiva da assinatura é semelhante ao processo de *block-matching* utilizado na predição de movimento em codificação de vídeo, com a excepção de que aqui a medida de semelhança é baseada na assinatura. A assinatura deriva das características da *bounding box* actual e é actualizada de cinco em cinco tramas, i.e. de cinco em cinco tramas é feita uma detecção de texto para recalcular a assinatura de cada *bounding box*. O processo de seguimento termina quando num determinado número pré-definido de tramas contíguas não existir correspondência de linhas de texto.

Antes de serem reconhecidos, os objectos de texto são sujeitos a processamento com a finalidade de os validar. Assim, algumas verificações são feitas com o objectivo de descartar falsas detecções de texto tais como:

- Os objectos que ocorram por períodos com uma duração inferior a 1 segundo são considerados como falsos alarmes;
- Todos os objectos cujo seguimento falhou em mais de 25% das tramas são descartados.

Os objectos de texto que cumpriram as condições anteriormente apresentadas constituem o conjunto de objectos de texto que serão segmentados.

2.6.3.3 Segmentação de Texto

Antes da aplicação do texto a um sistema OCR para efectuar o seu reconhecimento, as *bounding boxes* das imagens ou tramas de vídeo são sujeitas a uma segmentação de modo a transformá-las em mapas de bits binários onde o texto é representado a branco e o fundo a preto para o texto inverso. No caso do texto normal, o texto é representado a preto e o fundo a branco. Esta segmentação decorre em três etapas:

1ª Ajustamento da resolução

A segmentação do texto inicia-se com o ajustamento da resolução das *bounding boxes*. Estas são escaladas por interpolação cúbica para uma altura fixa de 100 *pixels* de forma a preservar a sua relação largura-altura por duas razões principais:

- Para aumentar a resolução dos caracteres mais pequenos de forma a obter melhores resultados no processo de segmentação uma vez que a baixa resolução muitas vezes existente no vídeo é a maior fonte de problemas na segmentação e reconhecimento de texto;
- O texto com uma altura superior a 100 *pixels* não melhora a segmentação nem o desempenho dos sistemas OCR. Reduzindo o seu tamanho, diminui-se significativamente a complexidade computacional.

2ª Remoção de fundos complexos

Depois de efectuados os ajustamentos na resolução das *bounding boxes*, procede-se à remoção dos fundos com texturas complexas. Esta remoção tem um tratamento diferente caso se trate de imagens ou de vídeo.

- **Imagens** – A técnica adoptada para a remoção do fundo em imagens parte do pressuposto que o texto tem um contraste suficientemente elevado em relação ao fundo que é suficiente para que este seja legível. A ideia básica consiste em aumentar as *bounding boxes* para garantir que os *pixels* correspondentes a regiões de texto não sejam tocados por estas. Uma vez efectuado este aumento, utilizam-se como sementes para agrupar todos os *pixels* do fundo da imagem os *pixels* situados sobre o limite das *bounding boxes* (que idealmente não pertencem ao texto). Os *pixels* da *bounding box* que não diferirem mais do que um valor pré-definido dos *pixels* que servem de sementes são agrupados como fundo. De seguida, todas as regiões que tenham uma altura e uma largura inferiores a determinado valor pré-definido ou uma largura superior a um valor também pré-definido são descartadas, i.e. são convertidas em fundo;
- **Vídeo** – Atendendo a que, no caso do vídeo, os objectos de texto consistem em vários mapas de bits da mesma linha de texto obtidos em tramas sucessivas, mais uma vez se explora a redundância temporal de forma a remover o fundo texturado existente em torno dos caracteres. A técnica utilizada parte do pressuposto que os vários mapas de bits podem ser empilhados para que os caracteres fiquem perfeitamente alinhados, uns em relação aos outros. Se se olhar para um *pixel* específico ao longo do tempo, verifica-se

que se ele pertencer a uma zona de texto, o seu valor varia muito lentamente, enquanto que os *pixels* pertencentes a uma zona do fundo variam mais rapidamente. A probabilidade dos *pixels* do fundo mudarem é elevada, devido às alterações do fundo ou do movimento da linha de texto. Para remover o fundo são utilizados cerca de 40 mapas de bits, igualmente espaçados no tempo, para formar a pilha de mapas de bits perfeitamente alinhados. Deste modo, os *pixels* cujo valor variar pouco ao longo do tempo são classificados como texto e dão origem a uma nova *bounding box*. A esta *bounding box* é aplicada a técnica adoptada para a remoção do fundo em imagens, descrita anteriormente.

3ª Binarização

Neste passo, os mapas de bits resultantes da fase anterior são preparados de forma a poderem ser reconhecidos por um sistema OCR. Estes mapas são convertidos num mapa de bits binário definindo um valor de limiar global para a sua intensidade, localizado entre a intensidade do texto e do fundo. Foi adoptada a cor branca para o fundo no caso de texto normal e preta para o caso do texto inverso. Assim, cada *pixel* do mapa de bits que exceder o valor de limiar verá a sua cor convertida para preto ou seja tornar-se-á texto no caso de texto normal. Pelo contrário, verá a sua cor convertida para preto no caso de texto inverso.

2.6.3.4 Reconhecimento de Texto

No sistema aqui descrito e proposto em [Lienhart02] o reconhecimento do texto é efectuado através de um sistema OCR comercial que reconhece o texto existente nas imagens binárias e o converte para código ASCII. O sistema OCR utilizado foi o OmniPagePro 10.

2.6.3.5 Avaliação do Desempenho

Para efectuar a avaliação do desempenho do sistema proposto em [Lienhart02] foram utilizadas 23 pequenas sequências de vídeo num total de cerca de 10 minutos, com uma resolução espacial que varia entre 352×240 e 1920×1280 *pixels* e uma resolução temporal de 25 tps. As sequências de vídeo contém 2187 caracteres e foram seleccionadas a partir de programas informativos, filmes e comerciais. Para além das 10 sequências de vídeo foram, também, utilizadas sete páginas web na avaliação do desempenho.

O sistema proposto foi avaliado em termos da sua capacidade de detecção do texto existente nos vídeos e nas páginas web, segmentação do texto localizado e reconhecimento do texto segmentado. Deste modo, a avaliação decorre em três fases distintas:

- 1ª **Avaliação da detecção** – Nesta fase efectua-se a avaliação da detecção para cada trama e página web. Antes de avaliar a detecção, foi criada manualmente a *ground truth* para cada trama ou página web, i.e. o número de *bounding boxes* (que delimitam o texto) existente em cada trama ou página web. A avaliação do desempenho em termos de detecção é efectuada no final de três das cinco fases do processo de detecção de texto: criação das *bounding boxes* iniciais, refinamento das *bounding boxes* e análise de movimento do texto.

Para cada fase da detecção é avaliada a relação existente entre as *bounding boxes* Correctamente Criadas (CC), Falsamente Criadas (FC) e Não Criadas (NC) e o número total de *bounding boxes* existentes na *ground truth*. Uma *bounding box*, A , criada automaticamente pelo sistema é considerada correctamente criada, se e só se o seu emparelhamento (*matching*) em relação a uma *bounding box*, B , da *ground truth* coincidir em mais de 80%.

$$CC = \frac{100}{M} \times \sum \max\{d(a, g)\}, \quad g \in G \text{ e } a \in A \quad (2.9)$$

$$FC = \frac{100}{M} \times \left(N - \sum \max\{d(a, g)\} \right), \quad g \in G \text{ e } a \in A \quad (2.10)$$

$$NC = 100 - CC \quad (2.11)$$

Onde $d(a, g) = \begin{cases} 1, & \text{if } \min(|a \cap g|/|a|, |a \cap g|/|g|) \geq 0.8, \\ 0, & \text{else} \end{cases}$, e $A = \{a_1, \dots, a_n\}$ e

$G = (g_1, \dots, g_M)$ são os conjuntos de *pixels* que representam as *bounding boxes* criadas automaticamente pelo sistema e as *bounding boxes* da *ground truth* com tamanhos $N = |A|$ e $M = |G|$, respectivamente. $|a|$ e $|g|$ são o número de *pixels* em cada *bounding box* e $a \cap g$ representa o conjunto de *pixels* comuns a a e g .

Os resultados obtidos para cada uma das fases anteriormente identificadas são apresentados na Tabela 2.6.

Tabela 2.6 – Desempenho em termos de detecção do texto.

	CC	FC	NC
Criação das <i>bounding boxes</i> iniciais	48.8%	74.0%	51.2%
Refinamento das <i>bounding boxes</i>	69.5%	76.3%	30.5%
Análise de movimento	94.7%	18.0%	5.3%

Para imagens individuais ou páginas web, o sistema detecta correctamente 69.5% de todas as *bounding boxes*. Os 30.5% de *bounding boxes* não detectadas são provenientes de texto de pequenas dimensões, nomeadamente texto inferior a 10 *pixels*. O desempenho da detecção aumenta para 94.7% quando é explorada a redundância temporal existente nas sequências de vídeo e nesse caso unicamente 5.3% do texto não foi detectado.

- 2^a **Avaliação da segmentação** – Nesta fase efectua-se a avaliação do desempenho do sistema em segmentar correctamente o texto detectado. Para verificar se o texto é correctamente segmentado, é efectuada uma inspecção visual dos mapas de bits binários criados pela fase de segmentação. Os resultados obtidos para a fase de segmentação do texto são apresentados na Tabela 2.7.

Tabela 2.7 – Desempenho em termos de segmentação do texto.

CC	Caracteres danificados	NC
79.6%	18.0%	13.5% (inclui os 5.3% da fase de detecção)

De todos os caracteres existentes no conjunto de vídeos, 79.6% foram segmentados correctamente e 7.6% foram danificados (por exemplo, algumas das suas partes foram perdidas durante o seu processamento). Os caracteres danificados podem ser facilmente reconhecidos por humanos mas tipicamente não por um sistema automático. Se se considerar que 5.3% do texto não foi efectivamente detectado, somente 7.2% dos caracteres foram perdidos nesta etapa.

- 3^a **Avaliação do reconhecimento** – Nesta fase efectua-se a avaliação do desempenho do sistema em reconhecer correctamente o texto segmentado. Para efectuar o reconhecimento do texto e de modo a possibilitar a avaliação do sistema em termos de reconhecimento de texto foi utilizado o sistema OCR OmniPage Pro 10. Os resultados obtidos para o reconhecimento do texto são apresentados na Tabela 2.8.

Tabela 2.8 – Desempenho em termos de reconhecimento do texto.

CC	NC
69.9%	30.1% (inclui os 13.5% da fase de segmentação)

Foram reconhecidos correctamente 87.8% $\left(= \frac{69.9}{79.6} \times 100 \right)$ dos caracteres correctamente segmentados. De todos os caracteres que fazem parte da *ground truth*, foram correctamente reconhecidos 69.9%.

O sistema proposto em [Lienhart02] e aqui descrito propõe a utilização de um classificador neuronal para efectuar a extracção de texto gráfico escrito na horizontal, tanto em imagens como em sequências de vídeo. Para efectuar o seguimento do texto é utilizada uma técnica que se baseia na comparação de tramas sucessivas. Sempre que uma palavra ou uma linha de texto nova surge é gerada uma assinatura. Esta assinatura é utilizada para efectuar a comparação ao longo de cinco tramas, i.e. a cada cinco tramas a assinatura é recalculada. Neste sistema, é definido um valor de limiar global na fase de binarização, quando poderiam ser definidos valores de limiar adaptativos para cada *bounding box*. O uso de limiares adaptativos locais permitiria otimizar a segmentação dos objectos de texto, uma vez que nem todos os objectos de texto têm o mesmo contraste em relação ao fundo. Para além disso, a técnica utilizada para determinar a cor do texto e do fundo das *bounding boxes* falha quando o texto dentro da mesma *bounding box* possui mais do que uma cor. Os resultados obtidos são

considerados pelo autor como muito bons, tanto para as imagens como para as sequências de vídeo.

2.6.4 Extracção de Texto em Sequências de Vídeo com Integração de Múltiplas Tramas

A escolha para apresentação detalhada do sistema de extracção de texto proposto em [Li02] deveu-se essencialmente ao facto de ser um sistema recente que faz a extracção tanto de texto gráfico como de texto de cena, sendo reportado desempenhos muito bons nomeadamente para texto gráfico. Tal como o sistema apresentado anteriormente, este sistema utiliza também um classificador neuronal para classificar cada região da imagem como texto ou não texto.

O sistema proposto em [Li02] para a extracção de texto em sequências de vídeo digital através da integração de múltiplas tramas explora o facto de o texto, numa sequência de vídeo, se estender ao longo de dezenas ou mesmo de centenas de tramas (vários segundos). Dois conceitos são importantes para uma melhor compreensão deste método:

- **Bloco de texto** – Um bloco de texto consiste num conjunto de palavras ou linhas de texto, delimitadas por *bounding boxes*, próximas umas das outras e existentes numa determinada trama. Cada bloco de texto é constituído por dois tipos de *pixels*: *pixels* de texto e *pixels* de fundo; o valor dos *pixels* do bloco de texto varia através do espaço (ou seja dentro da *bounding box*). Na Figura 2.34 são ilustrados exemplos de blocos de texto numa imagem;

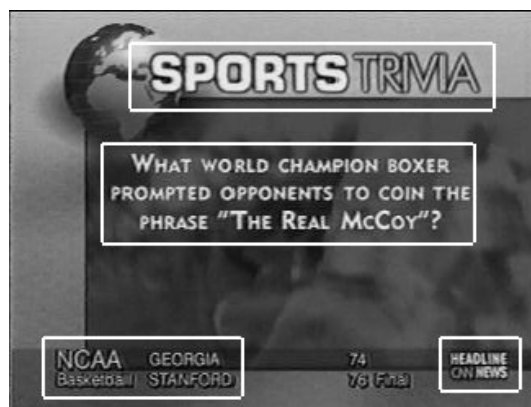


Figura 2.34 – Exemplos de blocos de texto segundo o método proposto em [Li02].

- **Objecto de texto** – Um objecto de texto é formado pelo conjunto de blocos de texto que contém as mesmas palavras ou linhas de texto ao longo das várias tramas sucessivas. O valor dos *pixels* do objecto de texto varia quer através do espaço (ou seja dentro da *bounding box*), quer através do tempo (ao longo das tramas);
- **Texto normal** – O texto normal é definido como sendo aquele cujo valor da intensidade é superior à intensidade do fundo;

- **Texto inverso** – O texto inverso é definido como sendo aquele cujo valor da intensidade é inferior à intensidade do fundo.

O sistema proposto é capaz de extrair texto com as seguintes características:

- Texto gráfico ou de cena;
- Texto com caracteres de diferentes tamanhos, fontes e estilos;
- Texto com caracteres orientados em qualquer direcção.

No sistema de extracção de texto em questão, o texto é extraído ao nível do bloco de texto e essa extracção decorre em quatro fases distintas: detecção e classificação do texto existente em uma das várias tramas, seguimento do texto extraído ao longo do tempo, segmentação dos blocos de texto e reconhecimento do texto, como ilustra a Figura 2.35.

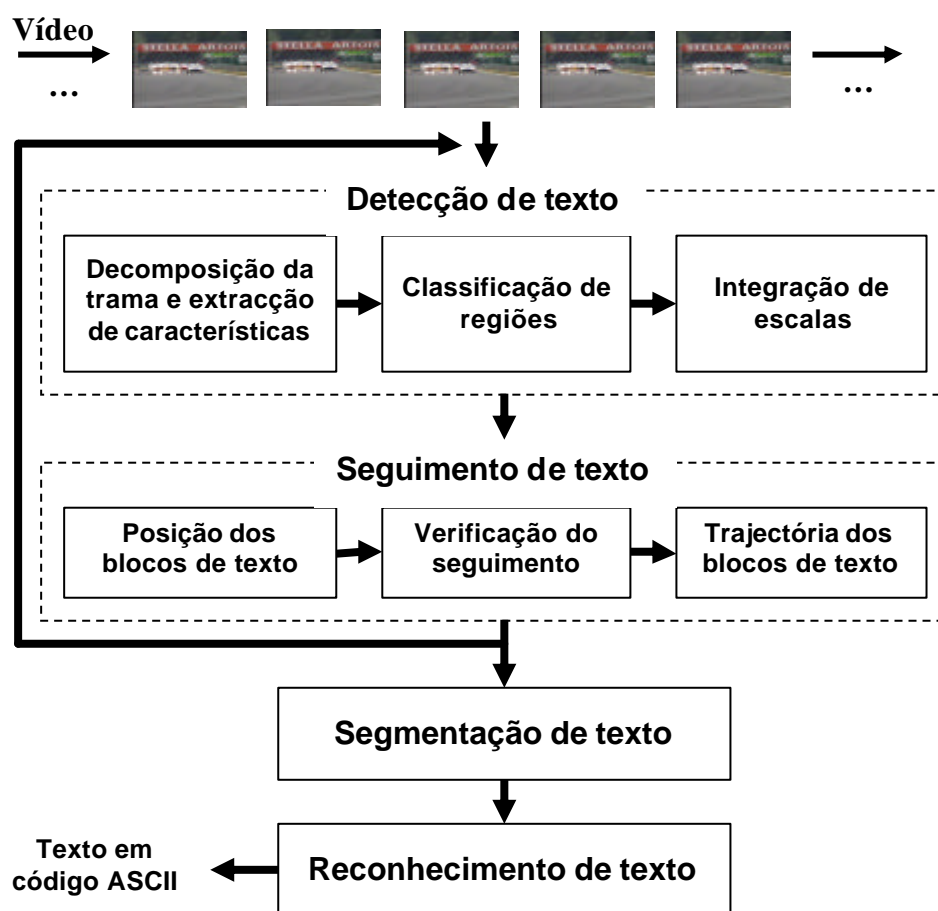


Figura 2.35 – Arquitectura do sistema de extracção de texto em sequências de vídeo com integração de múltiplas tramas proposto em [Li02].

As quatro fases pelas quais passa a extracção de texto no sistema proposto em [Li02] são descritas de seguida.

2.6.4.1 Detecção de Texto

A primeira fase consiste na detecção do texto existente nas tramas individuais para formar os blocos de texto. Aos blocos de texto formados na fase de detecção de texto chama-se ao longo da descrição do sistema *blocos de texto de referência* para que possam ser distinguidos daqueles formados através do seguimento do texto. Assim, e de forma a reduzir a complexidade do método, é explorado o facto do texto estar presente no vídeo durante várias tramas consecutivas. Por este facto, a sua detecção é feita periodicamente e a fase de seguimento é aplicada no intervalo entre cada trama onde é usado o processo de detecção. A Figura 2.36 ilustra graficamente como estes dois processos se encadeiam.

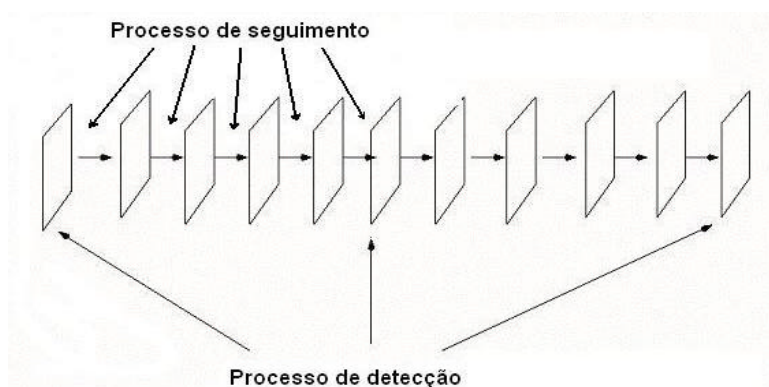


Figura 2.36 – Processos de detecção e seguimento de texto para sequências de vídeo [Li02].

Este sistema usa uma rede neuronal híbrida *wavelets*/neuronal previamente treinada para efectuar a detecção e classificação do texto existente nas tramas de vídeo. Com o intuito de facilitar a detecção de texto de vários tamanhos, foi utilizada uma pirâmide de imagens com três andares, gerada a partir da imagem original, variando a sua resolução espacial em cada nível. A detecção das regiões de texto é efectuada para cada nível e posteriormente extrapolada para a resolução original. Um diagrama esquemático do método de detecção de texto é apresentado na Figura 2.37.

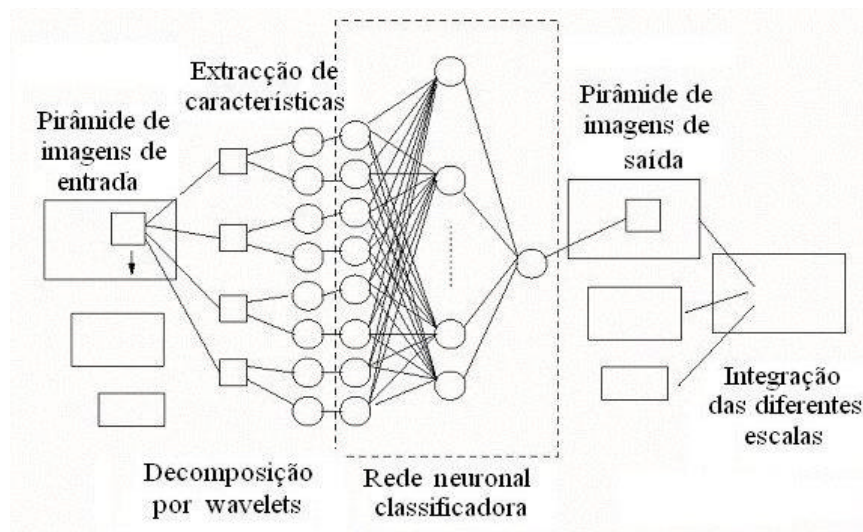


Figura 2.37 – Arquitectura do detector de texto nas tramas de vídeo [Li02].

A detecção do texto existente em cada trama de vídeo decorre em três passos: decomposição da trama e extracção de características, classificação das regiões e integração das escalas. Estes passos são de seguida descritos de uma forma mais pormenorizada:

1º Decomposição da trama e extracção de características – Para decompor as tramas de vídeo em várias resoluções espaciais, usa-se a transformada *wavelet* ou *wavelets*. A transformada *wavelet* é uma transformada que decompõe a imagem em várias componentes frequenciais. Esta decomposição é efectuada por filtragens sucessivas usando-se filtros de *wavelet* passa-baixo e passa-alto. A filtragem é feita para toda a imagem e para cada banda de frequência existe uma versão filtrada da imagem cuja resolução espacial está associada à frequência da filtragem. O resultado da aplicação desta transformada é uma representação hierárquica da imagem onde a cada nível está associada a informação de uma banda de frequência. Como as regiões de texto apresentam uma elevada actividade nas altas frequências, as *wavelets* permitem detectar as fronteiras das regiões de texto uma vez que produzem coeficientes elevados nas mesmas. Neste sistema, as tramas são decompostas em três sub-bandas de alta-frequência.

Para cada sub-banda da trama de vídeo são extraídas as suas características. Como características são utilizados os momentos geométricos ou invariantes dos coeficientes *wavelet*. Estes momentos invariantes baseiam-se nos momentos centrais de primeira, segunda e terceira ordem. Para um bloco I de dimensões $N \times N$, os momentos centrais de primeira, segunda e terceira ordem são dados pelas expressões seguintes:

$$m_0(I) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} I(i, j) \quad (2.12)$$

$$m_2(I) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I(i, j) - m_0(I))^2 \quad (2.13)$$

$$m_3(I) = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (I(i, j) - m_0(I))^3 \quad (2.14)$$

2º Classificação de regiões – No processo de classificação do texto existente nas tramas de vídeo, utiliza-se uma janela (tipicamente com 16×16 *pixels*) para efectuar o varrimento de cada sub-banda da trama de vídeo. A classificação de cada janela como texto ou não texto é feita através de uma rede neuronal previamente treinada, tendo à entrada os valores dos momentos acima calculados.

Quando uma janela é classificada como texto, todos os *pixels* que a constituem são marcados como texto; os *pixels* que não fazem parte de nenhuma janela de texto são marcados como não texto. Em torno das áreas formadas pelos *pixels* que foram marcados como texto e dos *pixels* que lhes são conexos são formadas *bounding boxes*.

3º Integração de escalas – Uma vez concluída a detecção de texto para os vários níveis de resolução, as *bounding boxes* nas várias escalas são mapeadas para a imagem original para assim formar a imagem que contém os blocos de texto de referência.

2.6.4.2 Seguimento de Texto

Sempre que é detectado texto numa trama de vídeo, o processo de seguimento é iniciado. O movimento de texto em vídeo pode ser de três tipos: estático, i.e. ausência de movimento, movimentos lineares (por exemplo, *scrolling*) e movimentos não lineares (por exemplo, *zoom in/out*, rotação e movimentos livres de texto). Nos dois primeiros casos, uma simples técnica de emparelhamento (*matching*) de imagens consegue fazer o seguimento do texto; no terceiro caso, uma técnica mais evoluída é necessária. Neste sistema foi utilizado um método baseado na *Sum of Squared Differences (SSD)*. Assim, o seguimento do texto é efectuado em três passos:

1º Posição dos blocos de texto – Neste passo é efectuado o seguimento de cada bloco de texto, utilizando-se para tal a SSD. Na técnica SSD considera-se que um bloco de texto pode ser tratado como um conjunto de *pixels* S fechado, onde a sua *bounding box* corresponde ao seu limite. O limite B e o interior I de S fornecem informação complementar: B determina a forma e tamanho de S e I o seu conteúdo em termos de intensidade e textura. Assim, a técnica *SSD* analisa o interior I na trama n e verifica a sua semelhança com a imagem da trama $n+1$ para poder determinar a posição da *bounding box* correspondente nessa trama.

Supondo que a matriz de intensidade de um bloco de texto de referência (blocos de texto detectados nas tramas onde foi efectuada a detecção de texto) é representada por $I(x,y)$, a sua posição na trama corrente pode ser determinada através da procura da posição onde a comparação entre as duas matrizes de intensidade (a do bloco de referência e a da trama corrente) apresente uma SSD mínima. A comparação é efectuada num espaço de procura de raio r centrado na posição estimada para o bloco na trama corrente. Para estimar a posição do espaço de procura é utilizada uma técnica de predição que consiste no seguinte: suponha-se que $\overrightarrow{x_n}$ e $\overrightarrow{x_{n-1}}$ são as posições dum bloco de texto na trama corrente e na anterior, respectivamente; a posição do bloco de texto na posição seguinte $\overrightarrow{x_{n+1}}$ é dada pela expressão (2.15).

$$\overrightarrow{x_{n+1}} = \overrightarrow{x_n} + (\overrightarrow{x_x} - \overrightarrow{x_{n-1}}) \quad (2.15)$$

Esta técnica funciona somente para o seguimento de texto estático ou com movimento linear. Contudo quando se analisa uma sequência de vídeo não se sabe que tipo de movimento possui o texto. Todavia, o emparelhamento baseado na SSD retorna sempre a posição com a SSD mínima (até mesmo quando o bloco emparelhado não corresponde a texto). Assim, torna-se necessário medir o nível de confiança do seguimento efectuado.

2º Verificação do seguimento – Neste passo verifica-se o nível de confiança do seguimento efectuado na etapa anterior, i.e. se o seguimento está correcto ou não. Para tal, é medida a desigualdade entre os blocos de texto através do *Mean Square Error* (MSE). O MSE entre dois blocos de texto é definido pela seguinte expressão (2.16):

$$MSE(I_i, I_j) = \frac{1}{w \times h} \sqrt{\sum_{x=0}^w \sum_{y=0}^h (I_i(x, y) - I_j(x, y))^2} \quad (2.16)$$

Onde I_i , I_j são as matrizes de intensidade dos dois blocos de texto e w , h são as dimensões dos mesmos (em *pixels*). Dois tipos de MSE são calculados: o primeiro MSE_b é calculado entre as tramas n e $n+1$; o segundo MSE_r é calculado entre a trama de referência (usualmente a primeira) e a trama corrente. Deste modo, pode analisar-se a rapidez com que os blocos de texto variam, bem como quantificar a variação do bloco de texto em relação à sua referência. A Figura 2.38 ilustra as curvas obtidas para MSE_r e MSE_b para três sequências de vídeo: o texto da primeira sequência possui movimento de *scrolling* para cima e um fundo simples; o texto da segunda sequência possui movimento de *scrolling* para cima e um fundo complexo; finalmente, a última sequência inclui jogadores de futebol com texto nas camisolas que deve ser seguido.

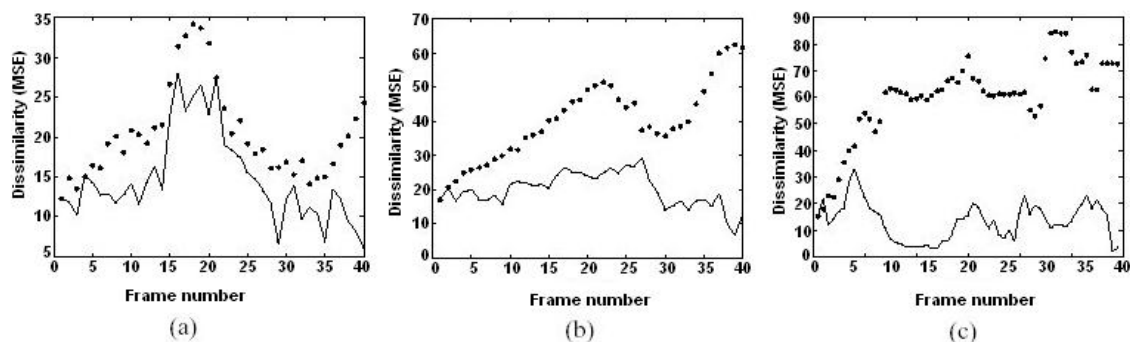


Figura 2.38 – Exemplos das curvas obtidas para MSE_r e MSE_b para três sequências de vídeo. Linha de estrelas (*) representa a desigualdade entre a trama de referência e a trama corrente (MSE_r). A linha sólida (–) representa a desigualdade entre a trama n e a trama $n+1$ (MSE_b). (a) ilustra o seguimento do texto numa sequência de vídeo com o fundo simples, (b) ilustra o seguimento do texto numa sequência de vídeo com o fundo complexo e (c) ilustra o seguimento dos números nas camisolas de jogadores de futebol [Li02].

Nas duas primeiras sequências (Figura 2.38 (a) e (b)) os blocos de texto foram correctamente seguidos, os valores de MSE_r e MSE_b são pequenos. Na última sequência (Figura 2.38 (c)) o algoritmo não consegue efectuar o seguimento dos blocos de texto, os valores de MSE_r e MSE_b são muito elevados quando comparados com os das duas

primeiras sequências. Para qualquer uma das três sequências o valor de MSE_b é sempre menor do que o valor de MSE_r , tal deve-se ao facto de no vídeo, a diferença entre duas tramas consecutivas ser usualmente pequena; assim, MSE_b é muito pequeno, quer quando o texto é correctamente seguido, quer quando não é. Pelo contrário, MSE_r é muito melhor como medida de confiança uma vez que representa a diferença entre o bloco de texto na primeira trama (bloco de texto de referência) e o bloco de texto ao longo das várias tramas onde é feito o seu seguimento. Como medida de confiança m_r entre os blocos de texto seguidos e os blocos de referência é utilizado a soma de todos os MSE_r ou seja

$$m_r = \sum_{i=1}^N MSE_r(i) \quad (2.17)$$

Para verificar se o texto é correctamente seguido, foram definidos empiricamente dois valores de limiar Th_1 e Th_2 . Se m_r for muito pequeno (ver Figura 2.38(a)), nomeadamente se $m_r < Th_1$, assume-se que o texto é correctamente seguido. Pelo contrário, se m_r for muito grande (ver Figura 2.38(c)), nomeadamente $m_r > Th_2$, conclui-se que o texto não é correctamente seguido. No caso de m_r se situar entre Th_1 e Th_2 , é necessária outra medida de confiança.

3º Trajectória dos blocos de texto – Neste passo é traçado o caminho que um bloco de texto efectua ao longo das várias tramas durante a sua existência. A trajectória do bloco pode ser utilizado como uma medida de confiança suplementar para o seguimento no caso de m_r se situar entre Th_1 e Th_2 . Este caminho é definido como a sequência dos pontos centrais dos blocos de texto ao longo do tempo:

$$(X_1, Y_1), (X_2, Y_2), \dots, (X_N, Y_N) \quad (2.18)$$

onde (X_k, Y_k) são as coordenadas do centro do bloco de texto k .

Quando o texto possui um movimento rectilíneo, a sua trajectória é usualmente muito regular e previsível (uma linha recta ou um ponto para texto estático). Quando o seguimento falha, a trajectória varia de forma aleatória e em direcções imprevisíveis. Quando a trajectória é uma linha recta ainda que m_r seja grande, o seguimento do texto é considerado válido. Na Figura 2.39 (a) e (b) são ilustradas as trajectórias correspondentes às sequências das Figura 2.38(b) e Figura 2.38(c), respectivamente. No primeiro caso, em que o algoritmo efectuou correctamente o seguimento do texto, a trajectória dos blocos de texto varia de uma forma previsível. No segundo caso, em que o algoritmo não conseguiu efectuar o seguimento do texto, a trajectória dos blocos de texto varia de forma imprevisível.

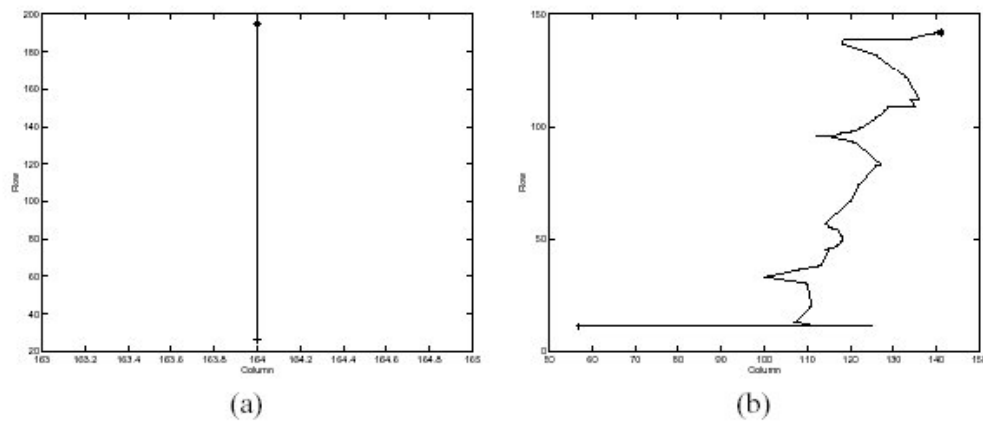


Figura 2.39 – Exemplos de trajetórias: (a) correspondente à sequência da Figura 2.38(b) e (b) correspondente à sequência da Figura 2.38(c) [Li02].

De modo a quantificar o nível de confiança da trajetória, foi utilizada a linha recta $y = ax + b$ para aproximar a trajetória. Para um caminho com N pontos correspondentes ao centros dos blocos de texto, o vector $\vec{P} = (a, b)'$ pode ser estimado como:

$$\vec{P} = (X'X)^{-1} X'Y \text{ onde } X = \begin{pmatrix} X_1 & 1 \\ X_2 & 1 \\ \dots & \dots \\ X_N & 1 \end{pmatrix} \text{ e } Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_N \end{pmatrix} \quad (2.19)$$

Para medir o erro na aproximação entre a trajetória e a linha recta utiliza-se a métrica *Least Mean Square* (LMS). Depois de (a, b) ser estimado pela equação 2.19, o erro LMS é dado pela expressão:

$$err = \frac{1}{N} \sum_{i=1}^N (Y_i - a \times X_i - b)^2 \quad (2.20)$$

A trajetória é uma linha recta se $err = 0$; este é o caso da Figura 2.39(a). Para a Figura 2.39(b), err é muito grande; sempre que err aumenta, isso significa que a trajetória se afasta cada vez mais da linha recta.

Quando não se consegue decidir o nível de confiança unicamente com m_r , utiliza-se err como medida suplementar. Para tal, define-se um terceiro valor de limiar Th_3 . Sempre que m_r se situar entre Th_1 e Th_2 e $err < Th_3$, considera-se o texto correctamente seguido. Th_1 , Th_2 e Th_3 são determinados empiricamente através de testes exaustivos.

2.6.4.3 Segmentação de Texto

Depois de detectado o texto existente nas tramas do vídeo, torna-se necessário separar os caracteres do fundo. Esta separação tem como objectivo facilitar o reconhecimento do texto por parte dos sistemas OCR comerciais, atendendo a que muitos destes sistemas só

reconhecem caracteres em imagens binárias com o texto representado por *pixels* pretos e o fundo representado por *pixels* brancos. Assim, antes de se aplicar um sistema OCR às tramas do vídeo processadas torna-se necessário:

- **Aumentar a resolução da imagem** – O aumento da resolução da imagem torna-se necessário para imagens com baixa resolução com vista a que a resolução dos caracteres mais pequenos aumente por forma a obterem-se melhores resultados no processo de reconhecimento. A baixa resolução muitas vezes existente nos vídeos é a maior fonte de problemas no reconhecimento de texto por parte dos sistemas OCR. Assim, os blocos de texto de referência são sobre-amostrados com um factor 2 ou seja a sua resolução é duplicada, usando um polinómio de interpolação bilinear. Este factor foi considerado como um bom compromisso entre o custo de computação e o desempenho obtido com os sistemas OCR;
- **Identificar o texto normal e inverso** – O esquema usado para identificar o texto inverso e o texto normal passa pelo cálculo de um valor de limiar da intensidade Th . Considera-se para tal que o fundo ocupa a maior parte do histograma da imagem. Assim, faz-se a decisão entre fundo e não fundo através da comparação do valor de limiar Th com o valor da intensidade do fundo Bg que corresponde à maior parte do histograma:
 - ◆ se $Th > Bg$ está-se na presença de texto normal;
 - ◆ se $Th < Bg$ está-se na presença de texto inverso.

Com vista a optimizar a binarização dos blocos de texto, foram utilizados limiares adaptativos locais uma vez que nem todos os blocos de texto têm o mesmo contraste em relação ao fundo. Deste modo, para variar de forma adequada o valor de limiar Th ao longo da imagem durante o processo de binarização da mesma, foi utilizado o método de *Niblack* modificado [Niblack86]; este método permite calcular um valor de limiar específico para cada bloco de texto. Todavia a sua eficácia não é a melhor para áreas com texturas pouco complexas. Na Figura 2.40 ilustra-se o resultado da binarização de um bloco de texto utilizando um limiar global e um limiar adaptativo. Através da observação da Figura 2.40, pode verificar-se que a utilização de um valor de limiar adaptativo (Figura 2.40 (c)), torna o processo de binarização do bloco de texto mais eficaz uma vez que possibilita a definição de um valor de limiar para a binarização de cada *pixel* em função dos valores de luminância dos *pixels* na sua vizinhança. Por outro lado, a definição de um valor de contraste global para o bloco de texto pode resultar numa binarização eficaz para determinadas regiões do bloco de texto mas pouco eficaz para outras (Figura 2.40 (b)).

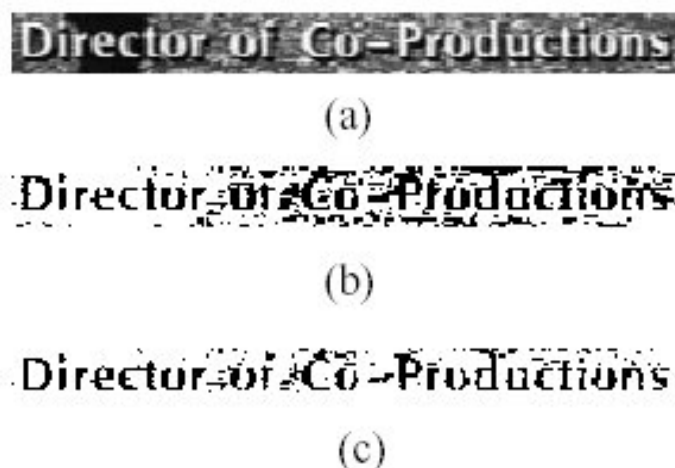


Figura 2.40 – Exemplos de binarização: (a) bloco de texto extraído de uma trama de vídeo; (b) binarização com Th global da imagem em (a); (c) binarização com Th adaptativo utilizando o método de Niblack's da imagem em (a)[Li02].

Durante o processo de binarização da imagem, os *pixels* correspondentes ao texto são representados a preto e os correspondentes ao fundo são representados a branco como requerido por alguns sistemas OCR.

2.6.4.4 Reconhecimento de Texto

Depois de terminado o pós-processamento dos blocos de texto com o objectivo de os converter em imagens binárias com o texto representado a preto e o fundo a branco, é aplicado às imagens binárias um sistema OCR comercial que reconhece o texto existente nestas e o converte para código ASCII. Neste sistema é utilizado o OCR Xerox TextBridge Pro98.

2.6.4.5 Avaliação do Desempenho

Para efectuar a avaliação do desempenho do sistema foram utilizadas cinco sequências de vídeo. Os conteúdos são de três tipos diferentes: filmes, comerciais e programas informativos. O reconhecimento do texto é efectuado sobre um único bloco de texto para cada objecto de texto. Para representar cada objecto de texto, foi escolhido o bloco de texto correspondente à trama do meio do objecto de texto.

Para mostrar o aumento de desempenho proporcionado pela análise do movimento, foi avaliado o desempenho do algoritmo para uma única trama, i.e. foi aplicada a fase de segmentação imediatamente a seguir à fase de detecção de texto, a uma trama da sequência de vídeo onde o texto existente nesta esteja presente. Deste modo, torna-se possível verificar o aumento de desempenho proporcionado pela análise do movimento.

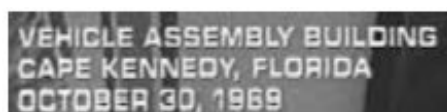
Os resultados obtidos no reconhecimento do texto para os vários tipos de conteúdos são apresentados na Tabela 2.9 .

Tabela 2.9 – Desempenho em termos de reconhecimento de texto para os vários tipos de conteúdos.

Tipo de vídeo	Tipo de movimento	Nº de tramas do vídeo	Nº de blocos de texto no vídeo	Nº de caracteres no vídeo	Caracteres correctamente reconhecidos pelo OCR	Caracteres correctamente reconhecidos pelo OCR (utilizando uma única trama)
Comerciais	Texto estático Fundo movimento	50	1	54	52 (98%)	46 (86%)
Programa de informação 1	Texto estático Fundo estático	120	3	27	17 (64%)	16 (63%)
Programa de informação 2	Texto estático Fundo estático	30	2	38	29 (76%)	29 (76%)
Filme1	Texto movimento Fundo movimento	200	23	305	274 (89.4%)	189 (62.2%)
Filme2	Texto movimento Fundo movimento	250	35	515	453 (87.3%)	221 (43%)
Total		650	63	939	852 (88%)	501 (53%)

Da análise dos resultados apresentados na Tabela 2.9, pode verificar-se que para o caso das duas sequências de informação onde o texto e o fundo são estáticos não existem diferenças significativas entre o desempenho do sistema quando é efectuada a extracção do texto utilizando uma única trama ou várias tramas, i.e. quando não existe movimento do texto ou do fundo o valor acrescentado da análise de movimento é praticamente nulo. Todavia, quando existe movimento do texto e/ou do fundo, existe uma melhoria do desempenho em termos de precisão quando a extracção é feita utilizando várias tramas.

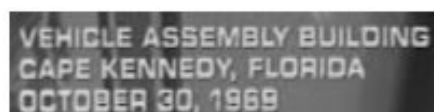
Na Figura 2.41 são ilustrados dois exemplos do aumento da precisão do reconhecimento para a situação em que o texto e/ou o fundo possuem movimento, quando se utilizam várias tramas na extracção do texto.



**VEHICLE ASSEMBLY BUILDING
CAPE KENNEDY, FLORIDA
OCTOBER 30, 1969**

V_\$ICLE ASSEMBLY BUILDING
CAb2E KENNEDY, FLORIDA
OCTOBER 3D~ 1969

(a)



**VEHICLE ASSEMBLY BUILDING
CAPE KENNEDY, FLORIDA
OCTOBER 30, 1969**

VEHICLE ASSEMBLY BUILOING
CAPE KENNEDY , FLORIDA
OCTOBER 30, 1969

(b)



Figura 2.41 – Exemplos do aumento da precisão do reconhecimento quando se utilizam várias tramas na extracção do texto: (a) resultado do OCR utilizando uma trama quando o texto é estático e o fundo possui movimento; (b) resultado do OCR utilizando varias tramas quando o texto é estático e o fundo possui movimento; (c) resultado do OCR utilizando uma trama quando o texto e o fundo possuiu movimento e (d) resultado do OCR utilizando várias tramas quando o texto e o fundo possuem movimento [Li02].

O sistema proposto em [Li02] e aqui descrito utiliza, tal como o sistema anterior, um classificador neuronal para efectuar a extracção de texto. Este pode ser gráfico e/ou de cena e estar escrito em qualquer direcção (não foram apresentados resultados para texto de cena inclinado). Para efectuar o seguimento do texto, é utilizada uma técnica que se baseia na comparação de tramas sucessivas. Sempre que uma palavra ou uma linha de texto nova surge, é gerado um bloco de texto de referência, sendo a sua posição procurada ao longo das várias tramas. Todavia, a detecção de texto é feita ao nível do bloco de texto, podendo um bloco de texto ser formado por várias linhas do mesmo que se encontrem próximas. Por este motivo, podem surgir problemas no processo de seguimento se as linhas de texto não se moverem homogeneamente. Esta solução não faz a rotação do texto inclinado para a horizontal, tornando deste modo o seu reconhecimento praticamente impossível pelos sistemas OCR comerciais.

2.7 Comentários Finais

Ao longo deste capítulo fez-se, ainda que sumariamente, a apresentação de vários sistemas e técnicas disponíveis na literatura visando a extracção automática de texto em imagem e vídeo. As técnicas básicas foram apresentadas de acordo com os módulos definidos para uma arquitectura básica de extracção de texto, nomeadamente segmentação, classificação, seguimento e reconhecimento.

A segmentação de imagem é, na maior parte das aplicações que visam a extracção de texto em vídeo ou imagens, um passo preliminar e essencial. Embora muitas técnicas existam para esse efeito, a escolha de uma em detrimento de outra está intimamente relacionada com o tipo de características do conteúdo em questão, bem como com o tipo de processamento que se pretende, ou pode, adoptar para efectuar a extracção de texto no contexto aplicativo em questão. Na Tabela 2.10 são resumidas as vantagens e desvantagens das várias técnicas de segmentação descritas no presente capítulo.

Tabela 2.10 – Sumário das vantagens e desvantagens das técnicas de segmentação apresentadas.

Técnica de Segmentação		Vantagens	Desvantagens
Espacial	Baseada na Amplitude	<ul style="list-style-type: none"> Baixo custo computacional; Eficazes para imagens simples. 	<ul style="list-style-type: none"> Número elevado de pequenas regiões que resultam das imagens texturadas; Não explora a relação espacial existente entre <i>pixels</i> vizinhos.
	Baseada na Textura	<ul style="list-style-type: none"> Eficientes na detecção de homogeneidades em imagens com texturas de elevada variedade. 	<ul style="list-style-type: none"> Elevado custo computacional.
	Baseada em Fronteiras	<ul style="list-style-type: none"> Custo aceitável para o cálculo dos detectores de fronteira. 	<ul style="list-style-type: none"> Elevada sensibilidade ao ruído; Necessidade de um processamento complexo para produzir fronteiras fechadas.
	Baseada em Regiões	<ul style="list-style-type: none"> Elevada eficiência na identificação de regiões homogéneas em termos de características espaciais; Elevada exactidão na localização das fronteiras. 	<ul style="list-style-type: none"> Elevado custo computacional; Elevado número de regiões que surgem como resultado da segmentação.
Temporal	Baseada na Detecção de Alterações	<ul style="list-style-type: none"> Baixo custo computacional. 	<ul style="list-style-type: none"> Incapacidade de detectar o movimento diferente dos objectos; os objectos com movimento e os objectos estáticos são agrupados em conjunto.
	Baseada em Movimento	<ul style="list-style-type: none"> Elevada eficiência na detecção de regiões homogéneas em termos de movimento. 	<ul style="list-style-type: none"> Incapacidade de detectar objectos estáticos no tempo; Pouca precisão na localização das fronteiras das regiões; Elevado custo computacional.
Espacial e Temporal		<ul style="list-style-type: none"> Elevada eficiência na identificação de regiões homogéneas em termos de características espaciais e/ou temporais; Boa capacidade para efectuar o seguimento de objectos ao longo das sequências de imagens. 	<ul style="list-style-type: none"> Elevado custo computacional.

Na fase de classificação, as várias regiões provenientes da segmentação são classificados como regiões de texto ou não texto. Para levar a cabo tal tarefa, foram descritas várias técnicas que permitem descrever as regiões para posterior classificação. A descrição das regiões tem como principal objectivo a captura das diferenças essenciais existentes entre as várias regiões e deve ser o mais insensível possível a variações de localização, tamanho ou orientação das regiões. Na bibliografia usada, o problema da classificação foi, na maioria dos casos, resolvido recorrendo a redes neuronais previamente treinadas ou a métodos baseados

na análise geométrica das regiões. Na Tabela 2.11 são resumidas as vantagens e desvantagens dos métodos maioritariamente utilizados para efectuar a classificação das regiões no âmbito da detecção de texto.

Tabela 2.11 – Sumário das vantagens e desvantagens dos métodos de classificação maioritariamente utilizados na extracção de texto.

Técnica de Classificação	Vantagens	Desvantagens
Análise Geométrica das Regiões	<ul style="list-style-type: none"> • Facilidade de implementação; • Não necessitam de treino. 	<ul style="list-style-type: none"> • Dificuldades na classificação de texto de vários tamanhos; • Dificuldades na classificação de texto quando os caracteres se tocam entre si;
Redes Neurais	<ul style="list-style-type: none"> • Capacidade de aprender. 	<ul style="list-style-type: none"> • A sua eficácia depende muito do treino dado à rede; • Grande dificuldade em treinar um classificador genérico, uma vez que o texto existente nos vídeos possui vários tamanhos, fontes, estilos, etc.

Na fase de seguimento, as várias regiões provenientes da fase de classificação são seguidas ao longo do tempo de modo a explorar a redundância temporal existente no vídeo para melhorar a sua classificação, efectuada na fase anterior. Para tal, foram descritos dois tipos de abordagens que permitem efectuar o seguimento: emparelhamento entre duas tramas e filtragem temporal recursiva. Na bibliografia usada, o problema do seguimento foi, na maioria dos casos, resolvido recorrendo às técnicas que efectuem o seguimento focando-se em duas tramas de cada vez ou seja relacionam o resultado do momento anterior com o do momento actual. Na Tabela 2.12 são resumidas as vantagens e desvantagens dos dois tipos de abordagens apresentados para efectuar o seguimento das regiões no âmbito da detecção de texto.

Tabela 2.12 – Sumário das vantagens e desvantagens dos dois tipos de abordagens apresentados para efectuar o seguimento do texto.

Técnica de Seguimento	Vantagens	Desvantagens
Emparelhamento Entre Duas Tramas	<ul style="list-style-type: none"> • Facilidade de implementação; 	<ul style="list-style-type: none"> • Dificuldades no seguimento de objectos que não estejam presentes em todas as tramas; • Dificuldades na classificação de texto quando os caracteres se tocam entre si.
Filtragem Temporal Recursiva	<ul style="list-style-type: none"> • Permitem o seguimento de objectos que se ocultem. 	<ul style="list-style-type: none"> • Dificuldade de implementação.

Relativamente ao reconhecimento das regiões que foram classificadas como texto na fase anterior, as técnicas apresentadas permitem perceber qual o ‘estado da arte’. O campo do reconhecimento tem sido objecto de grande investigação ao longo dos anos mais recentes e numa grande variedade de disciplinas. Como resultado, muitos artigos e muitos livros foram publicados, documentando um impressionante número de abordagens, umas mais teóricas, outras mais heurísticas, para os vários aspectos da análise automática de imagens.

Ainda neste capítulo, foram descritos vários sistemas completos considerados mais representativos e utilizando muitas das tecnologias apresentadas; estes sistemas oferecem várias soluções para o problema da extracção de texto em imagens e sequências de vídeo. Para cada um dos sistemas apresentados, foi analisada a sua arquitectura básica, as suas vantagens e desvantagens bem como as limitações impostas aos conteúdos a analisar e ao texto a extrair. Na Tabela 2.13 apresenta-se um resumo das características supracitadas para cada um dos sistemas analisados.

Tabela 2.13 – Resumo das características dos sistemas de extracção de texto apresentados

Extracção de Texto Gráfico e de Cena em Imagens [Messelodi99]	
Caracterização do Texto	<ul style="list-style-type: none"> • Caracteres de diferentes tamanhos, fontes e estilos; • Caracteres orientados em qualquer direcção; • Caracteres que façam parte da mesma palavra devem possuir todos a mesma cor; • Caracteres têm limitações em termos de tamanho; um caracter não pode ser maior nem menor que um dado número de <i>pixels</i> pré-definido; • Ocorrências de texto só são consideradas se existirem pelo menos dois caracteres seguidos.
Vantagens	<ul style="list-style-type: none"> • Capacidade para identificar texto constituído por linhas orientadas em qualquer direcção; • Texto pode ser de cena ou gráfico e pode ainda caracterizar-se por diferentes tamanhos, fonte e estilos.
Desvantagens	<ul style="list-style-type: none"> • Só funciona para sequências de vídeo tomadas como uma sequência de imagens independentes não explorando a redundância temporal existente no vídeo.
Extracção de Texto Gráfico em Sequências de Vídeo [Lienhart00]	
Caracterização do Texto	<ul style="list-style-type: none"> • Caracteres devem encontrar-se no primeiro plano e nunca parcialmente ocultos; • Caracteres que fazem parte da mesma palavra ou linha de texto devem possuir todos a mesma cor; • Caracteres não devem alterar a sua cor, forma, tamanho e orientação de trama para trama; • Caracteres têm limitações em termos de tamanho; um caracter não pode ser maior nem menor que um dado número de <i>pixels</i>; • Caracteres devem ser estacionários ou ter movimentos lineares; os movimentos lineares devem ser horizontais ou verticais; • Caracteres devem permanecer na imagem durante várias tramas consecutivas; • Ocorrências de texto só são consideradas se existirem pelo menos três caracteres seguidos.

Vantagens	<ul style="list-style-type: none"> • Funciona para sequências de vídeo e explora a redundância temporal existente para melhorar a eficiência; • Eficaz na detecção de texto gráfico.
Desvantagens	<ul style="list-style-type: none"> • Pouco eficaz na detecção de texto de cena, uma vez que explora, o alto contraste do texto em relação ao fundo e a textura simples do texto existente no vídeo (típicas de texto gráfico); • Só detecta texto escrito na horizontal; • Dificuldade na detecção de texto quando os caracteres se tocam.
Extracção de Texto em Imagens e Vídeos [Lienhart02]	
Caracterização do Texto	<ul style="list-style-type: none"> • Somente texto horizontal é considerado uma vez que este ocorre em mais de 99% dos casos de texto gráfico; • Ocorrências de texto só são consideradas se existirem pelo menos dois caracteres seguidos; • Só o texto cuja cor para uma mesma linha não se altera ao longo do tempo é considerado; • Caracteres que fazem parte da mesma palavra ou linha de texto devem possuir todos a mesma cor.
Vantagens	<ul style="list-style-type: none"> • Funciona quer para vídeo, quer para imagens isoladas; • No caso do vídeo faz o seguimento do texto após a sua detecção e aproveita a redundância temporal existente para aumentar a eficiência da detecção de texto; • Funciona suficientemente bem, quer com resoluções baixas, quer com resoluções elevadas.
Desvantagens	<ul style="list-style-type: none"> • Definição de um valor de limiar global na fase de binarização, quando poderiam ser definidos valores de limiar adaptativos para cada <i>bounding box</i>; • Só detecta texto escrito na horizontal; • Pouco eficaz na detecção de texto de cena quando este possui um baixo contraste.
Extracção de Texto em Sequências de Vídeo com Integração de Múltiplas Tramas [Li02]	
Caracterização do Texto	<ul style="list-style-type: none"> • Texto pode ser gráfico ou de cena; • Caracteres de diferentes tamanhos, fontes e estilos; • Caracteres orientados em qualquer direcção; • Caracteres que fazem parte da mesma palavra ou linha de texto devem possuir todos a mesma cor.
Vantagens	<ul style="list-style-type: none"> • Faz a detecção e o seguimento quer de texto de cena, quer de texto gráfico; • Texto pode ter diferentes tamanhos, fontes, estilos e orientações (não foram apresentados resultados para texto inclinado); • São definidos valores de limiar adaptativos para cada bloco de texto na fase de binarização das tramas.
Desvantagens	<ul style="list-style-type: none"> • A detecção de texto é feita ao nível do bloco, podendo um bloco de texto ser formado por mais do que uma linha desde que estas se encontrem próximas. Se as linhas de texto pertencentes ao mesmo bloco possuírem movimentos diferentes, é impossível fazer o seu seguimento; • Apresenta dificuldades em fazer o seguimento do texto quando o seu movimento é brusco; • Não faz a rotação do texto inclinado para a horizontal, tornando deste modo o seu reconhecimento praticamente impossível para os sistemas OCR comerciais.

Uma vez que a extracção de texto em imagens e sequências de vídeo é um problema complexo para o qual não existe uma técnica adequada para todos os tipos de conteúdos e situações, a sua solução passa muitas vezes pela combinação de várias técnicas, aproveitando as vantagens de cada uma, de forma a obter uma solução adequada às necessidades das várias aplicações. Todavia, continua a ser de fundamental interesse o aperfeiçoamento das técnicas já existentes e o desenvolvimento de novas soluções com o objectivo final de superar as dificuldades até agora sentidas. É neste contexto que se insere o trabalho desenvolvido nesta Tese e apresentado nos capítulos seguintes.

Capítulo 3

Algoritmo Para Extracção de Texto em Imagens

O principal objectivo deste capítulo é a apresentação do algoritmo desenvolvido nesta Tese para efectuar a detecção de texto em imagens. Esta detecção baseia-se na segmentação espacial das imagens, na análise do contraste existente entre o texto e o fundo da imagem, na análise geométrica do texto e, ainda, no posicionamento do mesmo na imagem. O texto detectado nas imagens através do processo proposto neste capítulo pode ter várias aplicações, nomeadamente acrescentar uma componente semântica à descrição do vídeo correspondente usando eventualmente os descritores adequados da norma MPEG-7 [Manjunath02]. A componente semântica acrescentada pode revelar-se útil para procura de vídeo, navegação automática ou ainda vigilância. No primeiro caso, a componente semântica do texto existente nas imagens pode ser utilizada para efectuar a procura em bases de dados de imagens ou vídeo digitais. No segundo caso, a componente semântica correspondente ao texto existente nas placas da estrada, nomes de ruas etc. pode servir para em navegação automática por exemplo, confirmar automaticamente determinado percurso. O terceiro exemplo pode revelar-se fundamental no seguimento da trajectória de determinada viatura através da verificação da sua matrícula. Para além destes exemplos pode ainda considerar-se que a detecção de texto assume importância na classificação de vídeos, na análise de eventos ou ainda na codificação eficiente do texto como um objecto textual independente.

3.1 Arquitectura Básica

Para que possa ser feita a descrição detalhada do método desenvolvido para a extracção de texto em imagens, é fundamental que, previamente, se apresente a sua arquitectura básica, ou seja, a sequência de processos aplicados à imagem através dos quais se extrai o texto contido na imagem em questão. Várias são, todavia, as alternativas técnicas existentes na literatura para os vários módulos da arquitectura básica tendo as mais importantes sido objecto de

estudo no capítulo 2 da presente Tese. Nesse estudo foram analisadas as suas vantagens e desvantagens bem como as características do texto por elas detectado.

Para a definição da arquitectura básica a ser usada na extracção de texto em imagens, há que ter em conta não só o tipo de imagens a serem analisadas mas também as características do texto que nelas se encontra. Contudo, quando a variedade de texto e de imagens é muito grande, torna-se imperativo conceber um algoritmo que funcione bem para grande parte das situações. Deste modo, a concepção da solução proposta neste capítulo foi influenciada por algumas das técnicas apresentadas no capítulo 2, nomeadamente em termos das suas capacidades relativamente os tipos de texto detectados. As soluções aqui propostas quer para a segmentação, quer para a detecção de caracteres, foram sobretudo influenciadas pelos sistemas baseados em componentes conexos estudados na secção 2.6 [Messelodi99, Lienhart00]. Para ambos estes módulos, partiu-se de técnicas conhecidas, tendo-se introduzido alterações de modo a alargar a sua gama de aplicação e a melhorar o seu desempenho na detecção de texto. Por exemplo, na fase de segmentação foram desenvolvidas técnicas para melhorar a precisão das fronteiras das regiões conexas detectadas. Na detecção de caracteres foi combinada a detecção de fronteiras com a análise de contraste com o objectivo de melhorar a sua eficiência para imagens pouco contrastadas.

Para diminuir a influência de alguns efeitos indesejáveis no desempenho final do processo de extracção de texto, foi proposta uma técnica para a simplificação das imagens que preserva as zonas de elevado contraste (normalmente correspondentes a regiões de texto). Para tal, esta técnica combina a detecção de fronteiras com um filtro de mediana. Para a detecção de palavras foram propostas técnicas que permitem tanto detectar palavras com inclinações compreendidas entre 0 – 90°, como efectuar a sua rotação para a horizontal de modo a poderem ser reconhecidas por sistemas OCR.

Assim, foi adoptada como arquitectura básica para o sistema de extracção de texto em imagens proposto neste capítulo aquela que se apresenta na Figura 3.1.

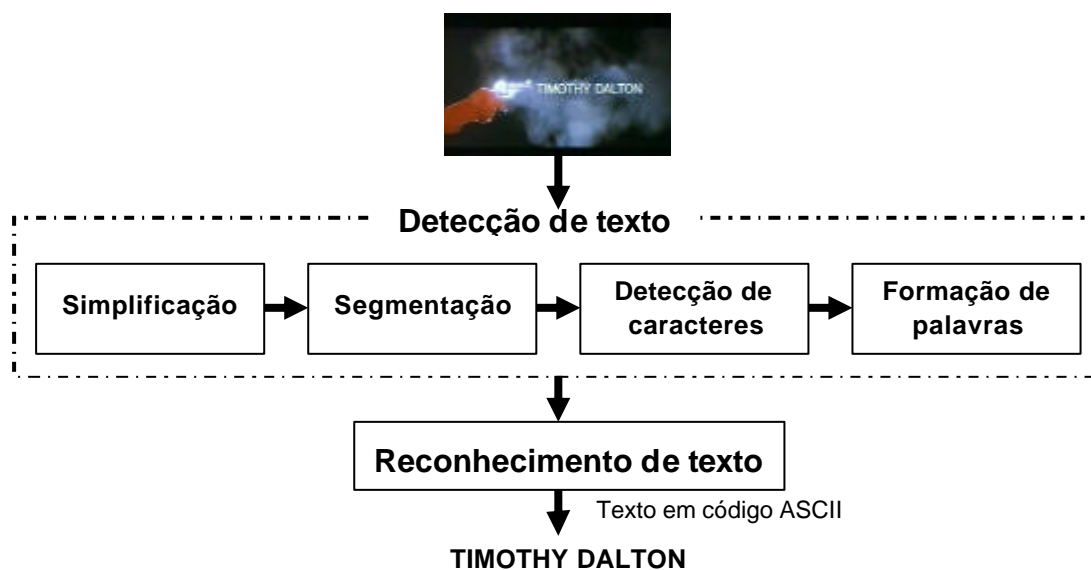


Figura 3.1 – Arquitectura básica proposta para o algoritmo de extracção de texto em imagens.

Como se pode observar na Figura 3.1, a extracção de texto em imagens decorre em duas fases principais bem distintas. A primeira fase visa a detecção do texto, enquanto a segunda fase visa o seu reconhecimento:

- **Detecção de texto** – Esta fase visa a detecção do texto existente nas imagens ou tramas de vídeo e pode ser dividida em quatro fases: simplificação da imagem ou tramas de vídeo, segmentação da imagem ou tramas de vídeo em regiões conexas, detecção de caracteres que correspondam a texto e formação de palavras:
 - ♦ **Simplificação da imagem** – Esta fase visa simplificar e diminuir a influência de alguns efeitos indesejáveis, tais como diferentes gradientes de luminosidade, ruído ou elevado número de cores. Para tal, a imagem é filtrada utilizando um filtro que combina a detecção de fronteiras com um filtro de mediana;
 - ♦ **Segmentação da imagem** – Esta fase visa a divisão da imagem em regiões homogéneas usando como critério o valor da luminância: cada uma destas regiões pode corresponder ou não a um carácter textual;
 - ♦ **Detecção de caracteres** – Esta fase da detecção do texto visa a classificação de cada uma das regiões candidatas provenientes da fase de segmentação como texto ou não texto. Para tal, é efectuada a classificação de cada uma das regiões provenientes da fase anterior, para assim se determinar quais as regiões que correspondem a caracteres de texto. Para este efeito, são utilizadas técnicas que se baseiam na análise do contraste e na geometria das regiões. As regiões que forem classificadas como não texto são descartadas;
 - ♦ **Formação de palavras** – Esta fase da detecção do texto visa o agrupamento das regiões classificadas como caracteres de texto na fase anterior de modo a formar palavras e linhas. Para tal, utilizam-se técnicas que se baseiam na análise espacial das regiões que foram classificadas como texto.
- **Reconhecimento de texto** – Esta fase visa o reconhecimento do texto existente na imagem. Para tal são usadas as regiões candidatas a texto determinadas na fase anterior, utilizando-se para o reconhecimento dois sistemas OCR: um desenvolvido para o caso específico da extracção de texto em imagens ou vídeo [Lienhart95] e o outro correspondente a uma versão comercial do OmniPage Pro 12.0 [ScanSoft].

Estas duas fases da extracção de texto em imagens serão discutidas em pormenor nas secções seguintes.

3.2 Detecção de Texto

A detecção do texto existente nas imagens visa formar um conjunto de regiões conexas classificadas como candidatas a texto e que, para além disso, cumpram os critérios para a formação de palavras. A detecção decorre em quatro fases distintas conforme se apresenta na Figura 3.2: simplificação da imagem, segmentação da imagem em regiões conexas, detecção de caracteres e formação de palavras.

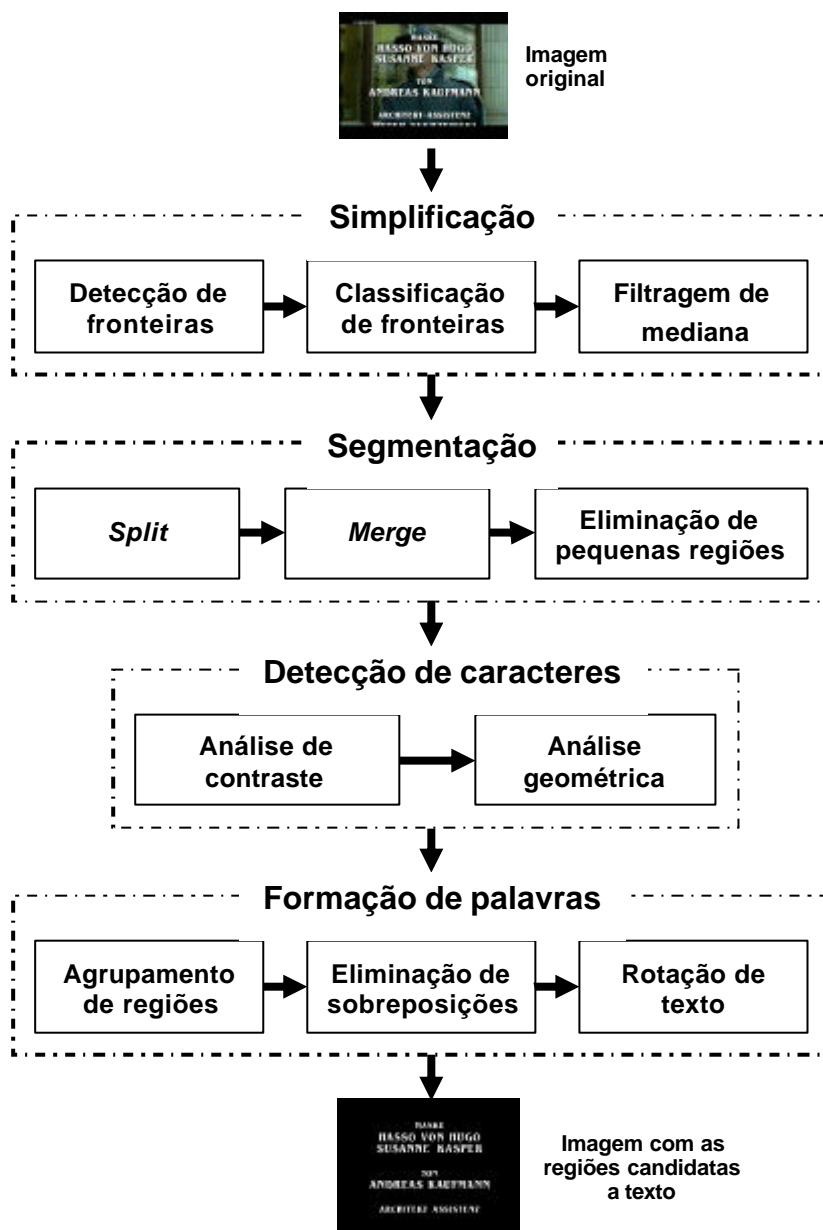


Figura 3.2 – Arquitectura do processo de detecção de texto.

Os vários módulos do algoritmo de detecção de texto em imagens serão discutidos nas secções seguintes.

3.2.1 Simplificação

O módulo de simplificação da imagem tem por objectivo a diminuição da influência, no desempenho final do processo de extracção de texto, de alguns efeitos indesejáveis motivados, por exemplo, pela presença de demasiado ruído na imagem. No âmbito da presente Tese, a simplificação da imagem é feita de forma a preservar o mais possível as fronteiras existentes na mesma. Esta preservação é essencial, no contexto da detecção de texto, uma vez que são essas fronteiras que vão determinar o carácter a ser reconhecido. Para

esse efeito, foi desenvolvido um filtro iterativo que combina a detecção de fronteiras com um filtro de mediana com uma janela de 3×3 pixels. O filtro de simplificação iterativo utiliza o resultado da filtragem da iteração n como imagem de entrada para a iteração $n+1$. Cada uma das iterações da filtragem inclui três etapas principais, tal como se ilustra na Figura 3.3: detecção de fronteiras, classificação de fronteiras e filtragem de mediana.

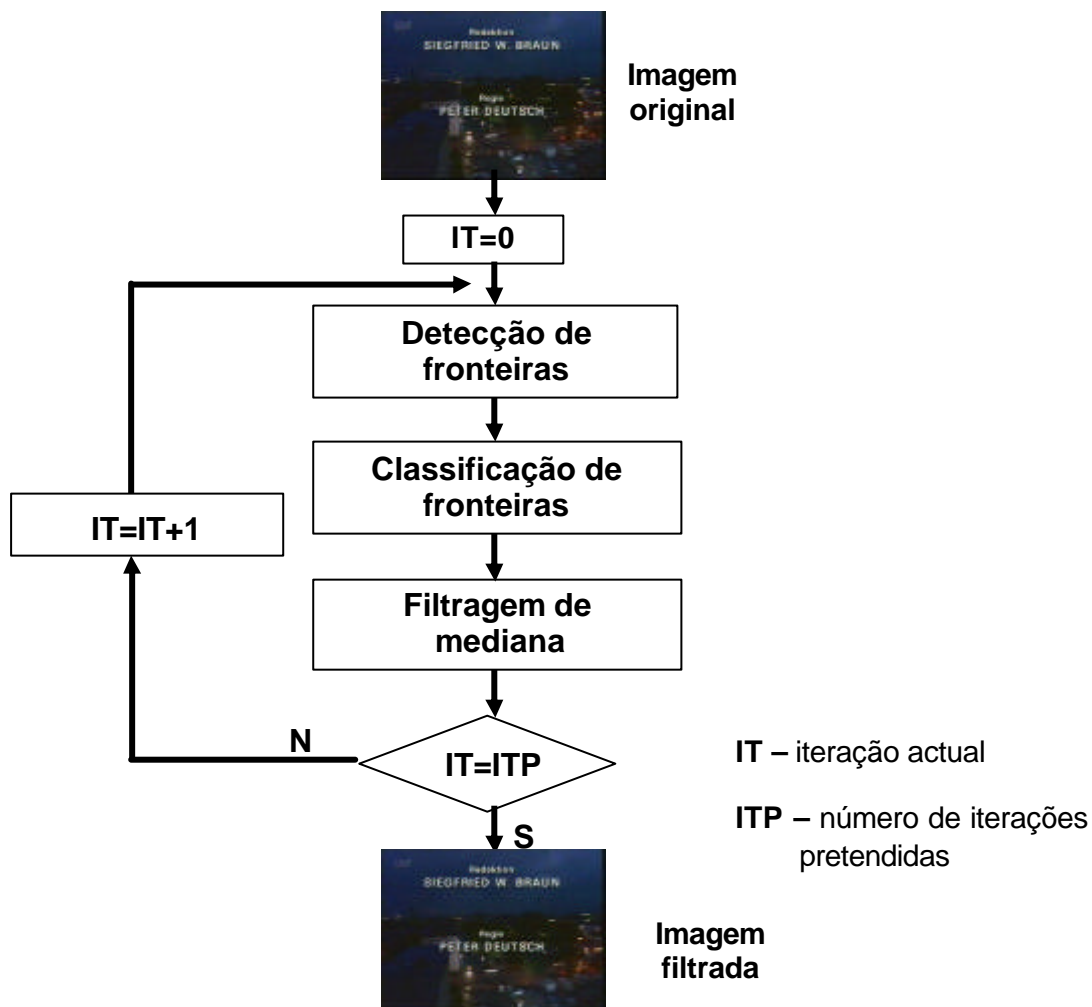


Figura 3.3 – Arquitectura do filtro iterativo para simplificação da imagem.

Descrevem-se de seguida as várias etapas de processamento que definem o filtro iterativo proposto neste capítulo:

1ª Etapa – Detecção de fronteiras

Esta etapa do processamento de simplificação consiste na detecção das fronteiras existentes na imagem original. Para esse fim, foi utilizado um detector de fronteiras baseado no método de Canny [Canny86], ou seja, foi utilizado o método de Canny com coeficientes de derivação otimizados com o objectivo de aumentar a acuidade do método, i.e. torná-lo mais sensível a variações no valor da luminância. Os coeficientes de derivação otimizados são obtidos com recurso à expressão (3.2) proposta em [Jähne97]. Desta forma, o esquema de detecção de fronteiras utilizado é ilustrado na Figura 3.4.

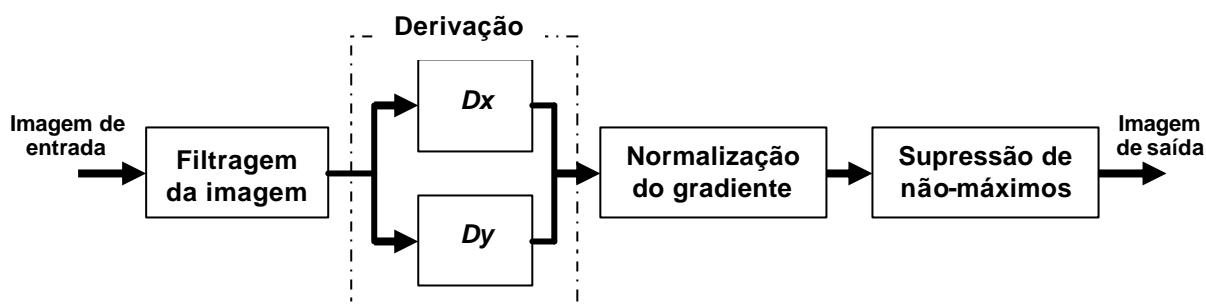


Figura 3.4 – Arquitectura do esquema de detecção de fronteiras de Canny.

Assim, a detecção das fronteiras segundo o método de Canny decorre segundo os seguintes passos:

- 1º **Filtragem da imagem** – No primeiro passo, a imagem original é filtrada para eliminar o ruído nela existente. Para tal é utilizado um filtro de gauss, com o operador (isotrópico) Gaussiano, aproximado por $C+1$ coeficientes de uma máscara binomial de ordem 6 com $2C+1$ coeficientes, $B^6 = \frac{1}{64} [1 \ 6 \ 15 \ 20 \ 15 \ 6 \ 1]$;
- 2º **Derivação da imagem** – No segundo passo, a imagem é derivada de modo a acentuar os contornos das várias regiões. Para tal, são calculadas as derivadas de primeira ordem, usando os gradientes (vertical, Dy , e horizontal, Dx) na vizinhança de cada *pixel*. As zonas da imagem para as quais o cálculo da derivada de primeira ordem produz valores mais elevados, correspondem a descontinuidades, i.e. fronteiras. Os valores de Dx e Dy são obtidos através da expressão (3.1):

$$D_n = G_n * I \quad (3.1)$$

Onde $G_n = \frac{\partial G}{\partial n}$ representa a primeira derivada do operador Gaussiano G na direcção n e $*$ representa a convolução da imagem I com o operador G . O operador G é definido como:

$$G_x = \frac{1}{\sqrt{2p} \cdot s^3} \cdot x \cdot e^{\left(x^2 \left(-\frac{1}{2s^2}\right)\right)} \quad (3.2)$$

Onde $x=1, 2, 3, \dots$ representa os vários coeficientes de derivação e s representa a sensibilidade do detector e está directamente relacionado com o número de coeficientes de derivação da seguinte forma: $Número_{coeficientes} = \text{int}(3 \times s)$ com $s \in]0, 5[$. Assim, para valores de s mais elevados, o método é menos sensível a variações no valor da luminância da imagem. Na presente Tese, utiliza-se $s=1$, valor que foi obtido empiricamente através de testes exaustivos efectuados com vários tipos de imagens;

- 3º **Normalização do gradiente** – No terceiro passo, o gradiente é normalizado através do cálculo da sua magnitude. Assim, para cada *pixel* da imagem, é calculado o valor da magnitude do gradiente através da expressão (3.3):

$$M = \sqrt{Dx^2 + Dy^2} \quad (3.3)$$

- 4º **Localização de fronteiras** – No quarto passo, é aplicada a supressão de não-máximos em inglês (*non-maximal suppression*) à magnitude do gradiente para localizar os segmentos de fronteira mais relevantes de entre todos aqueles detectados no passo anterior, i.e. verifica-se para cada *pixel* se o valor da magnitude, numa vizinhança de 3×3 *pixels*, é um máximo local. Caso tal não se verifique, o valor da magnitude da fronteira é colocado a zero.

2ª Etapa – Classificação de fronteiras

Esta etapa consiste na classificação das fronteiras, resultantes da fase anterior, em fronteiras candidatas a pertencerem a caracteres e fronteiras não candidatas a pertencerem a caracteres. Para se tomar tal decisão, há que adoptar critérios que definam, com exactidão, quais as fronteiras que podem, ou não, pertencer a caracteres. Os critérios adoptados nesta Tese para definir quais as fronteiras que podem pertencer a caracteres são os seguintes:

- **Amplitude da fronteira** – Define como fronteiras candidatas a pertencer a caracteres todas aquelas cuja amplitude do gradiente seja superior a um dado limiar [Jain89]. Nesta Tese adoptou-se como limiar o valor de 5% do gradiente máximo, o que significa que todas as fronteiras cujo valor de gradiente é inferior a 5% do gradiente máximo da imagem são eliminadas. Este valor do limiar foi obtido empiricamente através de testes exaustivos, usando vários tipos de imagens;
- **Comprimento e proximidade da fronteira** – Define como fronteiras não candidatas a pertencer a caracteres, todas aquelas cujo comprimento seja muito curto e, simultaneamente, estejam relativamente isoladas de outras fronteiras:
 - ♦ **Critério de comprimento** – Selecciona para possível eliminação as fronteiras detectadas cujo comprimento (número de *pixels* que formam a fronteira) seja inferior a um dado limiar. Assim, todas as fronteiras cujo comprimento é inferior a 0,004% do tamanho da imagem original, ou seja, 4 *pixels* numa imagem *Common Intermediate Format* (CIF) são seleccionadas para possível eliminação. A definição deste valor está directamente relacionada com o tamanho mínimo do texto que é possível detectar utilizando o algoritmo aqui proposto para a detecção de texto;
 - ♦ **Critério de proximidade** – Selecciona para possível eliminação todas as fronteiras detectadas que não tenham na sua proximidade outras fronteiras. Assim, todas as fronteiras que não tenham na sua proximidade outras fronteiras, por exemplo, a uma distância de dois *pixels* (esta distância é independente da resolução da imagem) em qualquer direcção, são seleccionadas para possível eliminação.

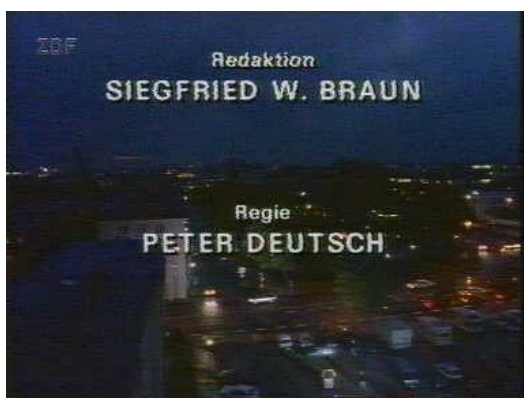
Todas as fronteiras que verifiquem simultaneamente os dois critérios anteriores correspondem a zonas da imagem pouco contrastadas ou a pequenas regiões. Atendendo a que, zonas com este tipo de características, não deverão corresponder a caracteres, estas fronteiras são classificadas como não pertencentes a caracteres e, consequentemente, eliminadas.

3ª Etapa – Filtragem de mediana

Esta etapa consiste na eliminação de parte do ruído existente na imagem. A necessidade da eliminação do ruído na imagem deve-se ao facto deste provocar a sobresegmentação da mesma durante o processo de segmentação. Esta sobresegmentação provoca muitas vezes a divisão das regiões conexas contribuindo, desta forma, para a diminuição do desempenho global do algoritmo de extracção de texto. Com o objectivo de evitar a sobresegmentação são, frequentemente, utilizados dois filtros de tendência central: o filtro de média e o filtro de mediana. Devido à sua própria definição matemática, a utilização da média como filtro para eliminação de ruído tem como desvantagem o ‘apagamento’ das fronteiras. Tal acontecimento deve-se ao facto de no cálculo da média se levarem em conta todos os valores disponíveis, podendo obter-se médias iguais tanto em zonas onde a transição se faz de forma suave, como em zonas onde a transição se faz de forma mais brusca. Por outro lado, a utilização do filtro de mediana, porque matematicamente a mediana é definida como sendo o valor central de um conjunto de valores ordenados, isola as amostras associadas aos níveis mais baixos ou mais elevados dando origem a valores mais representativos da área em questão. Assim, a utilização deste tipo de filtro permite a exploração deste fenómeno com o objectivo de eliminar o ruído existente na imagem. Para este efeito, é aplicada uma mediana com janela de 3×3 *pixels* sobre as zonas da imagem original que não coincidam com as fronteiras resultantes das etapas anteriores. O valor dos *pixels* que coincidem com as fronteiras mantém-se inalterado de forma a preservar as mesmas.

Este processo de detecção de fronteiras, simplificação e filtragem de mediana é percorrido, iterativamente, para a imagem em questão tantas vezes quantas as seleccionadas.

Testes exaustivos foram efectuados, usando vários tipos de imagens, tendo-se obtido bons resultados utilizando três iterações do filtro. A Figura 3.5 ilustra o resultado das várias fases da filtragem de uma imagem, utilizando o filtro iterativo aqui proposto. No exemplo apresentado foram efectuadas três iterações. A Figura 3.5 (b) ilustra o resultado da *detecção de fronteiras* na primeira iteração aplicada sobre a imagem original na Figura 3.5 (a); na Figura 3.5 (c), ilustra-se o resultado da *classificação de fronteiras* no final da terceira iteração, i.e. as fronteiras classificadas como candidatas a caracteres; na Figura 3.5 (d), ilustra-se o resultado da *filtragem de mediana* da imagem original, fora das zonas de fronteira classificadas como candidatas a caracteres.



(a)



(b)

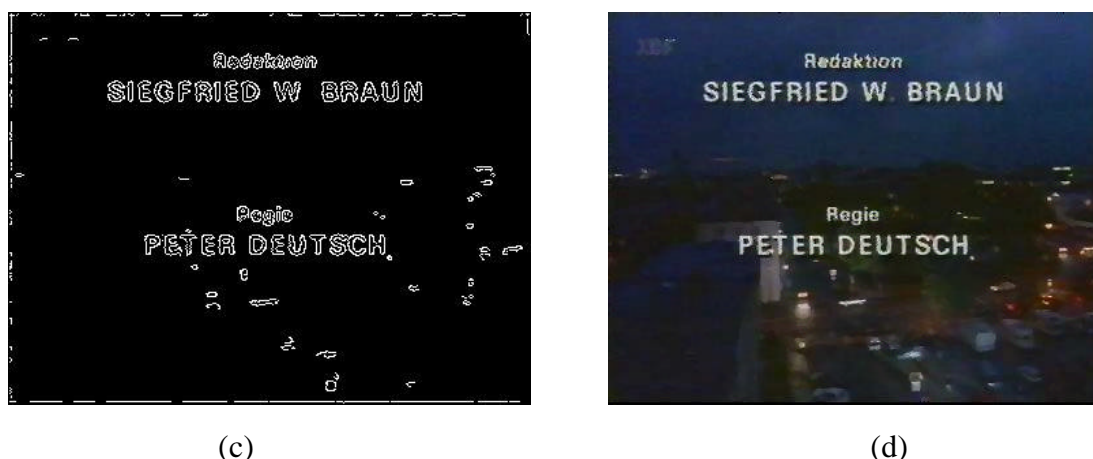


Figura 3.5 – Exemplo das várias fases da filtragem de uma imagem com o filtro iterativo proposto (três iterações): (a) imagem original; (b) detecção de fronteiras; (c) fronteiras classificadas como candidatas a caracteres; (d) filtragem mediana da imagem original fora das zonas de fronteira classificadas como candidatas a caracteres.

A utilização de outro tipo de filtros de simplificação não conduziu a resultados suficientemente satisfatórios para a extracção de texto, pelo que se tornou imperioso a procura de uma nova e mais adequada solução para a simplificação da imagem. A aplicação, de forma iterativa, do filtro proposto tornou possível melhorar o seu desempenho e, assim, atingir de forma mais eficaz os objectivos da sua utilização, ou seja, diminuir o ruído existente na imagem e preservar as suas fronteiras. Todavia, nos testes efectuados, verificou-se que para um número de iterações superior a cinco, o desempenho do filtro não melhora, havendo mesmo o risco das fronteiras das regiões de texto tenderem a desaparecer. Esta tendência deve-se ao facto da detecção de fronteiras na iteração n se efectuar sobre o resultado da iteração $n-1$. Deste modo, a acção sucessiva do filtro de mediana sobre a imagem torna a detecção de fronteiras mais difícil à medida que o número de iterações aumenta. Esta diminuição da eficiência na detecção de fronteiras, torna mais difícil proteger as mesmas da acção do filtro de mediana.

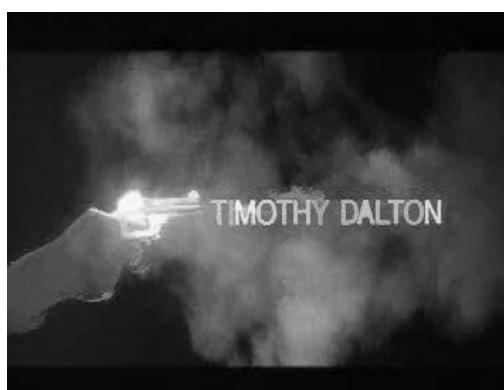
A escolha do filtro iterativo proposto em detrimento de outras soluções ficou também a dever-se ao facto de a técnica utilizada para efectuar a detecção de texto ser muito sensível a alterações, quer nas fronteiras entre o texto e o fundo da imagem, quer na cor do próprio texto. Desta forma, foi necessária a escolha de um filtro que preservasse essas características nas imagens o mais possível; neste sentido, o desempenho do filtro desenvolvido revelou-se mais eficaz quando comparado com outros filtros, como por exemplo os filtros morfológicos. A Figura 3.6 ilustra o desempenho do filtro iterativo face a um filtro morfológico *open-close* com reconstrução usando uma janela de 3×3 pixels [Cortez95].



(a)



(b)



(c)



(d)



(e)



(f)

Figura 3.6 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico *open-close* com reconstrução: (a) e (b) imagens originais; (c) e (d) imagens filtradas com o filtro morfológico *open-close* com reconstrução usando uma janela de 3×3 pixels; (e) e (f) imagens filtradas com o filtro iterativo proposto (três iterações).

Como se pode constatar na Figura 3.6, sempre que a espessura dos caracteres é inferior à janela utilizada (a dimensão mínima para a janela é 3×3 pixels), o filtro morfológico *open-close* com reconstrução altera a luminância dos pixels. Esta alteração tende a reduzir a

eficácia do algoritmo de segmentação, uma vez que este se baseia na diferença de luminância existente entre as várias regiões.

Todavia, existem situações em que a filtragem da imagem, quer através do filtro proposto, quer através do filtro *open-close* com reconstrução, contribui para uma diminuição do desempenho do algoritmo de detecção de texto. Esta diminuição de desempenho ocorre sobretudo quando:

- **Caracteres estão muito próximos uns dos outros** – Nesta situação, a filtragem contribui para a união dos caracteres adjacentes; esta situação pode ser observada na Figura 3.7 onde, quer o filtro proposto – Figura 3.7 (b) –, quer o filtro morfológico – Figura 3.7 (c) – provocam a união de caracteres, contribuindo assim para uma degradação do desempenho final do algoritmo de detecção de texto;

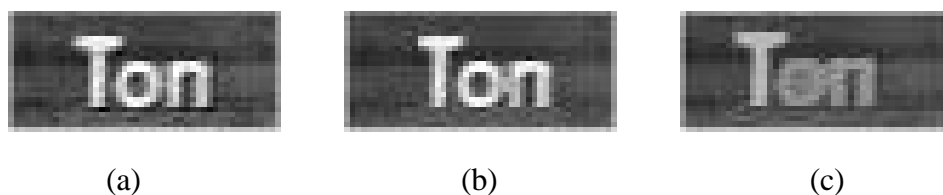


Figura 3.7 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico *open-close* com reconstrução a uma imagem onde os caracteres se encontram muito próximos uns dos outros: (a) imagem original; (b) imagem filtrada com o filtro iterativo proposto (três iterações); (c) imagem filtrada com o filtro morfológico *open-close* com reconstrução utilizando uma janela de 3×3 pixels.

- **Fronteiras são pouco contrastadas** – Nesta situação, a utilização do filtro iterativo degrada os limites dos caracteres uma vez que este se baseia na detecção de fronteiras e o baixo contraste dificulta a detecção das mesmas. Por conseguinte, a eficácia do algoritmo de detecção de texto também diminui pois a degradação dos limites dos caracteres pode originar a supressão dos mesmos; esta situação pode ser observada na Figura 3.8 onde, quer o filtro proposto – Figura 3.8 (b) –, quer o filtro morfológico – Figura 3.8 (c) – provocam a degradação das fronteiras dos caracteres, contribuindo deste modo para uma segmentação deficiente com a consequente degradação do desempenho final do algoritmo de detecção de texto.

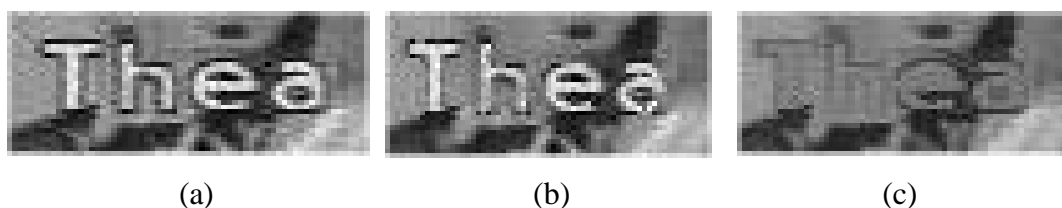


Figura 3.8 – Exemplos da aplicação do filtro iterativo proposto e de um filtro morfológico *open-close* com reconstrução a uma imagem onde as fronteiras entre os caracteres e o fundo da imagem são pouco contrastadas: (a) imagem original; (b) imagem filtrada com o filtro iterativo proposto (três iterações); (c) imagem filtrada com o filtro morfológico *open-close* com reconstrução utilizando uma janela de 3×3 pixels.

3.2.2 Segmentação

A segmentação da imagem tem por objectivo a divisão da mesma em regiões homogéneas, segundo um ou mais critérios. A segmentação efectuada no contexto do algoritmo proposto para extracção de texto consiste na divisão da imagem em regiões conexas usando como critério de homogeneidade a luminância; este critério de homogeneidade é também usado em diversos algoritmos de segmentação propostos na literatura [Lienhart95, Zhong95, Jain98, Messelodi99, Lienhart00] no contexto de soluções para extracção de texto em imagens. Nesta Tese, adoptou-se como algoritmo de segmentação o algoritmo de *split-and-merge* desenvolvido por Cortez et al. [Cortez95] e que se baseia numa decomposição hierárquica da imagem. A arquitectura do algoritmo de *split-and-merge* adoptado está ilustrada no esquema da Figura 3.9.

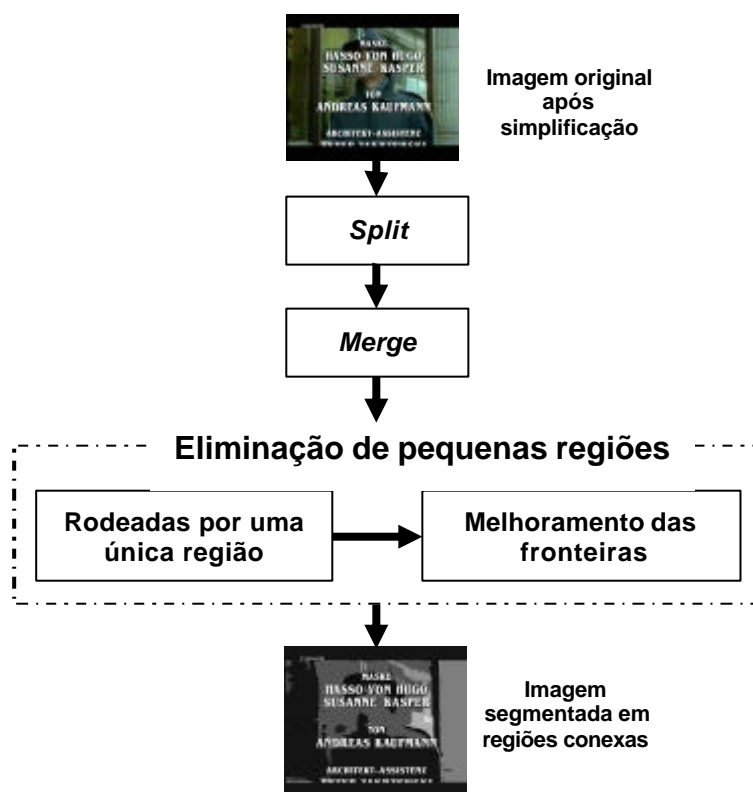


Figura 3.9 – Arquitectura do algoritmo de segmentação das imagens adoptado.

Como se pode observar na Figura 3.9, a segmentação da imagem utilizando o método *split-and-merge* decorre em três fases principais distintas, nomeadamente *split*, *merge* e eliminação de pequenas regiões, que se descrevem de seguida.

3.2.2.1 Split

Esta fase do algoritmo de segmentação tem como objectivo a formação de um conjunto inicial de regiões com vista a simplificar e reduzir o esforço computacional associado à fase de *merge*. Assim, segundo Cortez et al. [Cortez95], a fase de *split* consiste na divisão sucessiva da imagem em regiões quadradas, de acordo com o seguinte processo:

- **Divisão da imagem** – O processo de divisão da imagem consiste na divisão da imagem em sub-regiões que sejam homogéneas em termos da sua luminância. Para efectuar a representação das regiões originadas pelo processo de divisão, utiliza-se uma estrutura de dados baseada numa árvore quaternária. A divisão da imagem inicia-se com uma única região, do tamanho de toda a imagem, que é dividida em quatro sub-regiões de igual dimensão. Caso alguma destas regiões não seja homogénea, é novamente dividida (em quatro sub-regiões de igual dimensão), repetindo-se este processo até que se verifique o critério de paragem para todas as regiões;
- **Critério de homogeneidade** – Adoptou-se como critério de homogeneidade a Gama Dinâmica (GD) uma vez que este critério, nos testes efectuados em imagens com transições bruscas no valor da luminância, apresentou melhores resultados do que a variância. Assim, uma dada região R deve ser dividida em quatro sub-regiões iguais quando $GD = \max\{R\} - \min\{R\} = Th_{split}$ onde $\max\{R\}$ e $\min\{R\}$ são, respectivamente, o valor máximo e o valor mínimo da luminância na região R em questão e Th_{split} é o limiar de homogeneidade previamente fixado. A escolha do valor para Th_{split} resulta de um compromisso entre a velocidade e a precisão da segmentação, i.e. quanto maior for Th_{split} mais rápida é a fase de *merge*. Todavia, valores de Th_{split} muito elevados podem provocar a fusão entre as várias regiões, sobretudo em imagens pouco contrastadas. Nesta Tese, usa-se $Th_{split} = 30$, valor que garante a precisão necessária para a extracção de texto; este valor foi obtido empiricamente através de testes exaustivos;
- **Crítérios de paragem** – Com vista a controlar a divisão de cada imagem em regiões de tamanho sucessivamente menor, foram definidos dois critérios de paragem para o processo de divisão das regiões:
 - ♦ A dimensão da região é igual a um *pixel*;
 - ♦ A região cumpre o critério de homogeneidade ou seja $GD = \max\{R\} - \min\{R\} < Th_{split}$.

A fase de *split* termina quando não for possível gerar mais sub-regiões, i.e. quando um dos critérios de paragem for atingido para todas as regiões. Cada região é identificada com um identificador único (*ID*) e aos *pixels* que a formam é atribuído o valor médio da luminância calculado sobre todos os *pixels* pertencentes à mesma. A Figura 3.10 ilustra o processo de *split* para uma imagem com 4×4 *pixels* e um valor de $Th_{split} = 3$ [Montoya00].

6	7	1	3
8	6	5	4
8	8	5	6
7	8	6	6

(a)

6	7	1	3
8	6	5	4
8	8	5	6
7	8	6	6

(c)

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

(b)

0	2	3
	6	7
8	10	

(d)

Figura 3.10 – Exemplo da fase de *split*: (a) valor da luminância para os *pixels* da imagem; (b) *ID* dos *pixels* antes do início da fase de *split* (cada *pixel* uma região); (c) regiões formadas depois da fase de *split* com $Th_{split} = 3$ e (d) *ID* das regiões no final da fase de *split*.

Na Figura 3.11 ilustra-se a aplicação da fase de *split* a uma imagem real, usando vários valores para o limiar de homogeneidade. Na Figura 3.11 (b) é ilustrado o resultado da fase de *split* com $Th_{split}=30$, sendo a imagem dividida em 26337 regiões homogêneas em termos de luminância; na Figura 3.11 (c) e (d), foram utilizados valores para Th_{split} iguais a 45 e 60, originando a divisão da imagem em 19020 e 15546 regiões, respectivamente.



Figura 3.11 – Exemplo da aplicação da fase de *split*: (a) imagem original; (b), (c) e (d) imagens divididas em 26337, 19020 e 15546 regiões depois da fase de *split* com Th_{split} igual a 30, 45 e 60, respectivamente.

3.2.2.2 Merge

Esta fase tem como objectivo efectuar a fusão (*merge*) das regiões adjacentes resultantes da fase de *split* que sejam suficientemente parecidas segundo o critério de homogeneidade adoptado. A ordem pela qual é feito o *merge* das regiões é uma característica importante deste processo uma vez que vai influenciar fortemente o resultado final, ou seja, a imagem segmentada. O *merge* das regiões é feito de uma forma iterativa através do *merge* sucessivo de pares de regiões adjacentes que unidas originem uma região ainda homogênea. Esta fase termina quando o critério de homogeneidade não se verificar para todos os pares de regiões adjacentes, ou seja, quando não for possível fundir quaisquer regiões adjacentes numa única região suficientemente homogênea. Aos *pixels* que formam a nova região, recém-criada, é atribuído o valor médio da luminância obtido a partir das duas regiões que lhe deram origem.

Como critério de homogeneidade para o *merge* utilizou-se também a Gama Dinâmica (*GD*), ou seja, duas regiões adjacentes R_a e R_b são agrupadas se $\max \{R_a \cup R_b\} - \min \{R_a \cup R_b\} = Th_{merge}$ onde Th_{merge} é o limiar de homogeneidade para *merging* previamente fixado. Na presente Tese, utiliza-se $Th_{merge} = 35$, valor que foi obtido empiricamente através de testes exaustivos. A utilização de valores de Th_{merge} superiores a 35, no caso de imagens onde o contraste entre o texto e fundo é baixo, origina a fusão deste com as suas regiões vizinhas; valores de Th_{merge} baixos, provocam a segmentação dos caracteres em várias regiões.

A Figura 3.12 ilustra os resultados para a mesma imagem usada anteriormente, depois de aplicadas as fases de *split* e de *merge*. Na fase de *split* foi utilizado $Th_{split}=30$. Na Figura 3.12 (b) é ilustrado o resultado no final da fase de *merge* com $Th_{merge}=35$ na qual a imagem ficou segmentada em 1773 regiões homogêneas em termos de luminância; nas Figura 3.12 (c) e (d), foram utilizados valores para Th_{merge} iguais a 50 e 65, os quais originaram imagens segmentadas com 1004 e 566 regiões, respectivamente.



(a)



(b)



(c)



(d)

Figura 3.12 – Exemplo da aplicação da fase de *merge* a uma imagem dividida com $Th_{split}=30$: (a) imagem original; (b), (c) e (d) imagens segmentadas depois da fase de *merge* com Th_{merge} igual a 35, 50 e 65, respectivamente.

3.2.2.3 Eliminação de Pequenas Regiões

Esta fase do processamento tem como objectivo a remoção, da imagem segmentada, de pequenas regiões provenientes essencialmente do ruído. A eliminação de pequenas regiões é feita em duas fases distintas:

- Eliminação de pequenas regiões em cuja vizinhança existe apenas uma região, ou seja, pequenas regiões completamente rodeadas por uma única região;
- Melhoramento das fronteiras através da eliminação de pequenas regiões que, por se encontrarem sobre a fronteira entre duas regiões maiores, contribuem para a degradação desta.

Estas duas fases da eliminação de pequenas regiões serão discutidas em pormenor nas secções seguintes.

1º Fase – Eliminação de pequenas regiões rodeadas por uma única região

Nesta fase, apenas as pequenas regiões que se encontrem envolvidas por uma única região são suprimidas usando, como critério de eliminação, o tamanho da região em termos de número de *pixels*. Assim, todas as regiões conexas (definidas em vizinhança 8) cujo tamanho em termos de número de *pixels* seja inferior a 0,004% do tamanho da imagem original (4 *pixels* numa imagem CIF) são englobadas na região que as rodeia. Este valor foi definido com base no tamanho mínimo do texto que é possível detectar com o algoritmo proposto. O valor que se atribui aos *pixels* da nova região é igual ao já usado anteriormente para a região envolvente, partindo do princípio que a região eliminada corresponde, essencialmente, a ruído. Um exemplo do efeito da eliminação de pequenas regiões rodeadas por uma única região numa imagem segmentada pode ser observado na Figura 3.13; no exemplo em questão foram eliminadas 61 pequenas regiões.



Figura 3.13 – Exemplo da aplicação da eliminação de pequenas regiões rodeadas por uma única região: (a) imagem segmentada depois da fase de *merge*; (b) imagem segmentada depois de eliminadas as pequenas regiões.

2ª Fase – Melhoramento das fronteiras através da eliminação de pequenas regiões

Nesta fase, pretende-se efectuar o melhoramento das fronteiras dos caracteres através da eliminação de pequenas regiões que se encontrem sobre as fronteiras dos mesmos, i.e. pequenas regiões que formem fronteira com mais do que uma região e que, de alguma forma, contribuem para a degradação dos limites dos caracteres. A necessidade de eliminar pequenas regiões surge devido à sobresegmentação originada pelo ruído existente nas imagens. Assim, o objectivo a atingir nesta fase é a detecção dessas pequenas regiões com vista a efectuar a sua fusão com regiões vizinhas de forma a melhorar as fronteiras dos caracteres.

A dificuldade na eliminação deste tipo de regiões prende-se com a dificuldade na determinação da região com a qual se deve efectuar o *merge* da pequena região em questão. Para identificar as regiões com as quais deve ser feito o *merge* das pequenas regiões, de forma a melhorar as fronteiras dos caracteres, vai ser utilizada uma técnica já utilizada por Lienhart em [Lienhart00] a qual combina a detecção de fronteiras com a sua filtragem baseada na orientação da variação local do contraste. Contudo em [Lienhart00] apenas é referido que foi utilizada esta combinação de técnicas, não apresentando o autor qualquer descrição sobre a sua implementação. A abordagem aqui proposta usa a mesma combinação de técnicas tendo-se desenvolvido a metodologia precisa para a sua aplicação. Esta metodologia verifica se a transição entre duas regiões detectadas coincide, ou não, com uma fronteira detectada e com uma variação local do contraste. Caso não se verifique esta coincidência, a separação entre essas duas regiões (uma das quais é pequena) é considerada como originada pelo ruído e, conseqüentemente, é feito o *merging* dessas regiões. Na Figura 3.14, ilustra-se o efeito da sobresegmentação originada pelo ruído existente nas imagens, bem como o tipo de pequenas regiões que se pretendem eliminar; na Figura 3.14 (c) ilustra-se o resultado ideal do melhoramento das fronteiras através da eliminação das pequenas regiões e na Figura 3.14 (d) apresenta-se o resultado utilizando o algoritmo proposto.

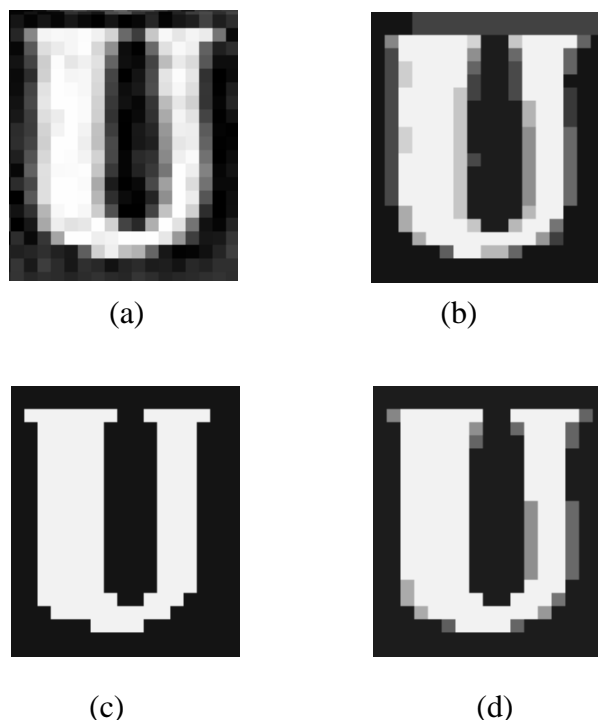


Figura 3.14 – Exemplo do efeito da sobresegmentação originada pelo ruído existente nas imagens: (a) imagem original; (b) imagem segmentada depois das fases de *split* e *merge*; (c)

resultado ideal da eliminação das pequenas regiões e (d) resultado utilizando o algoritmo proposto.

A Figura 3.15 ilustra a arquitectura do método de melhoramento de fronteiras utilizado para a identificação das pequenas regiões nas condições acima descritas, o qual considera quatro etapas distintas: detecção de fronteiras; cálculo da variação local do contraste e da variância; validação das fronteiras e, por último, *merge* das regiões relevantes.

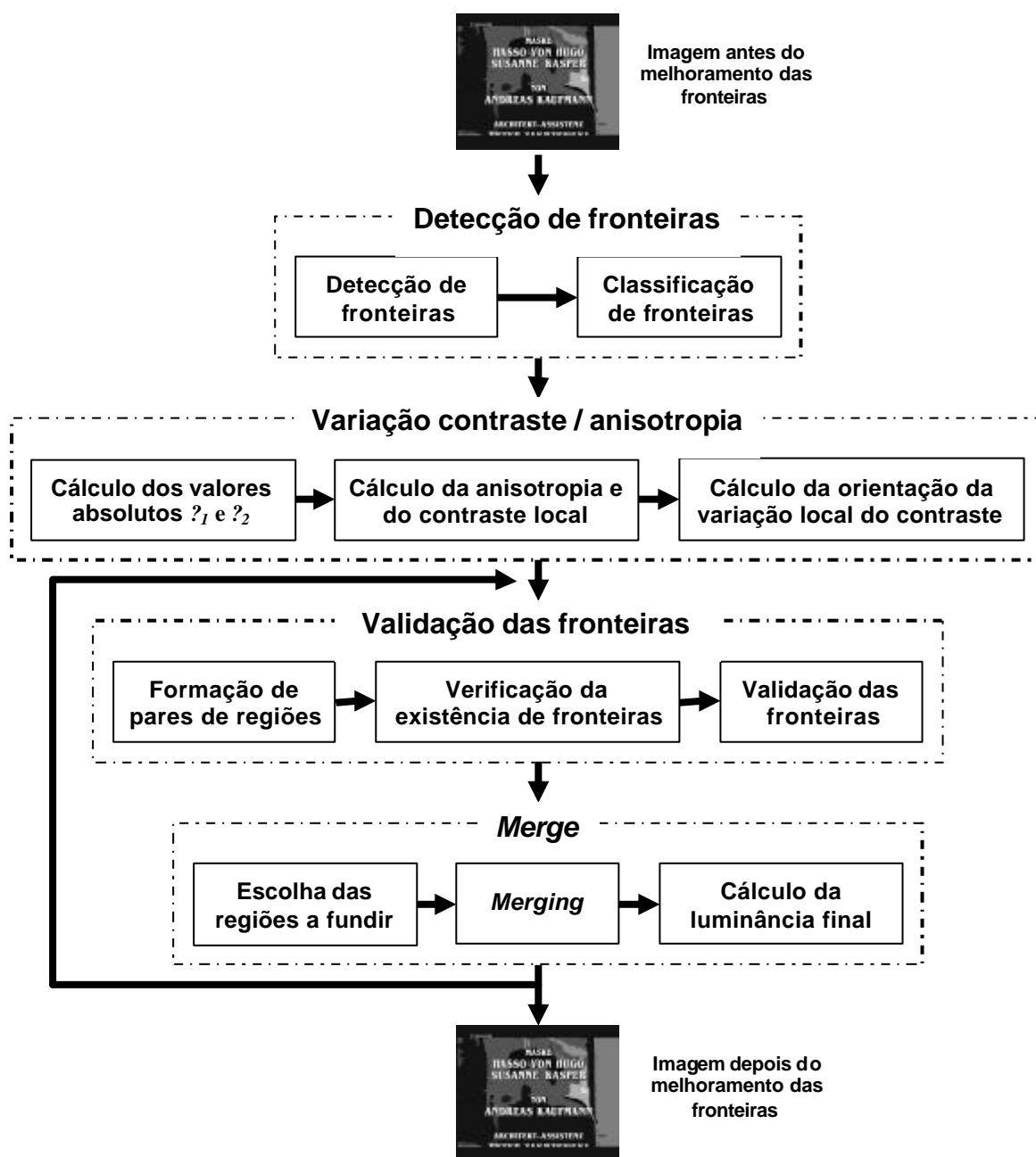


Figura 3.15 – Arquitectura do processo de melhoramento de fronteiras através da eliminação de pequenas regiões.

Estas quatro etapas do melhoramento de fronteiras serão discutidas em pormenor nas secções seguintes:

1ª Etapa – Detecção de fronteiras

Esta etapa consiste na detecção e classificação das fronteiras existentes na imagem original, através dos passos seguintes:

- 1º **Detecção de fronteiras** – Para efectuar a detecção das fronteiras existentes na imagem, utiliza-se um detector de fronteiras baseado no método de Canny [Canny86] em tudo idêntico ao usado na Secção 3.2.1;
- 2º **Classificação das fronteiras** – Para efectuar a classificação das fronteiras em fronteiras pertencentes a caracteres e fronteiras que não pertencem a caracteres, utiliza-se a técnica descrita na Secção 3.2.1.

2ª Etapa – Cálculo da orientação local da variação do contraste e da anisotropia

Esta etapa consiste no cálculo da orientação local da variação do contraste e da anisotropia e tem como objectivo a validação das fronteiras detectadas na fase anterior, i.e. verificar se estas correspondem, efectivamente, a fronteiras entre regiões onde existe uma variação local do contraste ou se correspondem a fronteiras com uma estrutura isotrópica, ou seja, estruturas cujas propriedades variam de igual modo em todas as direcções e logo originadas pelo ruído. Para tal, foi utilizada uma técnica baseada em tensores de inércia proposta por Jähne [Jähne97]. A vantagem da utilização de um tensor de inércia para representar a variação local do contraste prende-se com a sua capacidade para distinguir estruturas onde não existe variação do gradiente na vizinhança do *pixel* de estruturas isotrópicas (p.e. ruído incorrelacionado) onde os valores da variação do gradiente na vizinhança do *pixel* são iguais em todas as direcções. No caso de uma representação vectorial, ambas as estruturas resultariam num vector da variação do gradiente com magnitude igual a zero.

O tensor de inércia proposto em [Jähne97] representa a variação local do contraste usando um vector com valores absolutos τ_1 e τ_2 . De acordo com Jahne [Jähne97], a partir da análise dos valores absolutos da variação local do contraste pode determinar-se a existência, ou não, de variação local do contraste. Pode ainda verificar-se a existência, ou não, de uma estrutura isotrópica, i.e. uma estrutura cujas propriedades variam de igual modo em todas as direcções. Para concluir sobre o tipo de variação em questão, usam-se as seguintes regras:

- $\tau_1 = \tau_2 = 0$? Os valores absolutos são iguais a zero, o que implica que a magnitude do gradiente seja igual a zero; esta condição indica que não existe variação do contraste na vizinhança local do *pixel*;
- $\tau_1 > 0, \tau_2 = 0$? Apenas um valor absoluto é zero; esta condição indica que só existe variação do contraste numa direcção;
- $\tau_1 > 0, \tau_2 > 0$? Os valores absolutos são diferentes de zero; esta condição indica que o contraste varia em todas as direcções. No caso particular de $\tau_1 = \tau_2$, está-se perante uma estrutura isotrópica.

O algoritmo de cálculo da orientação local da variação do contraste e da anisotropia inclui vários passos:

1º Cálculo dos valores absolutos τ_1 e τ_2 – O cálculo dos valores absolutos τ_1 e τ_2 [Jähne97] é feito através da seguinte forma:

- Inicialmente, são calculadas as estruturas do tensor de inércia J_{xx} , J_{yy} e J_{xy} . Essas estruturas são calculadas através da aplicação da expressão (3.4):

$$J_{pq} = B(D_p.D_q) \quad (3.4)$$

Em que D_p e D_q representam as primeiras derivadas calculadas segundo as direcções p e q na vizinhança de cada *pixel*, respectivamente. Os valores de D_p e D_q são obtidos através da expressão (3.1), com o operador Gaussiano, G , aproximado por uma expansão de Taylor {45, -9, 1} e B representa o operador (isotrópico) Gaussiano, aproximado por $C+1$ coeficientes de uma máscara binomial de ordem 6 com $2C+1$ coeficientes:

$$B^6 = \frac{1}{64} [1 \ 6 \ 15 \ 20 \ 15 \ 6 \ 1] \quad (3.5)$$

- De seguida, são calculados os valores absolutos τ_1 e τ_2 , para cada *pixel*, através das seguintes expressões (3.6) e (3.7):

$$I_1 = \frac{J_{xx} + J_{yy}}{2} + \sqrt{\left(\frac{J_{xx} + J_{yy}}{2}\right)^2 - (J_{xx} - J_{yy}) - J_{xy}^3} \quad (3.6)$$

$$I_2 = \frac{J_{xx} + J_{yy}}{2} - \sqrt{\left(\frac{J_{xx} + J_{yy}}{2}\right)^2 - (J_{xx} - J_{yy}) - J_{xy}^3} \quad (3.7)$$

2º Cálculo da anisotropia e do contraste local – Segundo Belongie [Belongie97], o cálculo da anisotropia e do contraste local pode ser feito a partir dos valores absolutos τ_1 , τ_2 da seguinte forma:

- Anisotropia = $1 - \tau_2/\tau_1$;
- Contraste local = $\tau_1 + \tau_2$.

3º Cálculo do ângulo de orientação – O ângulo de orientação da variação local do contraste para cada *pixel* é calculado através da seguinte expressão (3.8):

$$\tan 2q = \frac{2J_{xy}}{J_{yy} - J_{xx}} \quad (3.8)$$

Na Figura 3.16 é apresentado um exemplo do cálculo do contraste local. A Figura 3.16 (b) representa as regiões da imagem onde o contraste local e a anisotropia são superiores a 0.001 e 0.5, respectivamente. O valor da anisotropia varia entre 0 (estruturas isotrópicas) e 1 (estruturas com variação do contraste numa única direcção).



Figura 3.16 – Exemplo da aplicação do tensor de inércia: (a) imagem original; (b) imagem com as regiões onde o contraste local é superior a 0.001 e a anisotropia é superior a 0.5.

3ª Etapa – Validação das fronteiras

Esta etapa destina-se a validar as fronteiras existentes entre as regiões determinadas através do processo de segmentação descrito na secção 3.2.2.2, i.e. verificar se estas fronteiras correspondem, efectivamente, a fronteiras entre regiões onde existe uma variação local do contraste ou se, pelo contrário, fazem a separação de regiões isotrópicas originadas por ruído e como tal devem ser eliminadas. O algoritmo de validação de fronteiras apresenta os seguintes passos:

- 1º **Formação de pares de regiões** – Todas as pequenas regiões com um tamanho (em termos do número de *pixels*) inferior a 0,01% do tamanho da imagem original (10 *pixels* numa imagem CIF) formam pares de regiões. Esses pares são constituídos por uma pequena região e uma das suas regiões vizinhas, pequena ou não;
- 2º **Verificação da existência de fronteiras** – Nesta fase, procede-se à verificação da existência, ou não, de fronteiras entre os pares de regiões formados na fase anterior. Para tal, deve verificar-se se a zona de separação entre as duas regiões que formam o par coincide com a localização de uma das fronteiras detectadas anteriormente através do filtro de Canny;
- 3º **Validação das fronteiras** – Para todos os pares de regiões cuja separação coincida com uma fronteira de Canny (determinados na fase anterior), verifica-se se essa fronteira também coincide com uma zona da imagem com variação local do contraste. Deste modo, torna-se possível verificar se essa fronteira pertence a uma região isotrópica originada pelo ruído, ou não. Para verificar a validade das fronteiras existentes entre regiões adjacentes, são analisados os pares de *pixels* que fazem parte da fronteira entre elas da seguinte forma: considerem-se dois *pixels* P_1 e P_2 que façam parte da fronteira entre duas regiões R_1 e R_2 , respectivamente; essa fronteira é considerada válida se mais do que dois pares de *pixels* pertencentes à fronteira, possuírem uma variação de contraste entre os seus *pixels* superior a um valor predefinido. Esta variação de contraste tem que ser perpendicular à direcção da fronteira, i.e. paralela a uma recta que passe pelos dois *pixels*. Para que tal se verifique, os pares de *pixels* têm que verificar as seguintes condições:

- A orientação da variação local do contraste deve ser paralela a uma recta que passe pelos dois *pixels*;
- O valor da anisotropia do par de *pixels* deve ser superior a um valor de limiar Th_{anis} previamente definido; na presente Tese foi utilizado $Th_{anis}=0.5$, valor que foi obtido empiricamente através de testes exaustivos;
- O valor local do contraste do par de *pixels* deve ser superior a um valor de limiar Th_{cont} , previamente definido; na presente Tese foi utilizado $Th_{cont}=0.001$, valor que foi obtido empiricamente através de testes exaustivos.

4ª Etapa – *Merging* das regiões

Esta etapa consiste em efectuar a fusão dos pares de regiões que não sejam separados por uma fronteira validada na fase anterior, através dos seguintes passos:

- 1º **Escolha das regiões a fundir** – De entre todos os pares de regiões formados pela pequena região em análise e as suas vizinhas e que não possuem uma fronteira válida a separá-los, escolhe-se para a fusão o par de regiões mais homogéneo, ou seja, aquele que dá origem a uma região fundida mais homogénea;
- 2º ***Merging*** – Efectua-se o *merge* do par de regiões seleccionado no passo anterior, desde que este cumpra o critério de homogeneidade. O critério de homogeneidade aqui utilizado é idêntico ao indicado na Secção 3.2.2.2, usando-se um valor superior para o limiar de homogeneidade; nesta Tese, usa-se $Th_{fusão}=55$, valor que foi obtido empiricamente através de testes exaustivos. A utilização de valores de $Th_{fusão}$ superiores a 55, no caso de ocorrer uma falha na detecção das fronteiras dos caracteres, origina a fusão destes com as suas regiões vizinhas;
- 3º **Cálculo da luminância final** – Aos *pixels* que formam a nova região recém-criada é atribuído o valor médio da luminância calculado sobre as duas regiões que lhe deram origem.

As etapas 3 e 4 do processo de melhoramento de fronteiras através da eliminação de pequenas regiões (validação das fronteiras e *merging* das regiões) são aplicadas de uma forma iterativa através do *merge* sucessivo do par de regiões mais homogéneo, ou seja, aquele que dá origem a uma região fundida mais homogénea. Este processo termina quando não for possível fundir qualquer par de regiões numa única região suficientemente homogénea.

A vantagem da eliminação de pequenas regiões através do presente método prende-se com a capacidade deste em utilizar as pequenas regiões, que se encontram sobre as fronteiras dos caracteres, para melhorar a qualidade dessas fronteiras. Deste modo conseguem-se eliminar lacunas existentes nas fronteiras dos caracteres, provocadas, essencialmente, por ruído existente nas imagens. Um exemplo do efeito da eliminação de pequenas regiões numa imagem com muito ruído pode ser observado na Figura 3.17. A Figura 3.17 (a) ilustra a imagem original; na Figura 3.17 (b), mostra-se a imagem original segmentada em 981 regiões conexas, depois das fases de *split* e de *merge*; na Figura 3.17 (c), é ilustrado o efeito da primeira fase da eliminação de pequenas regiões, ou seja, a eliminação de pequenas regiões completamente rodeadas por uma única região e na qual foram eliminadas 116 pequenas regiões; na Figura 3.17 (d), ilustra-se o efeito da segunda fase da eliminação de pequenas regiões e que permitiu a eliminação de 247 pequenas regiões, i.e. a eliminação de 247

pequenas regiões que por se encontrarem sobre a fronteira entre duas regiões maiores tornam possível o melhoramento da mesma.



(a)



(b)



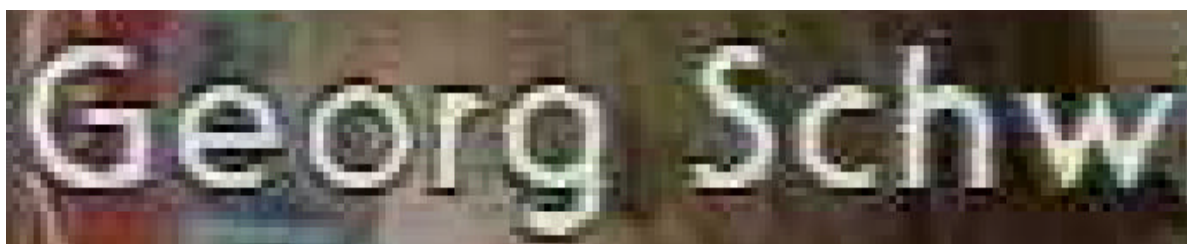
(c)



(d)

Figura 3.17 – Exemplo da aplicação da eliminação de pequenas regiões: (a) imagem original; (b) imagem segmentada antes de eliminadas as pequenas regiões; (c) imagem depois de eliminadas as pequenas regiões completamente rodeadas por uma única região e (d) imagem depois de eliminadas as pequenas regiões em cuja vizinhança existe mais de uma região.

Com a Figura 3.18 pretende-se ilustrar, de forma mais pormenorizada, o efeito da eliminação de pequenas regiões sobre as fronteiras dos caracteres através do método proposto. Desta forma, as Figura 3.18 (a), (b) e (c) correspondem a pequenas áreas retiradas das Figura 3.17 (a), (c) e (d), respectivamente.



(a)

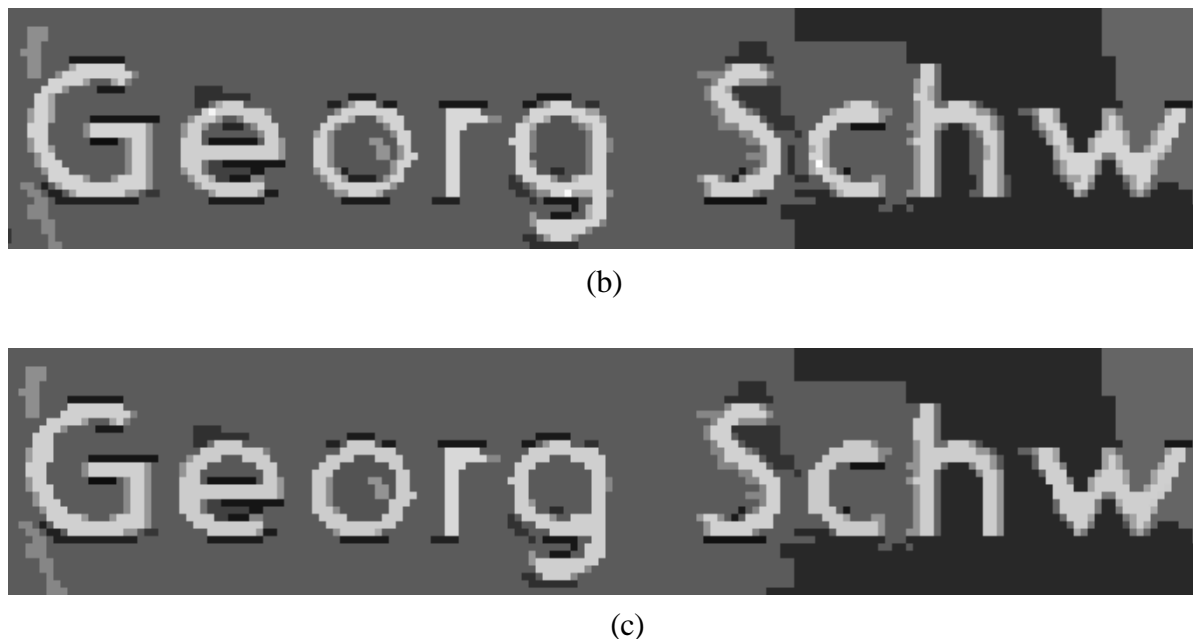


Figura 3.18 – Exemplo do melhoramento das fronteiras dos caracteres devido à eliminação de pequenas regiões: (a) imagem original; (b) imagem segmentada antes da eliminação de pequenas regiões; (c) imagem depois de efectuada a eliminação de pequenas regiões.

Ainda que os melhoramentos obtidos pudessem ser ainda mais significativos não há dúvida que o processamento proposto permite melhorar a ‘qualidade’ das fronteiras dos caracteres. Estes pequenos melhoramentos assumem importância fundamental pois podem evitar a divisão de um carácter em duas ou mais regiões, o que levaria à sua eliminação na fase de detecção de caracteres. Na Figura 3.18 é ilustrado este efeito com o carácter ‘c’ da palavra ‘Schw’.

3.2.3 Detecção de Caracteres

A detecção de caracteres visa a classificação de cada uma das regiões, provenientes da fase de segmentação, como carácter de texto, ou não. Para tal, deve efectuar-se a classificação de cada uma das regiões provenientes da fase de segmentação, determinando, assim, quais as regiões que correspondem a caracteres de texto. Para este efeito, usam-se de forma combinada duas técnicas: a análise de contraste e a análise geométrica das regiões [Fletcher88, Ohya94, Wu99, Lienhart95, Zhong95, Messelodi99, Lienhart00]. Todas as regiões que não forem classificadas como caracteres de texto são descartadas.

3.2.3.1 Análise de Contraste

A análise de contraste tem como objectivo a classificação, ou não, das regiões conexas provenientes da fase de segmentação como prováveis regiões de texto com base no contraste. Para tal, explora-se o elevado contraste tipicamente existente entre os caracteres e o fundo da imagem, principalmente para texto gráfico [Ohya94, Lienhart95, Messelodi99, Lienhart00].

Note-se no entanto, que este tipo de abordagem torna-se menos eficaz quando o contraste existente entre o texto e fundo da imagem é baixo.

No algoritmo de análise de contraste proposto nesta Tese, e que tem como objectivo melhorar a eficácia da análise de contraste para imagens onde o texto é pouco contrastado em relação ao fundo, combina-se a detecção de fronteiras com o método baseado no contraste local para cada posição da imagem proposto por Lienhart [Lienhart00]. A Figura 3.19 ilustra a arquitectura do esquema de análise de contraste proposto.

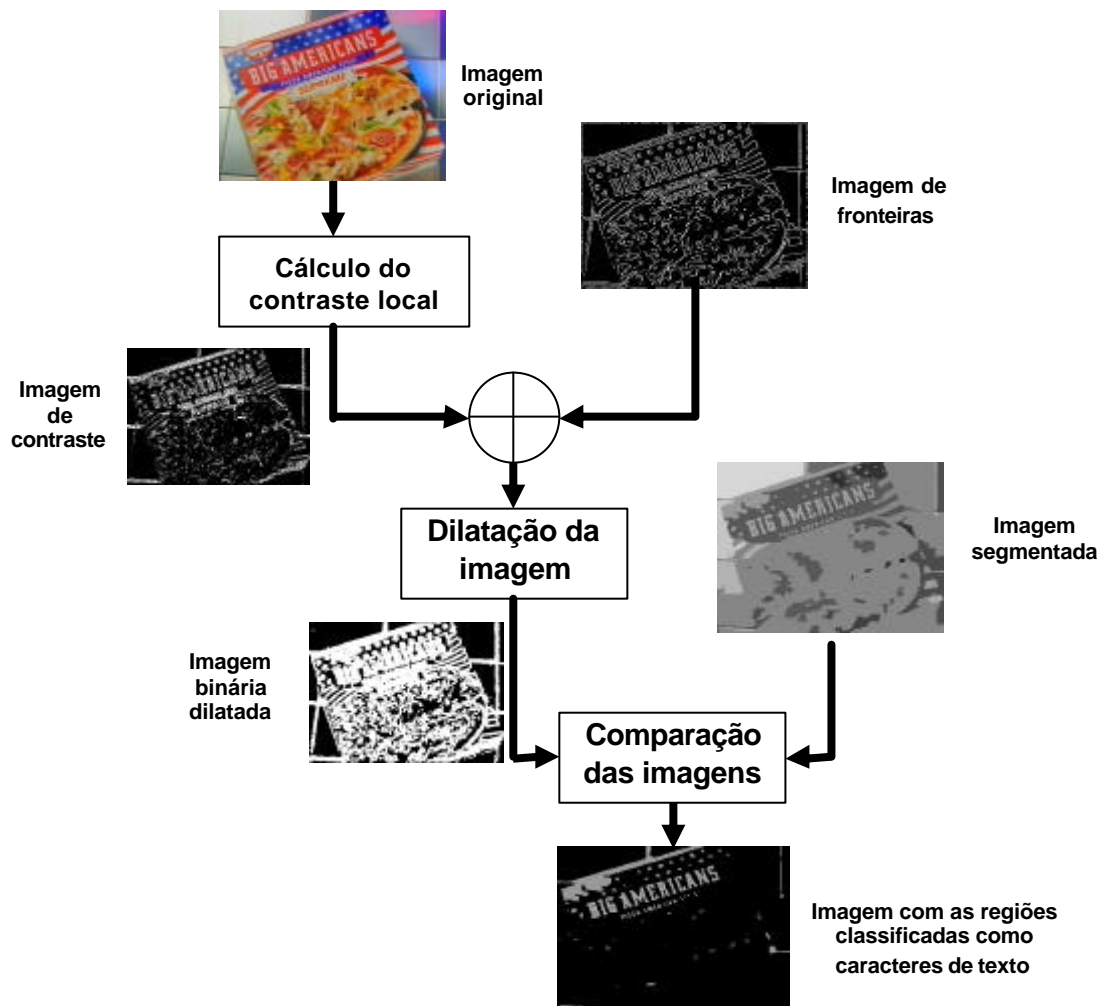


Figura 3.19 – Arquitectura do processo proposto para a análise de contraste.

Na arquitectura apresentada na Figura 3.19, pode observar-se que a análise de contraste decorre em três fases principais: cálculo do contraste local, dilatação da imagem de contraste e, por último, comparação da imagem de contraste dilatada com a imagem segmentada. Descrevem-se, de seguida, as várias etapas da análise de contraste:

1ª Etapa – Cálculo do contraste local

O cálculo do contraste local destina-se a obter uma imagem binária na qual as regiões de elevado contraste são representadas a branco e as restantes a preto. O cálculo do contraste local é feito através de:

- Primeiro, calcula-se o contraste local para cada posição da imagem. Segundo Lienhart [Lienhart00], o contraste local da posição $I(x,y)$ é dado pela expressão (3.9):

$$Cont_{local}(x, y) = \sum_{k=-r}^r \sum_{l=-r}^r G_{k,l} \cdot |I_{x,y} - I_{x-k,y-l}| \quad (3.9)$$

Onde $G_{k,l}$ é um filtro 2D de Gauss [Lienhart00], r (usa-se $r=3$) o tamanho da vizinhança local usada para o cálculo e $| |$ representa o valor absoluto da diferença entre os valores das luminâncias para o *pixel* em análise, antes e depois de filtrado.

O contraste local é, tipicamente, representado através de uma imagem binária onde as regiões com contraste superior a um determinado limiar, Th_{cont} , aparecem representadas a branco. Nesta Tese, usa-se $Th_{cont}=20$ para texto gráfico, mais contrastado, e $Th_{cont}=10$ para texto de cena, menos contrastado; estes valores foram obtidos na sequência de testes exaustivos com vários imagens.

- De seguida, efectua-se um **ou** da imagem binária resultante do passo anterior com a imagem das fronteiras calculada na Secção 3.2.1. com vista a obter uma nova imagem binária mais completa, onde as regiões de elevado contraste são representadas a branco.

2ª Etapa – Dilatação da imagem

A segunda fase consiste na dilatação da imagem binária, resultante da fase anterior, com o objectivo de garantir que a imagem do contraste inclui todas as regiões de elevado contraste existentes na imagem. Para efectuar a dilatação da imagem do contraste, recorre-se a um filtro morfológico de dilatação [Serra93]. Dada a matriz $I(x,y)$ com a luminância do *pixel* (x,y) na imagem I , a dilatação da imagem I por um elemento estruturante de tamanho $(2n+1) \times (2n+1)$, $d^n(I)$, é dada pela expressão (3.10):

$$d^{(n)}I(x, y) = \max \{I(x - dx, y - dy) : -n \leq dx \leq n \wedge -n \leq dy \leq n\} \quad (3.10)$$

Nesta Tese, usa-se $n = 3$, valor que foi obtido empiricamente, através de testes exaustivos realizados, quer com texto gráfico, quer com texto de cena. A utilização de valores de n superiores a 3 diminui a eficiência da análise do contraste pois, desta forma, a imagem binária dilatada sobrepõe-se a muitas regiões que não correspondem a texto. Com valores de n inferiores a 3, os caracteres com maior espessura não são completamente sobrepostos pela imagem binária dilatada, o que leva à sua eliminação.

3ª Etapa – Comparação das imagens

A terceira, e última, fase do processamento correspondente à análise de contraste consiste na comparação entre as regiões conexas da imagem segmentada e a imagem de contraste dilatada. As regiões da imagem segmentada que não sejam sobrepostas em mais de 80% da sua área pela parte branca da imagem do contraste dilatada são descartadas por se considerar não corresponderem a caracteres.

Na Figura 3.20 apresenta-se um exemplo da aplicação das três fases da análise do contraste: na Figura 3.20 (b), ilustra-se o resultado da detecção de fronteiras; na Figura 3.20 (c), ilustra-se o resultado do cálculo do contraste local utilizando $Th_{cont}=10$; na Figura 3.20 (d), é ilustrado o resultado da dilatação, com $n=3$, da imagem binária resultante da adição das Figura 3.20 (b) e (c); na Figura 3.20 (d), apresenta-se o resultado da fase de segmentação e,

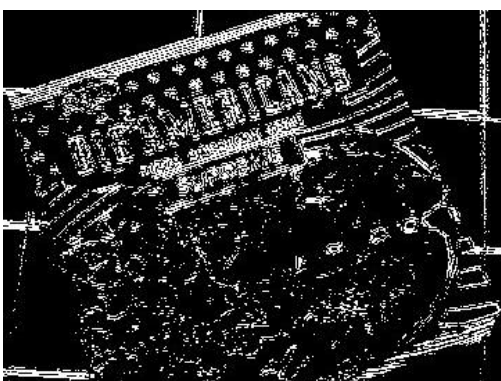
por último, a Figura 3.20 (e) ilustra o resultado da comparação da imagem segmentada com a imagem de contraste dilatada.



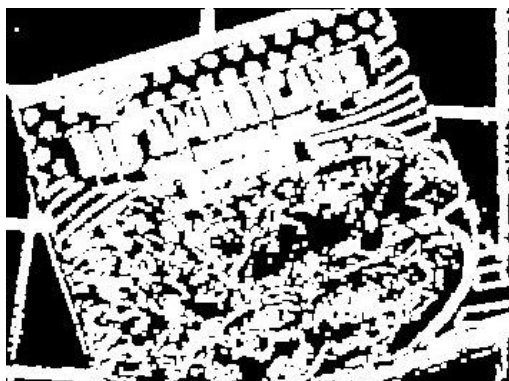
(a)



(b)



(c)



(d)



(e)



(f)

Figura 3.20 – Exemplo da aplicação da análise do contraste para efectuar a classificação das regiões: (a) imagem original; (b) imagem de fronteiras; (c) imagem de contraste com ($Th_{cont}=10$); (d) resultado da dilatação, com $n=3$, da imagem binária resultante da adição das imagens (b) e (c); (e) imagem segmentada; (f) imagem com as regiões classificadas como provável texto depois de efectuada a análise de contraste.

3.2.3.2 Análise Geométrica

A análise geométrica consiste na análise da forma e das dimensões das regiões conexas com o intuito de as classificar como caracteres de texto, ou não [Lienhart95, Zhong95, Messelodi99, Wu99, Lienhart00]. A análise geométrica é aplicada ao conjunto de regiões que foram classificadas como prováveis caracteres pela análise de contraste. Para efectuar a análise geométrica, foram utilizadas os seguintes parâmetros:

- **Altura** – Corresponde à altura da *bounding box* da região medida em *pixels*;
- **Largura** – Corresponde à largura da *bounding box* da região medida em *pixels*;
- **Relação entre a altura e a largura** da região;
- **Solidez** – Relação entre A_i , a área da região i , e A_{bb} , a área da *bounding box* correspondente à região i :

$$solidez(i) = \frac{A_i}{A_{bb}} \quad (3.11)$$

Os parâmetros anteriormente apresentados são aqueles que são utilizados usualmente na literatura para efectuar a classificação das regiões.

Como facilmente se compreende, os valores a adoptar para os limiares utilizados nas restrições geométricas a aplicar às regiões dependem do tamanho dos caracteres que se pretendem detectar. Nesta Tese, os valores para as restrições geométricas foram determinados empiricamente e baseados nos tipos de fontes Arial, Courier, Courier New e Times New Roman, a negrito (*bold*) e itálico, com os tamanhos 12, 24, 36 e 48 pt ($4 \times 4 \times 2 = 32$ tipos de fontes). A selecção recaiu sobre as fontes e tamanhos anteriormente referidos, uma vez que são estes os mais frequentemente utilizados no conjunto de imagens seleccionado para efectuar a avaliação do desempenho do algoritmo de extracção de texto em imagens (o qual pode ser observado na Secção 3.4.2); supondo que o conjunto de imagens de teste utilizado é representativo, então estas características serão as que mais frequentemente aparecem na vida real. Para além disso, segundo Lienhart [Lienhart00], fontes inferiores a 12 pt são demasiado pequenas para serem lidas em vídeo e, portanto, pouco frequentes uma vez que são inúteis. Os valores para as restrições geométricas que melhores resultados apresentaram para as imagens do conjunto de teste com uma resolução CIF são os seguintes:

- Altura da região $\in [4, A]$, $A = 0.25 \times \text{altura imagem}$;
- Largura da região $\in [1, L]$, $L = 0.25 \times \text{largura imagem}$;
- Relação altura/largura $\in [0.4, 10]$;
- Solidez $\in [0.15, 1]$.

A unidade de medida utilizada nas restrições geométricas supracitadas é o *pixel*. As regiões que cumprirem todas as restrições anteriormente citadas são classificadas como prováveis caracteres, enquanto as restantes são descartadas. Na Figura 3.21 pode observar-se a diferença entre a aplicação da análise geométrica isolada e em conjunto com a análise de

contraste. A Figura 3.21 (a) mostra o resultado da fase de segmentação; na Figura 3.21 (b), mostra-se o resultado da aplicação da análise geométrica sobre a imagem segmentada, e por último, na Figura 3.21 (c), ilustra-se o resultado da aplicação da análise geométrica em conjunto com a análise do contraste, sobre a imagem segmentada.



(a)



(b)



(c)

Figura 3.21 – Exemplo da aplicação da análise geométrica: (a) imagem segmentada; (b) imagem com as regiões classificadas como texto depois da aplicação da análise geométrica, sem a aplicação prévia da análise do contraste; (c) imagem com as regiões classificadas como texto depois da aplicação da análise geométrica e posteriormente à aplicação da análise do contraste.

3.2.4 Detecção de Palavras

A detecção de palavras visa o refinamento da detecção de caracteres atrás efectuada através da análise de contraste e da análise geométrica. Assim, para efectuar a extracção do texto, procede-se ao agrupamento das regiões, provenientes das fases anteriores e que foram classificadas como texto, de modo a formar palavras e linhas [Fletcher88, Zhong95, Messelodi99, Lienhart00]. No algoritmo desenvolvido no âmbito da presente Tese para efectuar a detecção de palavras, assume-se que o texto consiste em grupos de mais de duas regiões, alinhadas em qualquer direcção. Assume-se, também, que estas regiões estão relativamente próximas umas das outras se se encontrarem sobre a direcção da recta que

passa pelo centro das regiões de início e de fim da palavra. Estarão mais afastadas caso se encontrem na direcção perpendicular a essa recta. Desta forma, o algoritmo proposto para a detecção de palavras considera três fases distintas: agrupamento de regiões, eliminação de palavras sobrepostas e rotação do texto.

3.2.4.1 Agrupamento de Regiões

Esta fase visa o agrupamento das várias regiões resultantes da detecção de caracteres de modo a formar as palavras. Para efectuar os agrupamentos, há que estabelecer os critérios que permitem definir, com exactidão, o modo como estes vão ser constituídos. Tais critérios permitem não só definir os agrupamentos de regiões, mas também quais desses agrupamentos correspondem a palavras. Desta forma, os critérios para a formação de agrupamentos são os seguintes:

- **Proximidade** – As regiões que correspondem a caracteres devem estar suficientemente próximas umas das outras para que façam parte da mesma palavra. Assim, se se considerarem as regiões R_1, \dots, R_n , as quais foram classificadas como possíveis caracteres pelas fases anteriores, deve aplicar-se o critério de proximidade para que estas regiões possam ser agrupadas. Segundo este critério, testa-se para cada par de regiões R_i e R_j se a distância, d , entre os pontos correspondentes aos centros das *bounding boxes* de ambas é inferior a um determinado limiar, Th_{dist} . A distância d entre as duas regiões é definida da seguinte forma:

$$d = \sqrt{dx^2 + dy^2} < Th_{dist} \quad (3.12)$$

Em que dx e dy são as distâncias entre os centros das *bounding boxes* das duas regiões segundo as direcções horizontal e vertical, respectivamente. Com vista à definição prévia do valor de Th_{dist} , há que ter em consideração o facto da distância entre palavras ser tipicamente superior à distância entre caracteres que façam parte da mesma palavra. Assim, testes exaustivos permitiram definir Th_{dist} da seguinte forma:

$$Th_{dist} = 3 \times h_{min} \text{ onde } h_{min} = \min\{h_1, h_2, h_3\} \quad (3.13)$$

Em que $\{h_1, h_2, h_3\}$ representa a altura das *bounding boxes* das três regiões que deram origem à palavra. Quando a distância entre duas regiões é inferior a Th_{dist} , considera-se que estas fazem parte da mesma palavra.

- **Alinhamento** – As regiões que correspondem a caracteres devem estar alinhadas ao longo de uma dada direcção para que façam parte da mesma palavra. Assim, definiu-se um intervalo de tolerância para o alinhamento dessas mesmas regiões ao longo de uma direcção, dependendo esta tolerância da altura dos caracteres em questão. Para que uma região faça parte de uma dada palavra, a distância, d , entre o ponto correspondente ao seu centro e a recta que passa pelos centros das regiões de início e de fim da palavra tem que ser inferior a Th_{alin} , sendo o valor de Th_{alin} definido por:

$$Th_{alin} = \frac{h_{min}}{3} \quad (3.14)$$

Assim, a região R_i de coordenadas (x_{R_i}, y_{R_i}) faz parte da palavra $P_j = \{R_1, \dots, R_n\}$ de coordenadas $\{(x_{R_1}, y_{R_1}), \dots, (x_{R_n}, y_{R_n})\}$ se $d < Th_{alin}$, onde a distância, d , é dada pela expressão (3.15):

$$d = \left| \frac{x_{R_i} - m \times y_{R_i} - b}{\sqrt{1+b^2}} \right| \quad (3.15)$$

Dada a matriz $I(x,y)$ da imagem I , as coordenadas (x_R, y_R) da região, R , correspondem à posição do centro da *bounding box* da região, R , na matriz $I(x,y)$. O valor de b é a ordenada na origem, i.e. a distância, medida na vertical, da recta considerada à origem do referencial (considera-se como sendo a origem do referencial o ponto de coordenadas $I(0,0)$) e o m é o declive da recta que passa no centro das regiões R_l e R_n , dados pelas expressões (3.16) e (3.17), respectivamente:

$$b = y_{R_l} - m \times x_{R_l} \quad (3.16)$$

$$m = \frac{b}{a} \quad (3.17)$$

onde

$$a = \max\{x_{R_l}, \dots, x_{R_n}\} - \min\{x_{R_l}, \dots, x_{R_n}\} \quad (3.18)$$

$$b = y_{\max\{x_{R_l}, \dots, x_{R_n}\}} - y_{\min\{x_{R_l}, \dots, x_{R_n}\}} \quad (3.19)$$

Assim, quando a distância entre o centro da região e a recta que passa pelos centros das regiões de início e de fim da palavra é inferior a Th_{alin} , considera-se que a região faz parte da palavra;

- **Altura** – Para que façam parte da mesma palavra, as várias regiões correspondentes aos vários caracteres devem ter uma diferença mínima de altura. Para que isso se verifique, definem-se dois limiares: um para a altura mínima da palavra (Th_{hmin}) e outro para a altura máxima da palavra (Th_{hmax}). Testes exaustivos permitiram relacionar estes dois valores de limiar entre si da seguinte forma:

$$Th_{hmax} = \frac{5}{3}Th_{hmin} \text{ onde } Th_{hmin} = h_{min} \times 0.9 \quad (3.20)$$

A necessidade de definir dois limites de altura para os caracteres deve-se ao facto de, na mesma palavra, existirem normalmente letras de vários tamanhos. Assim, a região R_i faz parte da palavra P_j se:

$$Th_{hmin} < h(R_i) < Th_{hmax} \text{ onde } Th_{hmin} = h_{min}(P_j) \times 0.9 \quad (3.21)$$

Em que $h(R_i)$ e $h_{min}(P_j)$ são, respectivamente, a altura da região e a altura mínima da palavra;

- **Dimensão** – Os agrupamentos de regiões formados por mais de duas regiões são classificados como palavras, sendo os restantes eliminados. A eliminação das palavras com menos de três caracteres ocorre por se considerar que estas palavras possuem, nas aplicações consideradas, pouco valor semântico;
- **Luminância** – Para que façam parte da mesma palavra, as várias regiões correspondentes aos vários caracteres devem possuir uma luminância semelhante. Este critério é importante para evitar a geração de falsos caracteres, por exemplo devido a palavras com sombra ou palavras sobre fundo muito texturado. Para que isso se verifique, é definido um valor de limiar para a variação da luminância, Th_{lum} . Assim, a região R_i faz parte da palavra P_j se:

$$\left| Y_{(R_i)} - Y_{(P_j)} \right| < Th_{lum} \quad (3.22)$$

Em que $| |$ representa o valor absoluto da diferença entre a luminância da região e a luminância média da palavra.

O processo de agrupamento de regiões consiste na formação de palavras. Uma palavra válida $P_j = \{C_{i_1}, \dots, C_{i_n}\}$ é formada por, pelo menos, três caracteres (regiões) com as seguintes características:

- Os caracteres estão orientados segundo uma dada direcção;
- Os caracteres são vizinhos próximos;
- Os caracteres possuem uma diferença mínima de altura;
- Os caracteres possuem uma luminância média semelhante.

Inicialmente, todas as regiões classificadas como caracteres fazem parte de um único conjunto. Assim, o agrupamento de regiões pode ser descrito da seguinte forma:

- 1º A formação de palavras inicia-se com a procura de combinações de três caracteres que representem uma palavra válida, i.e. que cumpra os critérios supracitados;
- 2º Logo que seja encontrado um conjunto válido, os caracteres correspondentes são removidos do conjunto de caracteres e adicionados à nova palavra;
- 3º De seguida, para todos os caracteres restantes no conjunto de caracteres, verifica-se se estes cumprem os critérios, anteriormente estabelecidos, para se integrarem na nova palavra recém-formada. Aqueles que cumprirem os critérios são removidos do conjunto e adicionados à palavra.

Este processo de procura da próxima palavra válida e da adição dos caracteres que cumpram os critérios de formação de palavras, repete-se até que não seja possível formar mais palavras válidas, ou então, que não existam mais caracteres para agrupar. Os caracteres que não forem agrupados são eliminados.

O resultado da detecção de palavras é representado numa imagem binária, na qual se representa o texto a branco e o fundo da imagem a preto, a qual é designada como imagem binária.

O efeito do agrupamento das regiões em palavras é ilustrado na Figura 3.22. A Figura 3.22 (b) ilustra as regiões que foram classificadas como texto pela análise do contraste e pela análise geométrica. Na Figura 3.22 (c) ilustra-se o resultado do processo de agrupamento das regiões de modo a formarem conjuntos de regiões que correspondam a palavras.



(a)



(b)



(c)

Figura 3.22 – Exemplo da detecção de palavras: (a) imagem original; (b) imagem segmentada depois de efectuada a análise de contraste e a análise geométrica; (c) imagem depois da formação de palavras.

3.2.4.2 Eliminação de Palavras Sobrepostas

Esta fase consiste na eliminação de eventuais palavras que se sobreponham, i.e. palavras onde as *bounding boxes* de cada uma se intersectam. As sobreposições de palavras dão-se normalmente devido à formação de palavras falsas. Estas resultam do agrupamento de regiões que não fazem parte do texto mas que, pela sua forma, dimensão e posição, tenham sido classificadas como tal, por exemplo resultantes de sombras associadas a caracteres verdadeiros. Estas regiões não são incluídas nas palavras do texto principal durante a formação das mesmas, por possuírem um valor de luminância muito diferente. Um exemplo deste tipo de regiões é ilustrado na Figura 3.23. Nas Figura 3.23 (a) e (b) são ilustrados casos

de sobreposição de palavras facilmente detectáveis através da observação; nas Figura 3.23 (c) e (d) ilustra-se uma situação em que a sobreposição só é visível após a rotação do texto para a posição horizontal. De forma a facilitar a visualização das palavras sobrepostas, aquelas que são formadas a partir da junção de regiões que fazem parte do texto estão representadas através da cor branca, as que foram formadas a partir da junção de regiões que não fazem parte do texto estão representadas a vermelho. As situações de sobreposição de palavras estão assinaladas por *bounding boxes* de cor branca e vermelha.



(a)



(b)



(c)



(d)

Figura 3.23 – Exemplo de sobreposição de palavras: (a) e (c) imagens com sobreposição de palavras antes de efectuada a rotação do texto; (b) e (d) as imagens com sobreposição de palavras depois da rotação do texto.

Sempre que duas ou mais palavras se sobrepõem, i.e. as *bounding boxes* de cada uma delas se intersectam, aquelas que possuem menor área são eliminadas restando apenas a maior. A área de uma palavra, AP_i , é definida da seguinte forma:

$$AP_i = \sum_{k=1}^n AC_{i_k} \quad (3.23)$$

Em que AC_i representa a área de cada carácter (número de *pixels* do carácter) que forma a palavra e n o número total de caracteres existentes na palavra.

3.2.4.3 Rotação do Texto

Esta fase consiste na rotação do texto para a posição horizontal de forma a permitir o seu reconhecimento por parte dos sistemas OCR. Para tal, são considerados dois tipos de texto:

- **Texto vertical** – Este tipo de texto caracteriza-se pela existência de palavras com ângulos de inclinação superiores a 70 graus, formadas por caracteres horizontais (com ângulos de inclinação iguais a 0 graus);
- **Texto inclinado** – Este tipo de texto caracteriza-se pela existência de palavras com ângulos de inclinação inferiores a 70 graus, formadas por caracteres com uma inclinação idêntica à das palavras.

Exemplos dos dois tipos de texto, vertical e inclinado, são ilustrados na Figura 3.24 (a) e (b), respectivamente. O caso simples do texto horizontal aparece aqui como um caso particular do texto inclinado.

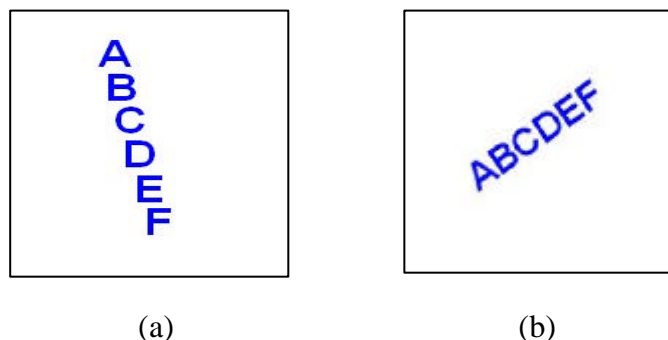


Figura 3.24 – Exemplo de texto vertical (a) e inclinado (b).

Para o cálculo da rotação do texto, assume-se que este se encontra escrito da esquerda para a direita. No caso particular de texto com uma inclinação de 90 graus, assume-se que este está escrito de cima para baixo. Assim, a rotação do texto pode ser efectuada em dois passos:

1º Cálculo do ângulo de rotação – O ângulo de rotação, θ , é definido como o ângulo entre o eixo dos xx e a recta que passa pelo centro das regiões de início e de fim da palavra. Assim, se se considerar a palavra, P , formado pelo conjunto de caracteres $\{C_1, \dots, C_n\}$ com *bounding box* de coordenadas $\{(x_{C_1}, y_{C_1}), \dots, (x_{C_n}, y_{C_n})\}$, o seu ângulo de rotação, θ_T , é dado por:

$$\theta_T = \tan^{-1}\left(\frac{b}{a}\right) \quad (3.24)$$

onde

$$a = \max\{x_{C_1}, \dots, x_{C_n}\} - \min\{x_{C_1}, \dots, x_{C_n}\} \quad (3.25)$$

$$b = y_{\max\{x_{C_1}, \dots, x_{C_n}\}} - y_{\min\{x_{C_1}, \dots, x_{C_n}\}} \quad (3.26)$$

Dada a matriz $I(x,y)$ da imagem I , as coordenadas (x_c, y_c) do carácter C , correspondem à posição do centro da *bounding box* do carácter, C , na matriz $I(x,y)$.

2º **Rotação do texto** – A rotação do texto efectua-se utilizando o ângulo anteriormente calculado e é efectuada de forma diferenciada para o texto vertical e inclinado, do seguinte modo:

- **Texto vertical** – No texto vertical, a rotação da palavra consiste na translação dos caracteres para a posição horizontal sem a rotação dos mesmos;
- **Texto inclinado** – No texto inclinado, a rotação da palavra consiste na translação dos caracteres para a posição horizontal acompanhada da sua rotação. Para efectuar a rotação do texto foi utilizado o algoritmo de rotação proposto por Alan Paeth [Paeth86].

O efeito da rotação do texto pode ser observado na Figura 3.25. As Figura 3.25 (a), (c) e (e) ilustram a rotação de texto vertical, enquanto as Figura 3.25 (b), (d) e (f) ilustram a rotação de texto inclinado.



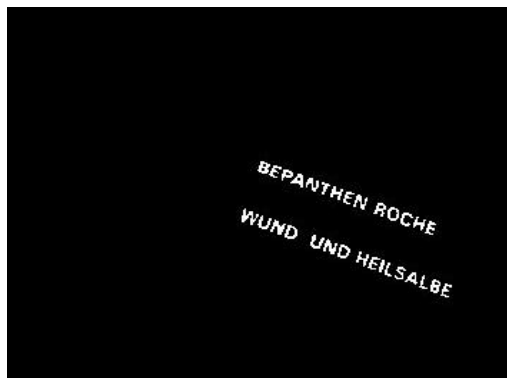
(a)



(b)



(c)



(d)



Figura 3.25 – Exemplo da rotação de texto: (a) e (b) originais de texto vertical e inclinado, respectivamente; (c) e (d) texto detectado antes da rotação; (e) e (f) texto detectado depois da rotação.

3.3 Reconhecimento de Texto

Esta fase visa o reconhecimento do texto existente na imagem usando as palavras determinadas na fase anterior, utilizando-se para tal um sistema OCR. Enquanto alguns autores desenvolveram os seus próprios sistemas OCR para a tarefa do reconhecimento [Lienhart95, Zhou97, Sato99], outros utilizaram versões comerciais [Zhong95, Wu99, Li02, Lienhart02]. Na presente Tese, utilizam-se dois sistemas OCR: um sistema OCR comercial (OmniPage Pro 12.0 [ScanSoft]) e um sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95] que foi integrado na aplicação desenvolvida nesta Tese em virtude de haver *software* disponível. Note-se que o desenvolvimento do sistema OCR propriamente dito, não fazia parte dos objectivos desta Tese e, por isso, são usado sistemas disponíveis.

3.3.1 Sistema OCR Comercial

O sistema OCR comercial utilizado é o OmniPage Pro 12.0 [ScanSoft]. Este sistema utiliza tecnologia baseada em redes neuronais para efectuar o reconhecimento dos caracteres. Devido à compra de outros sistemas OCR por parte da ScanSoft, proprietária do OmniPage Pro 12.0, foi integrado neste sistema a melhor tecnologia de vários sistemas OCR, tais como a *Predictive Word Recognition* (POWR) do WordScan Plus 4.0 [ScanSoft]. Esta tecnologia permite fazer o reconhecimento de uma palavra inteira, sem ter que reconhecer os seus caracteres individualmente. Da mesma forma, foi ainda integrada a tecnologia baseada na inteligência contextual do Recognita [ScanSoft].

O OmniPage Pro 12.0 consegue efectuar o reconhecimento do texto a partir das imagens originais não necessitando que lhes seja aplicado qualquer pré-processamento. A detecção é feita para texto horizontal e para texto inclinado. Todavia, este último, necessita de estar todo escrito na mesma direcção, uma vez que o OCR, para efectuar o reconhecimento do texto, começa por calcular uma orientação para o mesmo. O texto que não estiver orientado segundo a orientação previamente calculada é classificado como fazendo parte de gráficos ou imagens. Esta característica do OmniPage Pro 12.0 permite efectuar uma avaliação mais objectiva do desempenho do algoritmo proposto para a extracção de texto em imagens, quer para o texto horizontal, quer para o texto inclinado, uma vez que consegue efectuar o

reconhecimento dos mesmos a partir das imagens originais mas também a partir das imagens processadas segundo o método de extracção aqui proposto.

3.3.2 OCR Integrado na Aplicação

Na aplicação de extracção de texto desenvolvida foi integrado o sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95]. Este sistema foi concebido para reconhecer texto proveniente de sistemas de extracção de texto em imagens e vídeo, onde os caracteres resultantes da detecção podem ser constituídos por várias regiões de diferentes cores. Desta forma, o texto candidato ao reconhecimento deve cumprir dois critérios para que possa ser reconhecido pelo sistema:

- 1º A dimensão de pelo menos uma das regiões que formam cada caracter deve exceder uma percentagem mínima da dimensão deste;
- 2º A variação da cor em cada caracter não deve exceder um determinado limite.

O primeiro requisito baseia-se no pressuposto de que, pelo menos, o corpo principal do caracter é segmentado numa grande região monocromática pelo algoritmo de segmentação na fase da detecção do texto. O segundo requisito baseia-se no pressuposto de que um caracter raramente é constituído por cores muito diferentes.

Para efectuar o reconhecimento óptico dos caracteres, cada caracter é processado da seguinte forma:

- 1º O mapa binário de cada caracter é dividido em nove segmentos, como ilustrado na Figura 3.26 (a);
- 2º Para cada segmento, determina-se o número de *pixels* e verifica-se se pertence a uma das quatro classes descritas pelos 16 elementos de direcção (máscaras 2×2), Figura 3.26 (b): (H) horizontal; (V) vertical; (R) transversal direito e (L) transversal esquerdo. Isto origina num vector de características com 36 posições.

Assim, é utilizado um vector de características para efectuar o reconhecimento, que utiliza como características o número de *pixels*, a cor, a posição do segmento e a classe do segmento. O vector de características é normalizado e comparado com os vectores correspondentes aos caracteres da base de dados. Para efectuar a classificação do vector, é utilizado o algoritmo proposto por Cover et. al. [Cover67]. A base de dados foi treinada para 12 tipos de fontes diferentes; todavia, é possível efectuar o seu treino para mais tipos de fontes.

Este algoritmo de OCR está longe de ser perfeito, nomeadamente quando comparado com os pacotes de *software* comerciais; no entanto, pode ser facilmente integrado no algoritmo de extracção de texto em imagens e sequências de vídeo proposto nesta Tese.

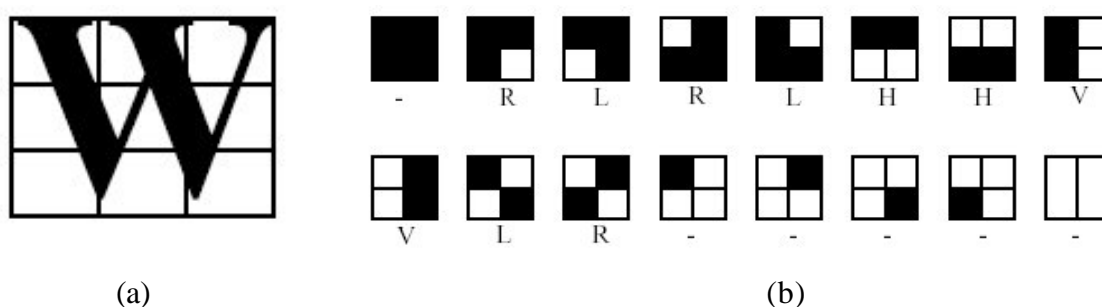


Figura 3.26 – Cálculo dos vectores de características para o reconhecimento óptico de caracteres [Lienhart95]: (a) divisão do character em nove segmentos (b) os 16 elementos de direcção.

3.4 Avaliação de Desempenho

Nesta secção vai efectuar-se a avaliação do desempenho do algoritmo de extracção de texto em imagens proposto neste capítulo. Como o objectivo final do algoritmo é o reconhecimento dos caracteres, torna-se natural a utilização de um sistema OCR para efectuar o reconhecimento dos mesmos. Tal como se disse, nos testes efectuados no âmbito da presente Tese, utilizam-se dois sistemas OCR: um sistema OCR comercial – OmniPage Pro 12.0 [ScanSoft] e um sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95], o qual foi integrado na aplicação de extracção de texto em imagens desenvolvida.

3.4.1 Métricas de Desempenho

Duas métricas são universalmente aceites para efectuar a avaliação do desempenho de sistemas de reconhecimento de informação [Salton83]: a precisão e a *recall*. Estas métricas foram adaptadas por vários investigadores para avaliar o desempenho dos seus algoritmos de extracção de texto em imagens [Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Na presente Tese, adoptaram-se as métricas *recall* e precisão para avaliar o desempenho do algoritmo de extracção de texto proposto. Esta escolha justifica-se essencialmente por permitir a comparação dos resultados obtidos por vários autores, uma vez que estas métricas são aquelas cuja utilização é mais comum. A definição adoptada para as métricas de desempenho usadas nesta Tese para avaliar as capacidades de detecção e reconhecimento de texto é a seguinte:

- **Avaliação da detecção de texto** – A avaliação do desempenho em termos da detecção do texto exprime a capacidade do algoritmo proposto em detectar correctamente caracteres de texto. Tipicamente, um aumento da precisão pode ser feita à custa de uma diminuição da *recall* e vice-versa; aumentar as duas métricas simultaneamente é uma tarefa mais complicada. Para isso, foram utilizadas as métricas *recall* e precisão definidas do seguinte modo:
 - ♦ **Recall** – Relação entre o número de caracteres correctamente detectados e o número de caracteres da *ground truth* da imagem.

$$Recall = \frac{CCD}{GT} \quad (3.27)$$

Onde *CCD* é o número de caracteres correctamente detectados e *GT* é o número de caracteres da *ground truth* da imagem. Esta métrica dá uma ideia da capacidade do algoritmo em detectar os caracteres que efectivamente existem independentemente do número de falsos caracteres que são detectados;

- ♦ **Precisão** – Relação entre o número de caracteres correctamente detectados e o número total de caracteres detectados.

$$Precisão = \frac{CCD}{TCD} \quad (3.28)$$

Onde *TCD* é o número total de caracteres detectados. Esta métrica dá uma ideia da capacidade do algoritmo em termos de detectar só e apenas caracteres que efectivamente existem e por isso simultaneamente da sua capacidade de não gerar falsos caracteres.

- **Avaliação do reconhecimento de texto** – A avaliação do desempenho em termos de reconhecimento do texto exprime a capacidade do sistema usado para reconhecer correctamente o texto existente nas imagens. Este tipo de desempenho é determinado, não só, pela capacidade do algoritmo proposto para detectar texto mas também pela capacidade dos OCRs usados para reconhecer o texto detectado. Para fazer este tipo de avaliação, usaram-se as métricas tradicionais, precisão e *recall*, definidas do seguinte modo:

- ♦ **Recall** – Relação entre o número de caracteres que foram correctamente reconhecidos e o número de caracteres da *ground truth* da imagem.

$$Recall = \frac{CCR}{GT} \quad (3.29)$$

Onde *CCR* representa o número de caracteres que foram correctamente reconhecidos pelo sistema OCR;

- ♦ **Precisão** – Relação entre o número de caracteres que foram correctamente reconhecidos e o número total de caracteres reconhecidos pelo sistema OCR.

$$Precisão = \frac{CCR}{CSO} \quad (3.30)$$

Onde *CSO* corresponde ao número total de caracteres na saída do OCR.

Valores superiores a 80% para as métricas *recall* e precisão são considerados por muitos investigadores como sendo bons resultados [Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Para que o algoritmo apresente um desempenho elevado, é necessário que se obtenham valores elevados, quer para a métrica *recall*, quer para a métrica precisão. Assim, valores elevados na *recall* e baixos na precisão, indicam um elevado número de falsas detecções, i.e. o algoritmo para além do texto que faz parte da *ground truth* da imagem classifica também, como texto, outras regiões. Valores baixos na *recall* e elevados na precisão, indicam que o

número de falsas detecções é baixo, mas também que um elevado número de caracteres que fazem parte da *ground truth* de texto da imagem não foram detectados. Tipicamente, é difícil obter simultaneamente valores muito elevados de precisão e *recall* (quando um aumenta muito o outro tende a diminuir), devendo por isso procurar-se o melhor compromisso entre os dois valores; este compromisso depende do tipo de aplicações. Há aplicações onde ter falsos alarmes não implica, necessariamente, um erro grave, enquanto que noutras aplicações não detectar algo da *ground truth* é gravíssimo.

3.4.2 Condições e Metodologia de Avaliação do Desempenho

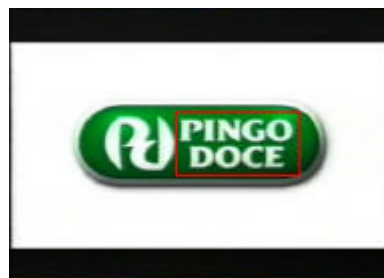
Com o objectivo de avaliar o desempenho do algoritmo proposto para a extracção de texto em imagens, foi utilizado um conjunto de 60 imagens de teste. Estas imagens foram seleccionadas a partir de emissões de TV e foram capturadas com o recurso à placa de captura de vídeo Pinnacle Linx Video Input Cable, usando-se resoluções espaciais (luminância) compreendidas entre 352×208 e 384×288 *pixels*. As imagens contêm texto de cena e texto gráfico, alinhado em qualquer direcção, com múltiplas fontes e tamanhos. As imagens seleccionadas para os testes foram retiradas de títulos e apresentações de filmes onde predomina o texto gráfico, bem como de anúncios e programas de informação onde predomina o texto de cena. A avaliação do desempenho do algoritmo proposto será efectuada tanto para o texto gráfico, como para texto de cena.

Na avaliação do desempenho em termos de detecção do texto e antes de processar cada imagem com o algoritmo de detecção de texto proposto, foi definida para cada imagem a sua *ground truth* em termos de texto ou seja determinou-se manualmente quais os caracteres de texto existente em cada imagem. Para tal, efectuou-se para cada imagem um levantamento manual dos caracteres existentes na mesma e que são relevantes para a detecção de texto, i.e. aqueles que formam palavras. São consideradas palavras a detectar todos os conjuntos de caracteres que possuam as seguintes características:

- Formados por mais de dois caracteres da mesma cor e que não se toquem entre si;
- Alinhados segundo uma dada direcção;
- Altura de cada palavra $\in [4, A]$, $A = 0.25 \times \text{altura imagem}$;
- Largura de cada palavra $\in [1, L]$, $L = 0.25 \times \text{largura imagem}$.

A unidade de medida utilizada na definição do texto que faz parte da *ground truth* é o *pixel*.

A Figura 3.27 apresenta vários exemplos de imagens que fazem parte do conjunto de teste. Através da sua observação, pode constatar-se a variedade de imagens escolhidas, quer ao nível dos diferentes tipos de texto, quer ao nível de fontes, tamanhos de caracteres e alinhamento do texto. Na Figura 3.27 (a) ilustram-se exemplos de imagens onde predomina o texto de cena e na Figura 3.27 (b) apresentam-se exemplos onde o texto predominante é gráfico. O texto que faz parte da *ground truth* de cada imagem encontra-se circunscrito por caixas de cor branca ou vermelha.





(a)







(b)

Figura 3.27 – Exemplos de imagens que fazem parte do conjunto de teste: (a) imagens onde predomina o texto de cena; (b) imagens onde predomina o texto gráfico.

Nesta secção contemplar-se-á tanto a avaliação do desempenho do algoritmo na detecção e classificação das regiões conexas como texto ou não texto, como a avaliação do desempenho em termos de reconhecimento do texto.

3.4.3 Resultados e Comentários

Nesta secção são apresentados os resultados obtidos para as várias avaliações de desempenho efectuadas. Na Tabela 3.1 apresentam-se os valores dos vários parâmetros utilizados para a configuração do algoritmo de detecção de texto proposto. Estes valores foram aqueles que se revelaram mais eficazes para o conjunto de imagens utilizado na avaliação do desempenho.

Tabela 3.1 – Parâmetros utilizados para a avaliação do desempenho.

Limiares para a segmentação	Th_{GD}	
Fase de <i>Split</i>	30	
Fase de <i>Merge</i>	35	
Limiares para a análise de contraste	Th_{cont}	
Contraste entre regiões	10	
Restrições geométricas das regiões	Min	Max
Largura	1	$0,25 \times (\text{largura imagem})$
Altura	4	$0,25 \times (\text{altura imagem})$
Relação altura/largura	0,4	10
Solidez	0,15	1

3.4.3.1 Avaliação da Detecção de Texto

Na avaliação do desempenho em termos da detecção de texto foram processadas 60 imagens e para cada uma delas determinou-se, manualmente, se cada carácter foi detectado correctamente ou não pelo algoritmo proposto. A detecção correcta, ou não, dos caracteres é determinada com recurso à inspecção visual das imagens binárias criadas pelo algoritmo. A avaliação da detecção de texto foi efectuada para todo o texto tomado como *ground truth* nas imagens de teste. Além disso, fez-se também a avaliação da detecção de texto considerando apenas texto horizontal tendo então só sido usado como *ground truth* as partes relevantes de texto (horizontais) existentes nas imagens de teste. Os resultados obtidos para a detecção de texto horizontal na totalidade das imagens podem ser observados na Tabela 3.2.

Tabela 3.2 – Resultados médios obtidos para a detecção de texto horizontal para a totalidade das imagens.

Tipos de Texto	Nº caracteres	Recall	Precisão
Texto horizontal de cena	407	0.823	0.898
Texto horizontal gráfico	885	0.911	0.920
Totalidade do texto horizontal na <i>ground truth</i>	1292	0.883	0.914

Os resultados obtidos na avaliação do desempenho do algoritmo na detecção de todo o texto que faz parte da *ground truth*, i.e. texto horizontal, inclinado e vertical, para a totalidade das imagens podem ser observados na Tabela 3.3.

Tabela 3.3 – Resultados médios obtidos para a detecção de todo o texto para a totalidade das imagens.

Tipos de Texto	Nº caracteres	Recall	Precisão
Texto de cena	536	0.791	0.895
Texto gráfico	908	0.913	0.909
Totalidade do texto na <i>ground truth</i>	1444	0.868	0.904

O desempenho do algoritmo proposto, quer para texto horizontal, quer para todo o texto que faz parte da *ground truth*, pode considerar-se elevado, pois obtiveram-se valores para as métricas *recall* e precisão que são considerados, por outros investigadores, como sendo muito bons resultados [Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Da análise dos resultados pode constatar-se um melhor desempenho, ainda que ligeiro, na detecção do texto horizontal em relação ao texto genérico, tanto para a *recall* como para a precisão da ordem dos 1.5% e 1%, respectivamente. Nos testes efectuados com todo o texto que faz parte da *ground truth*, obtiveram-se, em termos de detecção de texto, valores para *recall* na ordem de 87%, factor

indicativo de apenas cerca de 13% do texto não ter sido detectado. No que respeita aos valores da precisão para a detecção de texto, estes andaram na ordem dos 90%, indicando que somente cerca de 10% dos caracteres foram falsamente detectados. No entanto, feita a análise em termos parciais, pode dizer-se que: os resultados obtidos apresentam valores mais elevados para o texto gráfico, comparativamente ao texto de cena, facto que era expectável.

- **Texto gráfico** – Para o texto gráfico os resultados obtidos tanto para a *recall* como para a precisão apresentam valores de cerca de 91%. Tais valores indicam que, para texto gráfico, apenas cerca 9% dos caracteres não foram detectados ou foram falsamente detectados;
- **Texto de cena** – Para o texto de cena, verificou-se um decréscimo significativo no valor da *recall* em relação ao texto gráfico, sendo este valor da ordem de 79%. Este resultado indica que, para texto de cena, cerca de 21% dos caracteres não foram detectados. No que respeita ao valor da precisão o decréscimo foi menor, passando a situar-se na ordem dos 89%, valor indicativo de que somente cerca de 11% dos caracteres foram falsamente detectados. A menor diminuição da precisão quando comparada com a *recall* deve-se essencialmente aos valores dos parâmetros utilizados nas fases de detecção de caracteres e de formação de palavras, i.e. privilegia-se a detecção correcta dos caracteres ainda que à custa de não detectar alguns caracteres. As falhas na detecção de caracteres acentuam-se (diminuição da *recall*) no texto de cena devido à sua maior diversidade de fontes e tamanhos.

Os resultados obtidos apresentam valores mais elevados para o texto gráfico, comparativamente ao texto de cena, facto que era expectável.

Note-se que este compromisso entre os valores das métricas *recall* e precisão depende dos valores adoptados para os vários parâmetros de configuração do algoritmo os quais estão intimamente relacionados com o tipo de aplicação em causa.

A detecção do texto gráfico apresenta valores superiores tanto para a *recall* como para a precisão, quer se trate de texto escrito unicamente na horizontal, quer se trate de texto escrito em qualquer direcção. O motivo principal para tal facto tem a ver com as características do texto gráfico, normalmente mais contrastado em relação ao fundo da imagem do que o texto de cena e mais bem definido do que este. Para além disso, este último apresenta uma maior variedade de fontes e tamanhos, o que o torna mais difícil de diferenciar de outras estruturas existentes nas imagens. Exemplos dessas dificuldades podem ser observados na Figura 3.29, Figura 3.29 e Figura 3.30.

Na Figura 3.30 podem observar-se várias regiões que são confundidas com o texto devido à sua forma e posicionamento.

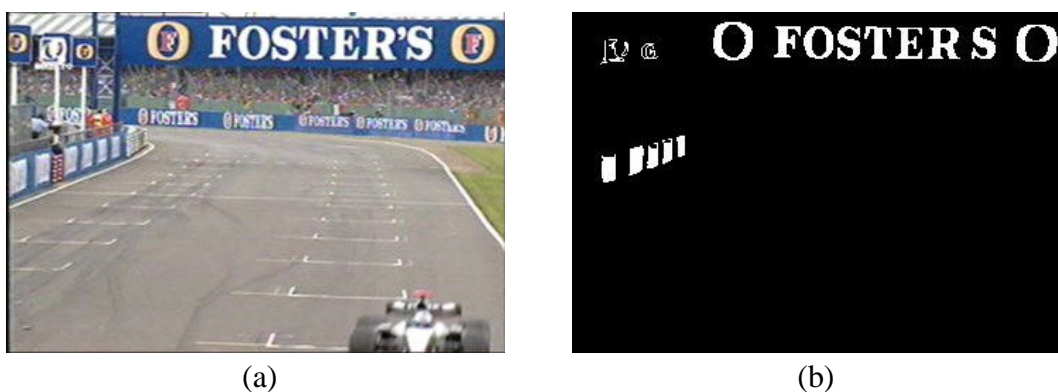


Figura 3.28 – Exemplo de regiões falsamente classificadas como texto devido à sua forma e posicionamento: (a) imagem original e (b) imagem binária com o resultado da detecção de texto.

Na Figura 3.29 ilustram-se falhas na detecção de caracteres por estes possuírem um baixo contraste em relação ao fundo da imagem, o que origina a sua fusão com as regiões na sua vizinhança. O texto marcado com a *bounding box* vermelha na Figura 3.29 (a) confunde-se com o fundo da imagem devido ao seu baixo contraste.

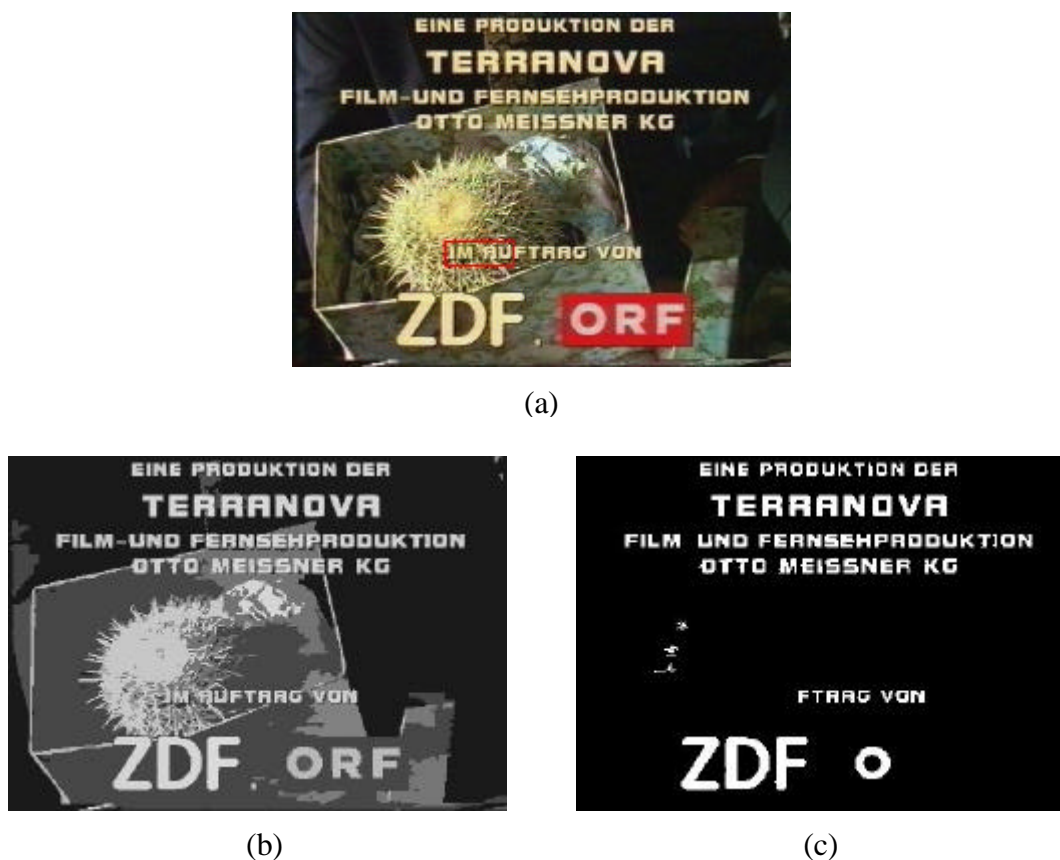


Figura 3.29 – Exemplo de falhas na detecção de texto devido ao baixo contraste existente entre o texto e fundo da imagem: (a) imagem original; (b) imagem com o resultado da segmentação e (c) imagem binária com o resultado da detecção de texto.

Na Figura 3.30, para além de falhas na detecção de caracteres, semelhantes às descritas anteriormente, pode ainda ser observado um caso em que o texto não é detectado por os caracteres se tocarem. Assim, apesar da palavra marcada com a *bounding box* vermelha na Figura 3.30 (a) possuir 4 caracteres, o algoritmo não a classifica como tal, uma vez que só reconhece dois caracteres.



(a)



(b)



(c)

Figura 3.30 – Exemplo de falhas na detecção de texto devido ao contacto entre os caracteres:
(a) imagem original; (b) imagem com o resultado da segmentação e (c) imagem binária com o resultado da detecção de texto.

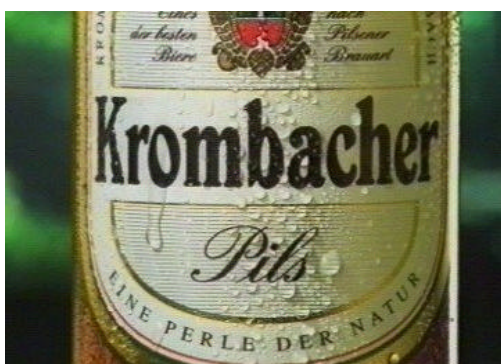
Na Tabela 3.4 pode observar-se a percentagem de caracteres, tanto para o texto escrito na horizontal como para a totalidade do texto, não detectados e também a percentagem de caracteres detectados que embora danificados (por exemplo, caracteres onde faltam algumas partes) podem ser reconhecidos por um humano.

Tabela 3.4 – Resultados médios obtidos para a detecção de texto em termos de caracteres não detectados e caracteres danificados.

Tipos de texto	Texto horizontal		Totalidade do texto	
	Caracteres não detectados	Caracteres danificados detectados	Caracteres não detectados	Caracteres danificados detectados
Texto de cena	11.8%	5.9%	14.2%	6.7%
Texto gráfico	3.4%	5.5%	3.3%	5.4%
Totalidade do texto	6.0%	5.7%	7.3%	5.9%

Tanto para os caracteres não detectados, como para os caracteres danificados detectados, verificou-se, quer para o texto horizontal, quer para a globalidade do texto, um melhor desempenho para o texto gráfico. Este aumento de desempenho para o texto gráfico deve-se essencialmente às características do texto gráfico, normalmente mais contrastado em relação ao fundo da imagem do que o texto de cena e mais bem definido do que este. A ocorrência de caracteres que não são detectados ou que são detectados mas estão danificados deve-se, essencialmente, ao contacto dos caracteres com regiões com as quais o seu contraste é pequeno, bem como à ocorrência de texto de pequenas dimensões mal contrastado com o fundo.

Na Figura 3.31 são ilustrados exemplos de caracteres danificados mas que podem ser reconhecidos por um humano: na Figura 3.31 (c), o *e* e o *r* no fim da palavra estão unidos; na Figura 3.31 (d), o *r* no final do texto está unido a uma região erradamente classificada como texto. Ainda que estas anomalias possam impedir a detecção automática deste texto, ou pelo menos o seu reconhecimento, um observador humano não teria qualquer dificuldade na sua identificação.



(a)



(b)

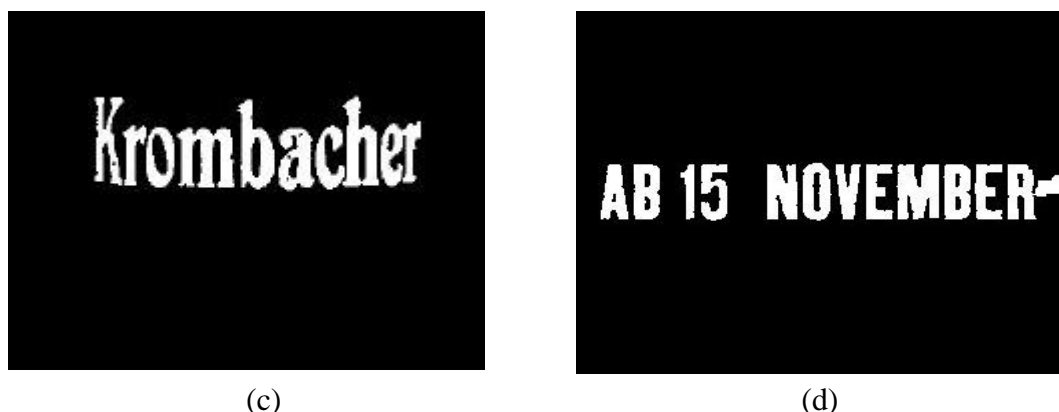


Figura 3.31 – Exemplo de caracteres danificados: (a) e (b) imagens originais; (c) e (d) imagens binárias com texto detectado.

3.4.3.2 Avaliação do Reconhecimento de Texto

Na avaliação do desempenho em termos de reconhecimento de texto são utilizadas as imagens resultantes da fase de detecção. Tal como foi referido no início da secção 3.4 para efectuar o reconhecimento, foram utilizados dois sistemas OCR. Todavia, antes de processar as imagens com o sistema OCR OmniPage Pro 12.0, a sua resolução é aumentada uma vez que este sistema OCR está preparado para efectuar o reconhecimento de caracteres em imagens com elevada resolução (superior a 75ppi). De seguida, as 60 imagens são processadas e para cada uma delas é determinado se cada carácter detectado é reconhecido correctamente ou não. Os resultados obtidos para o reconhecimento do texto horizontal, podem ser observados na Tabela 3.5 .

Tabela 3.5 – Resultados médios obtidos para o reconhecimento do texto horizontal.

Tipos de texto	OCR [Lienhart95]		OCR OmniPage Pro 12.0	
	<i>Recall</i>	<i>Precisão</i>	<i>Recall</i>	<i>Precisão</i>
Texto horizontal de cena	0.565	0.610	0.735	0.838
Texto horizontal gráfico	0.729	0.743	0.916	0.934
Totalidade do texto horizontal na <i>ground truth</i>	0.677	0.703	0.859	0.906

Os resultados obtidos na avaliação do desempenho de reconhecimento de todo o texto que faz parte da *ground truth*, i.e. texto horizontal, inclinado e vertical, para os vários tipos de texto e para a totalidade das imagens podem ser observados na Tabela 3.6.

Tabela 3.6 – Resultados médios obtidos para o reconhecimento de todo o texto.

Tipos de texto	OCR [Lienhart95]		OCR OmniPage Pro 12.0	
	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão
Texto de cena	0.509	0.590	0.698	0.815
Texto gráfico	0.728	0.721	0.902	0.919
Totalidade do texto na <i>ground truth</i>	0.647	0.677	0.826	0.884

No reconhecimento do texto, tal como na detecção, também se verificou para o texto horizontal um melhor desempenho tanto para a *recall* como para a precisão da ordem dos 3.3 e 2.2% para o OCR OmniPage Pro 12.0 e de 3 e 2.6%, para o OCR desenvolvido por Lienhart [Lienhart95], respectivamente. Os melhores valores tanto para a *recall* como para a precisão quando o texto está escrito na horizontal devem-se essencialmente à maior facilidade em identificar e eliminar falsas detecções durante o processo de formação de palavras.

Quando foi considerado todo o texto que faz parte da *ground truth*, obtiveram-se, utilizando o sistema OCR OmniPage Pro 12.0, valores para a *recall* da ordem de 83%, factor indicativo de apenas cerca de 17% do texto não ter sido reconhecido. No que respeita aos valores da precisão, estes andaram na ordem dos 88%, indicando que somente cerca de 12% dos caracteres foram falsamente reconhecidos. Com a utilização do sistema OCR desenvolvido por Lienhart [Lienhart95], os valores da *recall* e da precisão foram da ordem dos 65% e 68%, respectivamente.

De seguida será feita a análise de forma separada para o texto gráfico e para o texto de cena, tendo em conta todo o texto que faz parte da *ground truth*, por possibilitar uma análise mais objectiva do desempenho do algoritmo. Assim, obtiveram-se em termos de reconhecimento de texto, os seguintes valores:

- **Texto gráfico** – Quando o texto é gráfico, os valores obtidos para a *recall* e para a precisão, utilizando o sistema OCR OmniPage Pro 12.0, são da ordem dos 90 e 92%, respectivamente. Tais valores indicam que, para texto gráfico, apenas cerca de 10% dos caracteres não foram reconhecidos e que 8% foram falsamente identificados. Com a utilização do sistema OCR desenvolvido por Lienhart [Lienhart95], a *recall* e a precisão apresentam valores de cerca 73 e 72%, respectivamente. Este melhor desempenho de reconhecimento para imagens com texto gráfico, deve-se essencialmente às características do texto gráfico, normalmente mais contrastado em relação ao fundo da imagem, características estas que o tornam mais fácil de distinguir de outras estruturas existentes na imagem e, assim, facilitam a sua detecção e reconhecimento. Para além disso, a maioria das fontes utilizadas no texto gráfico correspondem, usualmente, a fontes bem conhecidas dos sistemas OCR, tais como Arial, Courier e Times New Roman;
- **Texto cena** – Quando predomina o texto de cena, e tal como na detecção de texto, também para o reconhecimento se verifica um decréscimo dos valores da *recall* e da precisão em relação ao texto gráfico. Com a utilização do sistema OCR OmniPage Pro 12.0, estes são da ordem de 70 e 82% para a *recall* e precisão, respectivamente. Estes

resultados indicam que, para texto de cena, cerca de 30% dos caracteres não foram reconhecidos e que 18% foram falsamente identificados. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], verifica-se igualmente um decréscimo da *recall* e da precisão em relação ao texto gráfico. Estas apresentam agora valores de cerca de 51 e 59%, respectivamente. O reconhecimento do texto de cena é penalizado, essencialmente devido ao elevado número de diferentes fontes que existem no mesmo ou fontes que foram mais ou menos alteradas, o que o torna difícil de diferenciar de outras estruturas existentes nas imagens.

A diferença de desempenho evidenciada pelos dois sistemas OCR resulta, essencialmente, do treino limitado efectuado à base de dados utilizada pelo sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95] a qual foi treinada unicamente para 12 tipos de fontes diferentes, tal como referido na Secção 3.3.1. Deste modo, o treino da base de dados com poucos tipos de fontes penaliza muito este sistema OCR, sobretudo no reconhecimento de texto de cena, onde o texto apresenta uma maior diversidade de fontes, estilos e tamanhos.

De forma a avaliar o desempenho do algoritmo desenvolvido de forma mais objectiva, foram efectuados testes que comparam o desempenho do OCR OmniPage Pro 12.0 isolado e em conjunto com o algoritmo de detecção de texto proposto neste capítulo. Para isso, foi efectuado o reconhecimento de todo o texto que faz parte da *ground truth* das 60 imagens, utilizando unicamente o OCR OmniPage Pro 12.0 e utilizando o algoritmo de detecção de texto proposto em conjunto com o OCR OmniPage Pro 12.0. Os resultados obtidos podem ser observados na Tabela 3.7.

Tabela 3.7 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da *ground truth*, utilizando unicamente o OCR OmniPage Pro 12.0 e utilizando o algoritmo de detecção de texto em conjunto com o OCR OmniPage Pro 12.0.

Tipos de texto	OCR OmniPage Pro 12.0 apenas		Algoritmo de Detecção de Texto + OCR OmniPage Pro 12.0	
	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão
Texto de cena	0.444	0.796	0.698	0.815
Texto gráfico	0.409	0.821	0.902	0.919
Totalidade do texto na <i>ground truth</i>	0.400	0.810	0.826	0.884

A utilização conjunta do algoritmo de detecção de texto proposto e do OCR OmniPage Pro 12.0 permite realizar uma melhor separação entre o texto e o fundo complexo da imagem. Como se pode ver na Tabela 3.7, esta separação permite um aumento muito significativo da *recall* do sistema OCR de 44.4% para 69.8% no texto de cena e de 40.9% para 90.2% no texto gráfico. Em termos globais, a *recall* do OmniPage Pro 12.0 aumenta de 40% para 82.6%. Em termos de precisão, o aumento de desempenho é menos significativo; todavia, há ainda assim um aumento global de 81% para 88.4%.

O aumento da *recall* quando são combinados o OCR e o algoritmo de detecção de texto proposto deve-se, essencialmente, ao facto do OCR utilizado isoladamente quando as imagens possuem fundos complexos e o texto é mal diferenciado em relação ao fundo,

classificar as regiões de texto como gráficos ou imagens, não efectuando o reconhecimento do mesmo; esta situação é ilustrada na Figura 3.32. Este procedimento por parte do OCR contribui para o aumento do número de caracteres não reconhecidos o que pela definição de *recall* (relação entre o número de caracteres de texto correctamente reconhecidos e o número de caracteres de texto existente na imagem) vai fazer com que o valor desta tenda a ser baixo, 40%. Quando combinado o OCR com o algoritmo de detecção de texto, o último elimina os fundos complexos da imagem, fornecendo ao OCR uma imagem binária com o fundo a preto e o texto a branco. Tal, permite ao OCR reconhecer todo o texto existente nas imagens ainda que estas possuam fundos com texturas complexas, o que faz com que o valor da *recall* aumente muito, de 40% para 82.6%.

Em termos de precisão (relação entre o número de caracteres de texto correctamente reconhecidos e o número total de caracteres de texto reconhecidos) esta também aumenta, ainda que menos, quando são combinados o OCR e o algoritmo de detecção de texto proposto. Tal, deve-se ao facto do OCR mesmo quando utilizado isoladamente possuir uma precisão elevada, 81%, ainda que esta seja à custa da diminuição da *recall*, i.e. à custa do aumento da percentagem de caracteres não reconhecidos. Quando combinado o OCR com o algoritmo de detecção de texto, como referido anteriormente, o último elimina os fundos complexos da imagem, permitindo ao OCR diminuir o número, já pequeno, de falsos reconhecimentos, o que faz com que o valor da precisão aumente ainda que ligeiramente, de 81% para 88.4%.

A aplicação directa do OmniPage Pro 12.0 foi testada exaustivamente com vários tipos de imagens, tendo-se verificado que este sistema OCR apresenta dificuldades no reconhecimento de texto em imagens com fundos muito complexos, como é o caso da imagem ilustrada na Figura 3.32



Figura 3.32 – Exemplo da dificuldade evidenciada pelo OCR OmniPage Pro 12.0 em reconhecer (directamente) texto em imagens com fundos complexos: (a) imagem original e (b) resultado do reconhecimento de texto efectuado pelo OCR.

Para além disso, o OmniPage Pro 12.0 também não reconhece com precisão o texto inclinado, ainda que este se encontre escrito sobre um fundo uniforme (sendo neste caso essencial o processamento incluído no algoritmo de detecção proposto neste capítulo). Na Figura 3.33 são ilustrados exemplos da influência do algoritmo de detecção de texto proposto

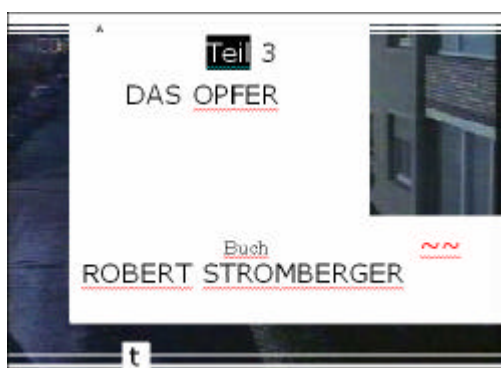
no reconhecimento de texto efectuado pelo OCR OmniPage Pro 12.0. Nas Figura 3.33 (a), (c), (e) e (g) ilustra-se um caso onde o texto está escrito na horizontal e é bem contrastado em relação ao fundo da imagem, o que favorece o reconhecimento por parte do OCR; nas Figura 3.33 (b), (d), (f) e (h) ilustra-se a detecção de texto inclinado.



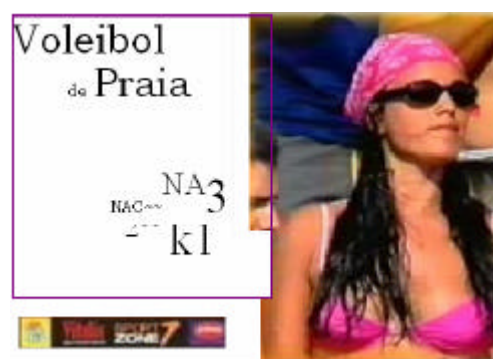
(a)



(b)



(c)



(d)



(e)



(f)

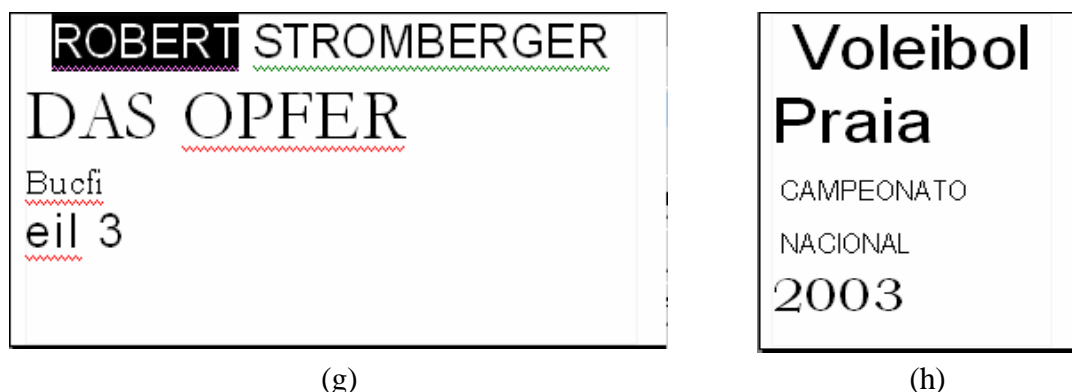


Figura 3.33 – Exemplo da influência do algoritmo de detecção de texto proposto, no reconhecimento de texto efectuado pelo OCR OmniPage Pro 12.0: (a) e (b) imagens originais; (c) e (d) resultados do reconhecimento de texto efectuado pelo OCR; (e) e (f), imagens fornecidas ao OCR pelo algoritmo de detecção de texto; (g) e (h) resultados do reconhecimento de texto efectuado pelo OCR em conjunto com o algoritmo de detecção de texto.

3.5 Comentários Finais

Ao longo deste capítulo foi proposto um algoritmo que permite a detecção, tanto de texto gráfico, como de texto de cena, em imagens ou tramas de vídeo. O texto pode ser constituído por caracteres de vários tamanhos, fontes e cores e estar escrito em qualquer direcção. O método proposto começa por efectuar a segmentação das imagens em regiões conexas que são, posteriormente, filtradas de acordo com várias restrições. As restrições impostas actuam ao nível do contraste, da forma e da localização espacial e o seu objectivo prende-se com a eliminação de regiões que não correspondem a texto.

As maiores contribuições deste algoritmo foram ao nível da detecção de palavras ou linhas de texto quando este é inclinado, i.e. do agrupamento das regiões classificadas como texto de modo a formarem palavras com um reduzido número de falsas detecções, bem como do aperfeiçoamento e adaptação para texto de cena (menos estruturado e com menor contraste em relação ao fundo) de técnicas já conhecidas mas que foram desenvolvidas tendo em vista o texto gráfico (mais estruturado e mais contrastado em relação ao fundo), nomeadamente técnicas de melhoramento de fronteiras e análise do contraste.

Após a sua implementação, o algoritmo proposto foi testado utilizando vários tipos de imagens retiradas de genéricos de filmes, noticiários, anúncios comerciais e eventos desportivos. Foram efectuados testes só com texto horizontal e com texto orientado em qualquer direcção. Os resultados foram analisados, tendo-se verificado na detecção do texto horizontal um melhor desempenho, ainda que ligeiro, em relação ao texto genérico, tanto para a *recall* como para a precisão. No reconhecimento do texto, tal como na detecção, também se verificou para o texto horizontal um melhor desempenho tanto para a *recall* como para a precisão.

O texto de cena tem, como característica principal, uma grande diversidade de fontes, estilos, tamanhos e orientações. Essas características tornam-no mais difícil de detectar pelo

algoritmo proposto e de reconhecer por parte dos sistemas OCR utilizados devido à sua complexidade estrutural. Desta forma, os resultados obtidos pelo algoritmo proposto para texto de cena, tornam-se menos bons quando comparados com os resultados para texto gráfico. Pode contudo concluir-se que o algoritmo proposto apresenta resultados muito satisfatórios em ambas as métricas (*recall* e precisão), tanto para texto gráfico como para texto de cena quando comparado com os sistemas que foram descritos no capítulo 2.

Pode pois considerar-se que o algoritmo proposto é bastante robusto pois detecta 87% do texto existente nas imagens, 90% do qual é correctamente detectado; utilizando o sistema OCR OmniPage Pro 12.0, é reconhecido 83% do texto existente nas imagens, sendo que 88% é correctamente reconhecido independentemente de este ser de cena ou gráfico e de estar escrito na horizontal ou ser inclinado.

Capítulo 4

Extracção de Texto em Sequências de Vídeo

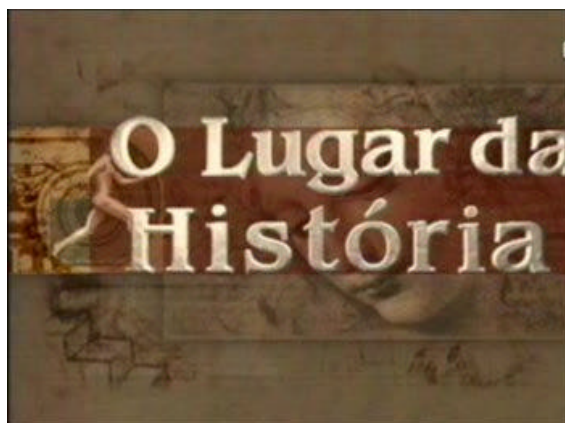
O principal objectivo deste capítulo é a apresentação do algoritmo desenvolvido para efectuar a detecção de texto em sequências de vídeo bem como o estudo do seu desempenho. Este tipo de detecção difere da detecção de texto em imagens essencialmente porque as sequências de vídeo possuem redundância temporal, i.e. cada linha de texto surge em várias tramas temporalmente contíguas, facto que não se verifica nas imagens isoladas. A redundância temporal, naturalmente existente no vídeo, pode ser explorada para:

1. Aumentar a probabilidade da detecção de texto, desde que este se repita dentro de determinadas condições em tramas sucessivas;
2. Remover falsas detecções em tramas individuais, se as detecções não se mantiverem consistentes ao longo do tempo;
3. Fazer a interpolação de linhas de texto que não foram ‘acidentalmente’ detectadas em algumas tramas individuais.

Deste modo, a detecção de texto em sequências de vídeo, para além das técnicas utilizadas na detecção de texto em imagens, e que foram descritas no capítulo 3 da presente Tese, explora também a redundância temporal existente no vídeo. À semelhança do que acontece com a detecção de texto em imagens, o texto detectado nas sequências de vídeo através do processo desenvolvido neste capítulo pode ter várias aplicações. Por exemplo, o texto poderá servir para acrescentar uma componente semântica à descrição do vídeo correspondente, eventualmente usando os descritores adequados da norma MPEG-7 [Manjunath02]. Pode, ainda, ser utilizado para navegação automática ou vigilância, classificação de vídeo, análise de eventos (por exemplo, eventos desportivos), ou ainda, para fazer a codificação eficiente do texto como um objecto textual independente, por exemplo usando a norma MPEG-4 [Pereira02]. Na Figura 4.1 são ilustrados exemplos de imagens para as quais se pode justificar a extracção de texto, nomeadamente para utilizar o texto extraído na classificação e indexação do vídeo em questão.



(a)



(b)

Figura 4.1 – Exemplos de imagens para as quais se pode justificar a extracção de texto para: (a) analisar o evento desportivo; (b) classificar o programa em questão.

4.1 Arquitectura Básica

Para que possa ser feita a descrição detalhada do método desenvolvido para a extracção de texto em sequências de vídeo, é fundamental que se comece por apresentar a sua arquitectura básica, ou seja, a sequência de processos aplicados às várias tramas de vídeo através dos quais se extrai o texto nelas contido.

De forma análoga ao que foi feito para a detecção de texto em imagens, também para a detecção de texto em sequências de vídeo, algumas das alternativas técnicas mais importantes existentes na literatura para os vários módulos da arquitectura básica foram já objecto de estudo no Capítulo 2 da presente Tese. Nesse estudo, foram analisadas as suas vantagens e desvantagens bem como as características do texto por elas detectado. Em consequência, foi adoptada como arquitectura básica para o sistema de extracção de texto em sequências de vídeo aqui proposto, aquela que se apresenta na Figura 4.2.

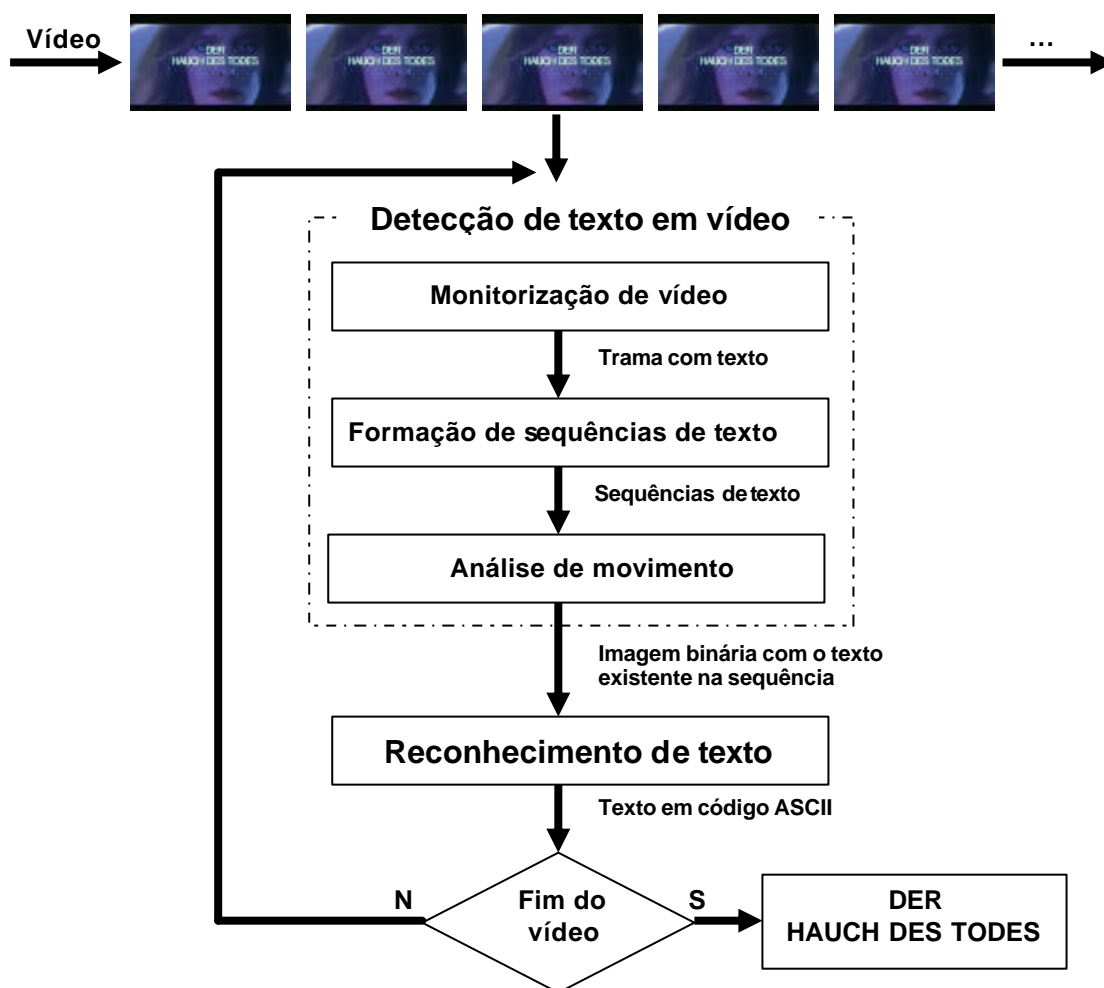


Figura 4.2 – Arquitectura básica do algoritmo de extracção de texto em sequências de vídeo.

Como se pode observar na Figura 4.2, a extracção de texto em sequências de vídeo decorre em duas fases principais bem distintas: a primeira fase visa a detecção do texto, enquanto que a segunda fase visa o seu reconhecimento:

1ª Fase – Detecção de texto em vídeo

Esta fase visa a detecção do texto existente nas tramas de vídeo e pode ser dividida em três etapas. Na 1ª etapa – **monitorização de vídeo** – é verificada a existência, ou não, de texto no vídeo através de uma análise temporal grosseira do mesmo ou seja apenas algumas tramas são periodicamente analisadas; na 2ª etapa – **formação de sequências de texto** – são efectuadas tanto a localização temporal do início e do fim das sequências de texto existentes no vídeo bem como a sua detecção; na 3ª etapa – **análise de movimento** – refina-se a detecção de texto e ajusta-se o início e o fim da sua ocorrência, através da exploração da redundância temporal existente no vídeo:

1ª Etapa – Monitorização de vídeo – Nesta etapa pretende-se efectuar a monitorização das tramas do vídeo em termos de conteúdo textual visando detectar a existência, ou não, de texto em cada trama analisada tendo como objectivo a localização temporal aproximada do início e fim das sequências de texto existentes no vídeo. Para tal, é efectuada a

detecção de texto em tramas de vídeo periodicamente espaçadas no tempo, com vista a diminuir o impacto computacional deste processo de monitorização;

2ª Etapa – Formação de sequências de texto – Nesta etapa pretendem atingir-se dois objectivos:

- a. Efectuar a localização temporal mais rigorosa do início e do fim das sequências de texto existentes no vídeo e que foram (grosseiramente) localizadas na etapa anterior;
- b. Efectuar a detecção do texto existente no intervalo de tramas anteriormente determinado.

Deste modo, logo que é detectado texto numa trama pela fase de monitorização, a detecção de texto passa a ser feita de forma menos espaçada no tempo, para a frente e para trás de forma a detectar o início e o fim da sequência de texto bem como o texto existente em cada trama;

3ª Etapa – Análise de movimento – Nesta etapa pretende-se explorar a redundância temporal existente no vídeo com o intuito de melhorar o desempenho da detecção de texto, nomeadamente a precisão do início e do fim da ocorrência de cada palavra ao longo do vídeo, bem como a sua localização precisa em cada trama.

2ª Fase – Reconhecimento de texto

Esta fase visa o reconhecimento do texto detectado nas tramas do vídeo. Para tal, são usadas as regiões candidatas a texto resultantes da detecção de texto efectuada na fase anterior, utilizando-se para o reconhecimento, tal como na extracção de texto em imagens apresentada no capítulo 3, dois sistemas OCR: um desenvolvido para o caso específico da extracção de texto em imagens ou vídeo [Lienhart95] e outro correspondente a uma versão comercial do OmniPage Pro 12.0 [ScanSoft].

O processamento proposto nesta Tese para as duas fases da extracção de texto em sequências de vídeo será apresentado em pormenor nas secções seguintes.

4.2 Detecção de Texto em Vídeo

A detecção do texto existente nas sequências de vídeo tem como objectivo identificar um conjunto de regiões conexas classificadas como candidatas a texto e que, para além disso, cumpram os critérios para a formação de palavras descritos na Secção 4.2.3.2. A detecção do texto é efectuada em três etapas de forma a reduzir a complexidade computacional: monitorização do vídeo, formação de sequências de texto e análise de movimento do mesmo. Abordagens semelhantes foram utilizadas por outros investigadores em [Sato99, Li00, Li02, Lienhart02, Wolf02]; contudo, estes sistemas foram concebidos para extrair texto gráfico escrito na horizontal, excepção feita para o sistema apresentado em [Li02] que foi concebido para extrair tanto texto gráfico como texto de cena escrito em qualquer direcção (o autor só apresenta resultados para texto gráfico escrito na horizontal). Com o sistema proposto nesta Tese, pretende alcançar-se uma solução robusta para efectuar a extracção tanto de texto de cena, como de texto gráfico, escrito em qualquer direcção, com um reduzido número de falsas detecções, nomeadamente para o texto inclinado.

A forma como as três fases da detecção de texto se encadeiam é ilustrada na Figura 4.3. Na primeira fase, o vídeo é monitorizado, ou seja, analisado com uma resolução temporal grosseira, por exemplo uma trama em cada 25, para detectar a existência de texto. Na segunda fase, o texto localizado na fase anterior é detectado e delimitado, i.e. determina-se o início e o fim da sua ocorrência temporal. Na terceira fase, efectua-se o seguimento do texto existente no vídeo e que foi detectado na fase anterior, com o objectivo de explorar a redundância temporal existente no vídeo de modo a melhorar o desempenho da detecção de texto.

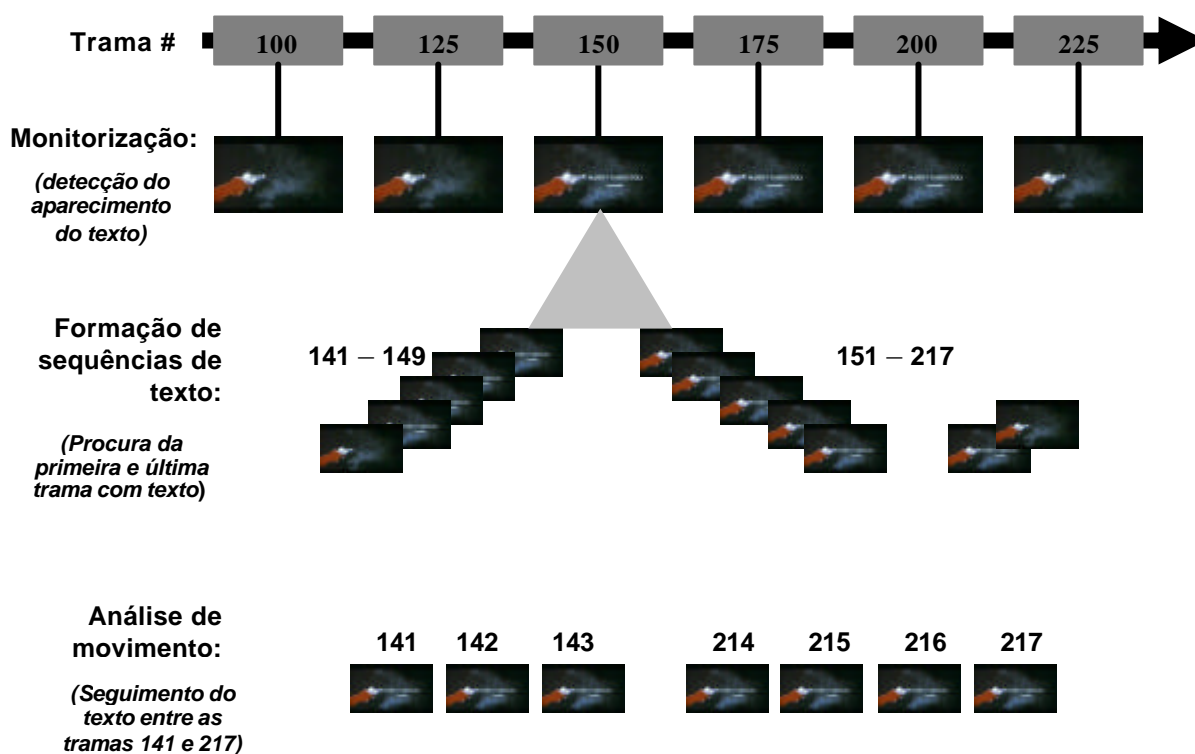


Figura 4.3 – Relação entre a monitorização do vídeo (1ª fase), a formação de sequências de texto (2ª fase) e a análise do movimento (3ª fase).

A arquitectura global do processo de detecção de texto em vídeo, ou seja, das fases de monitorização de texto, formação de sequências de texto e de análise de movimento é apresentada na Figura 4.4. O processo de detecção de texto inicia-se com a procura de texto no vídeo. Para tal, é feita a monitorização do vídeo de uma forma grosseira, de 25 em 25 tramas. Logo que é detectado texto numa trama, inicia-se a fase de formação de sequências de texto. Nesta fase, a detecção de texto passa a ser feita trama a trama, para a frente e para trás de modo a identificar com precisão o início e o fim de cada sequência de texto, bem como detectar o texto existente em cada trama. Esta fase termina quando for detectada uma sequência de três tramas sem texto, quer para a frente, quer para trás. Quando o intervalo de ocorrência do texto tiver sido determinado, começa a análise de movimento do texto no intervalo de ocorrência do mesmo. Terminada a fase de seguimento, retorna-se novamente à fase de monitorização grosseira do vídeo. Este processo de monitorização e seguimento repete-se até se atingir o fim da sequência de vídeo.

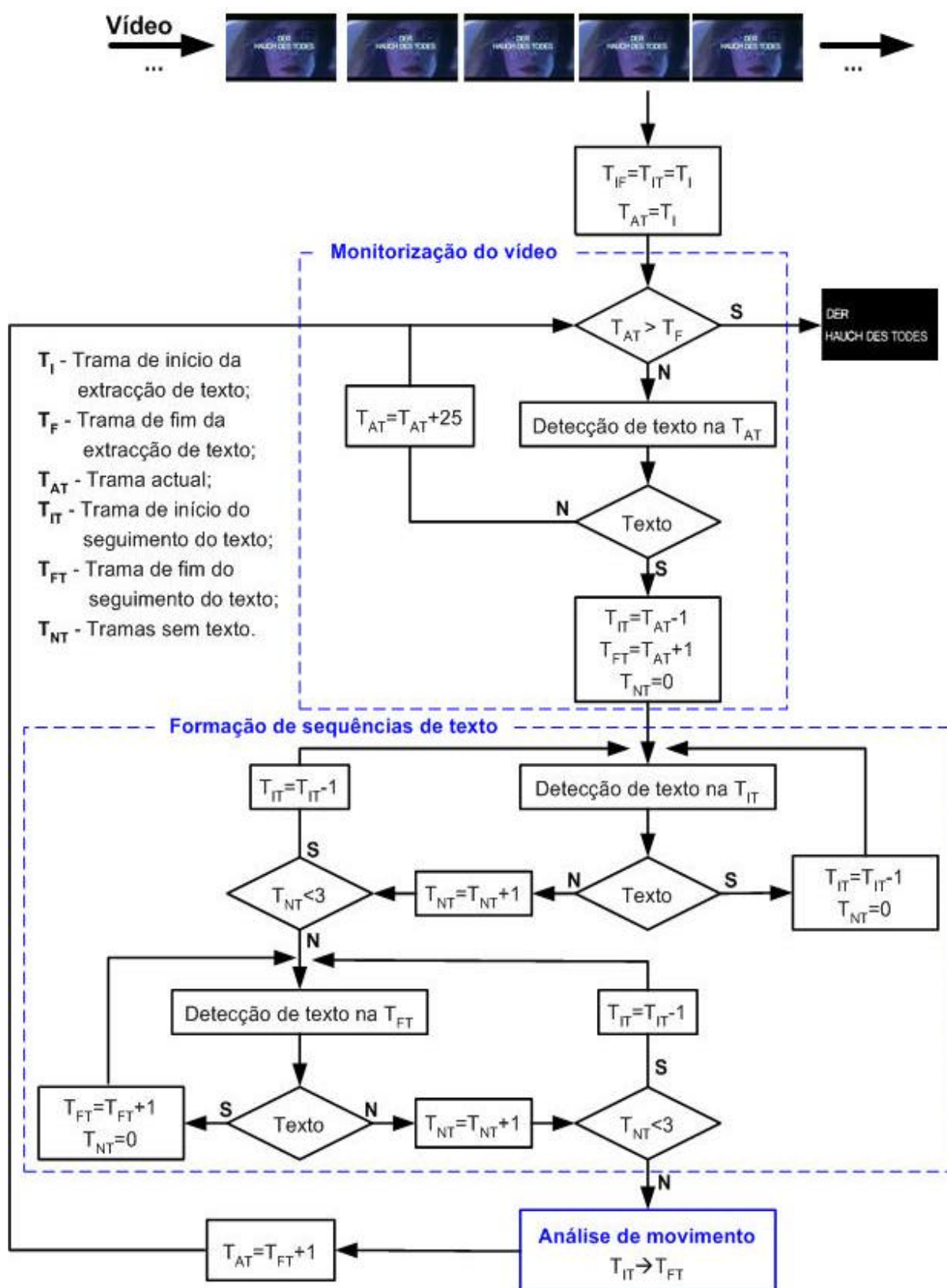


Figura 4.4 – Arquitectura do processo de detecção de texto em vídeo.

Como resultado do processo anteriormente descrito, obtêm-se uma imagem binária para cada sequência de texto onde o texto detectado é representado a branco e o fundo a preto. Nesta imagem é integrado todo o texto existente na sequência de texto. Terminada a fase de detecção de texto na sequência de vídeo, as imagens binárias com o texto detectado são processadas por um sistema OCR que faz o seu reconhecimento.

Os módulos de monitorização, formação de sequências de texto e análise de movimento do algoritmo de detecção do texto em sequências de vídeo serão discutidos nas secções seguintes.

4.2.1 Monitorização do Texto

A monitorização do texto tem como objectivo detectar a existência de sequências de texto no vídeo, ainda que com uma baixa precisão em termos de localização temporal. Deste modo, a monitorização do texto é efectuada com uma resolução temporal grosseira de forma a reduzir a complexidade computacional associada à extracção do texto. Assim, o algoritmo proposto no capítulo anterior para a detecção de texto em imagens, é aplicado às tramas de vídeo de uma forma periódica e espaçada no tempo para detectar a presença de texto nas mesmas. A periodicidade de monitorização é definida de tal modo que não haja perda de qualquer linha de texto, ou seja, o intervalo máximo para a monitorização é definido em função do tempo mínimo assumido para a duração da ocorrência de uma linha de texto. É sabido, através das investigações efectuadas sobre a visão humana, que são necessários 2 a 3 segundos para um ser humano efectuar o processamento de uma imagem complexa [Lindsay91, Lienhart02]. Portanto, pode-se assumir sem grandes riscos que, para que uma linha de texto seja humanamente perceptível, esta deve estar presente no vídeo pelo menos durante cerca de um segundo. Para vídeo a 25 tramas por segundo, o intervalo de monitorização ou seja entre detecções será então de 25 tramas.

Assim, a monitorização do texto inicia-se com a detecção do mesmo de 25 em 25 tramas. Para efectuar a detecção do texto nas tramas, recorreu-se a uma versão simplificada do algoritmo descrito no Capítulo 3 (Secção 3.2), desenvolvido para detectar texto em imagens. Tal como na detecção de texto em imagens, também aqui o objectivo da detecção de texto é a formação de um conjunto de regiões conexas, classificadas como candidatas a texto, e que, para além disso, cumpram os critérios para a formação de palavras. A detecção decorre igualmente em quatro fases distintas, conforme se apresenta na Figura 4.5: simplificação da imagem, segmentação da imagem em regiões conexas, detecção de caracteres e formação de palavras. Considera-se que existe texto numa trama analisada, sempre que exista pelo menos uma palavra válida. Esta pode estar escrita em qualquer direcção.

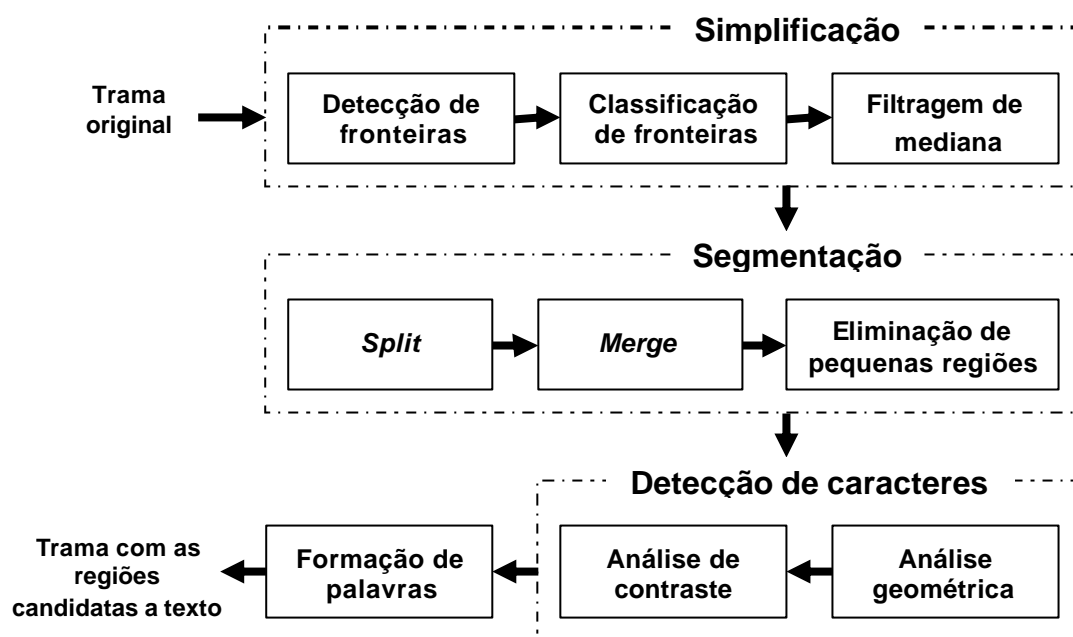


Figura 4.5 – Arquitectura do processo de detecção de texto para cada trama de vídeo analisada.

Os módulos de simplificação, segmentação e detecção de caracteres são em tudo idênticos aos utilizados na detecção de texto em imagens descrita na capítulo anterior. As diferenças entre os dois algoritmos de detecção de texto em imagens são, portanto, ao nível da formação de palavras. Assim, de seguida será apenas efectuada a descrição da formação de palavras.

Deste modo, a formação de palavras, no âmbito da monitorização do texto, visa unicamente verificar a existência de texto nas tramas de vídeo. Para tal, procede-se ao agrupamento das regiões provenientes das fases anteriores e que foram classificadas como texto, de modo a formar palavras [Fletcher88, Zhong95, Messelodi99, Lienhart00]. O algoritmo utilizado na formação de palavras em imagens, apresentado no capítulo anterior, difere do algoritmo de formação de palavras em vídeo, uma vez que o primeiro considera três fases distintas, agrupamento de regiões, eliminação de palavras sobrepostas e rotação do texto, enquanto que o último apenas considera a fase do agrupamento de regiões, que é em tudo idêntica à utilizada pelo algoritmo de formação de palavras em imagens. Esta diferença justifica-se uma vez que no vídeo se pretende apenas verificar a existência, ou não, de texto na trama. A detecção de palavras propriamente dita é efectuada, mais tarde, na análise de movimento e depois de efectuado o seguimento do texto.

Assim, ao ser detectado texto numa trama, inicia-se a fase de formação de sequências de texto.

4.2.2 Formação de Sequências de Texto

Antes de iniciar-se a descrição do processo de formação de sequências de texto, importa definir o conceito de sequência de texto utilizado nesta Tese. Assim, uma sequência de texto

consiste num conjunto de palavras ou mesmo linhas de texto que existem em tramas contíguas e que está delimitado, no início e no fim, por uma sequência de pelo menos três tramas contíguas sem texto. Uma sequência de texto pode ser formada, tanto por uma única palavra, como por um conjunto de palavras ou mesmo de linhas de texto com durações diferentes e com inícios e fins em tramas diferentes. Na Figura 4.6 são apresentados dois exemplos de sequências de texto. A primeira sequência de texto é formada por duas linhas de texto de igual duração com início na trama 4 e fim na trama 6, delimitada por duas sequências de três tramas sem texto, uma no início e outra no fim, tramas 1 a 3 e 7 a 9, respectivamente. A segunda é formada por oito linhas de texto, com diferentes duração, início e fim: esta sequência de texto inicia-se na trama 10 e termina na trama 20, delimitada igualmente por duas sequências de pelo menos três tramas sem texto, 7 a 9 no início e 21 a 23 no fim.



Figura 4.6 – Exemplos de sequências de texto.

Logo que for detectado texto numa trama pela fase de monitorização de texto, a detecção de texto passa a ser feita em todas as tramas intermédias, primeiro para trás e depois para a frente, com o objectivo de determinar tanto o início e o fim da ocorrência de texto, como detectar o texto existente em cada trama. A detecção tanto para trás como para a frente, termina quando forem encontradas para ambas as direcções sequências de três tramas contíguas sem texto. O início e o fim da ocorrência de texto definem o intervalo sobre o qual é efectuada a análise de movimento e as regiões classificadas como caracteres resultantes da detecção de texto efectuada sobre cada trama serão a informação sobre a qual é aplicada a análise de movimento. Para efectuar a detecção do texto existente nas tramas intermédias, é utilizado um algoritmo em tudo idêntico ao utilizado na fase de monitorização do texto, ver Figura 4.5. Nesta fase, apenas é delimitada a sequência de texto e não as linhas de texto de que esta é formada. A delimitação temporal das linhas de texto dentro de cada sequência de texto é efectuada na fase de seguimento de texto.

4.2.3 Análise de Movimento

A análise de movimento tem como principal objectivo explorar a redundância temporal existente no vídeo com o intuito de melhorar o desempenho da detecção de texto. Para além do refinamento da detecção de texto propriamente dita, determinar-se-á também com precisão o início e o fim de cada palavra ou linha de texto, bem como a sua localização em cada trama. Assim, através da análise de movimento, é possível identificar falsas detecções em tramas individuais, por exemplo caso não seja possível fazer o seguimento desse texto; essas falsas detecções serão então classificadas como não texto e consequentemente eliminadas. Para além disso, a análise de movimento possibilita a identificação de falhas na detecção de regiões e a sua recuperação através de interpolação a partir das tramas vizinhas. A análise de movimento permite ainda integrar todo o texto numa única imagem binária, independentemente da sua ocorrência temporal, facilitando deste modo a tarefa do reconhecimento do mesmo. Na Figura 4.7 apresenta-se um exemplo do tipo de informação extraída com a análise de movimento. A Figura 4.7 (a) corresponde a um conjunto de tramas com texto extraídas de uma sequência de vídeo; na Figura 4.7 (b) ilustra-se o resultado da extracção de texto para cada trama e na Figura 4.7 (c) ilustra-se o resultado da integração de todo o texto existente no vídeo numa única imagem final.



Figura 4.7 – Exemplo dos resultados obtidos com a análise do movimento: (a) tramas da sequência de vídeo com texto; (b) imagens com a detecção do texto para cada trama individual; e (c) imagem final resultante da integração de todo o texto existente na sequência.

No Capítulo 2 foram apresentados vários métodos para efectuar o seguimento do texto, todos eles com uma característica comum ou seja fazer o seguimento do texto com base na comparação de tramas sucessivas, i.e. relacionam o resultado do momento anterior com o do momento actual. Os vários métodos variam apenas na forma como estabelecem uma relação entre o momento anterior e o momento actual; por exemplo em [Lienhart02], o seguimento é feito ao nível da palavra, mas em [Li02] é feito ao nível do bloco de texto que pode conter várias palavras ou mesmo linhas de texto.

Na presente Tese, o método proposto para efectuar o seguimento do texto também se baseia na comparação de tramas sucessivas mas optou-se por efectuar o seguimento do texto ao nível do carácter, quando o mais natural seria efectuar o seguimento ao nível da palavra, uma vez que foi efectuada a sua detecção durante a fase de formação de sequências de texto. Esta opção tem fundamentalmente a ver com as vantagens apresentadas pelo seguimento do texto ao nível do carácter, sobretudo na recuperação de texto parcialmente detectado e no seguimento de pequenas palavras. As vantagens do seguimento ao nível do carácter são ilustradas na Figura 4.8. Considere-se, por exemplo, uma palavra formada pelos caracteres *A*, *B* e *C*. Os quadrados azuis e verdes representam detecções correctas dos caracteres e da palavra, respectivamente, (note-se que, na detecção de palavras, conjuntos de caracteres com uma dimensão inferior a 3 são eliminados) e os quadrados vermelhos representam falhas de detecção dos caracteres e das palavras nas várias tramas. Se o seguimento for efectuado ao nível da palavra, no caso ilustrado na Figura 4.8 é impossível efectuar o seguimento da palavra pois esta só é detectada duas vezes em sete tramas; fazendo o seguimento ao nível do carácter, é fácil seguir os três caracteres, recuperar as suas omissões através de interpolação e posteriormente agrupar os caracteres de modo a formar uma palavra.

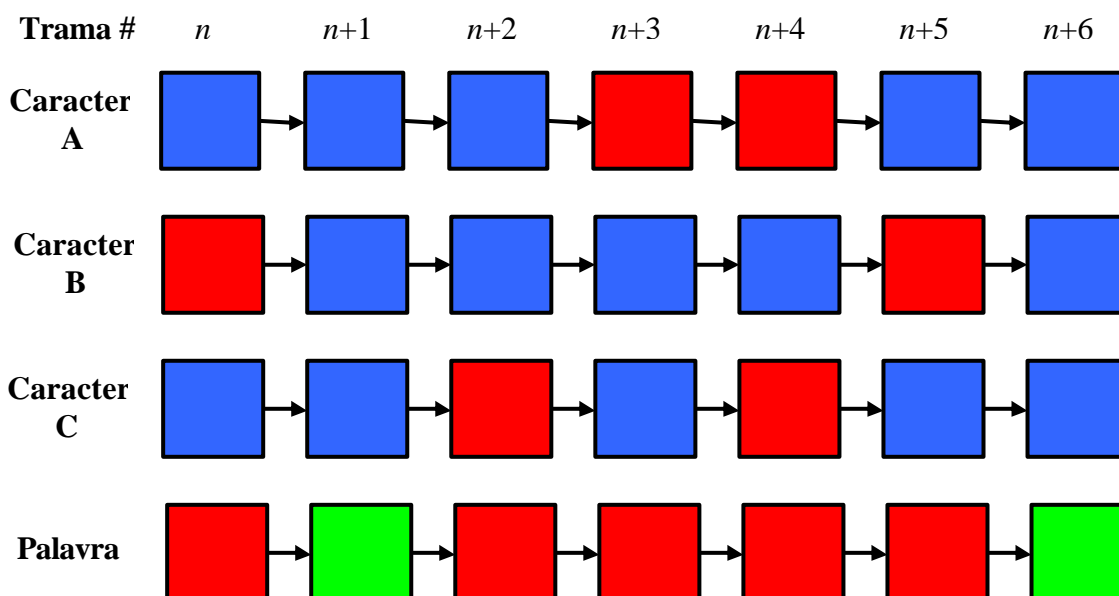


Figura 4.8 – Exemplo das vantagens do seguimento do texto ao nível do carácter versus seguimento ao nível da palavra.

A arquitectura do processo de análise de movimento no vídeo, ou seja, o seguimento e integração do texto é apresentada na Figura 4.9. A análise do movimento decorre, portanto, em três fases principais bem distintas. A primeira fase visa o seguimento do texto de modo a

formar as cadeias de caracteres, enquanto a segunda fase visa a integração do texto existente nas várias tramas, de modo a formar palavras ou mesmo linhas de texto. A terceira e última fase visa a apresentação do resultado da detecção do texto em vários formatos em função do tipo de utilização a dar ao texto detectado.

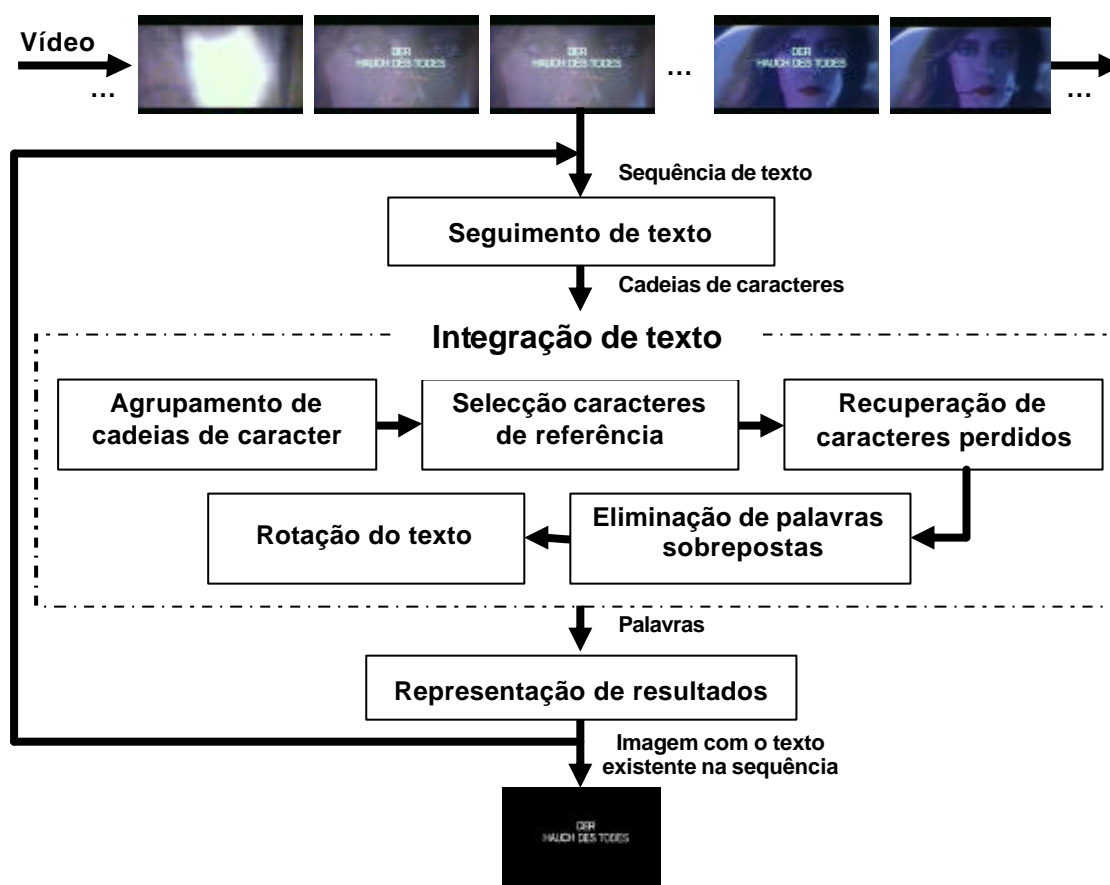


Figura 4.9 – Arquitectura do processo de análise de movimento em sequências de vídeo.

As três fases da análise do movimento serão apresentadas em pormenor nas secções seguintes.

4.2.3.1 Seguimento de Texto

A fase de seguimento do texto visa o seguimento, para cada sequência de texto detectada e ao longo das várias tramas que a compõem, das regiões conexas classificadas como caracteres durante a fase de formação de sequências de texto com vista a formar cadeias de caracteres. Uma cadeia de caracter representa um caracter ao longo da sua presença no vídeo e consiste numa colecção de regiões de algum modo semelhantes, classificadas como caracteres na fase de formação de sequências de texto (nesta fase é feita a detecção do texto) e que se repetem ao longo de várias tramas contíguas. Sempre que um caracter é detectado pela primeira vez, é definida uma assinatura para esse caracter com as seguintes características: luminância, dimensão, posição e forma. Cada trama contribui com uma e uma só região classificada como caracter para a formação de uma cadeia de caracter. Para formar as cadeias de caracteres, há

que estabelecer os critérios que permitem definir, com exactidão, o modo como estas vão ser constituídas. Desta forma, os critérios de semelhança para a formação de cadeias de caracteres baseiam-se nas características da assinatura e são os seguintes:

- **Luminância** – Para que um dado carácter C_i faça parte de uma cadeia de carácter, CC_j , deve possuir uma luminância semelhante à da cadeia de carácter, mais precisamente semelhante ao valor médio da luminância dos caracteres que já fazem parte da cadeia de carácter. Para que isso se verifique, é definido um valor de limiar para a variação da luminância, Th_{lum} . Assim, o carácter C_i faz parte da cadeia de carácter CC_j se:

$$\left| Y_{(C_i)} - Y_{(CC_j)} \right| < Th_{lum} \quad (4.1)$$

Onde $||$ representa o valor absoluto da diferença entre a luminância do carácter C_i e a luminância média dos caracteres que formam a cadeia de carácter CC_j . Nesta Tese, usa-se $Th_{lum} = 25$, valor que foi obtido empiricamente através de testes exaustivos;

- **Dimensão** – Para que um dado carácter C_i faça parte de uma cadeia de carácter, CC_j , a sua dimensão (número de *pixels*) deve ser aproximadamente a mesma que a dimensão média dos caracteres que formam essa cadeia. Para que isso se verifique, é definido um valor de limiar para a variação da dimensão, Th_{dim} . Assim, o carácter C_i faz parte da cadeia de carácter CC_j se:

$$\left| A_{C_i} - A_{CC_j} \right| < Th_{dim} \quad (4.2)$$

Onde $||$ representa o valor absoluto da diferença entre a área do carácter C_i e a área média da cadeia de carácter CC_j , medidas em *pixels* com Th_{dim} dado pela seguinte expressão:

$$Th_{dim} = A_{CC_j} \times 0.25 \quad (4.3)$$

ou seja, Th_{dim} é igual a 25% da área média (em *pixels*) dos caracteres que já fazem parte da cadeia de carácter CC_j ;

- **Posição** – Para que uma dado carácter C_i faça parte de uma cadeia de carácter CC_j , este deve possuir uma localização semelhante à localização estimada para a cadeia de carácter nessa trama, i.e. a distância, d , entre a posição do carácter C_i da trama T_k e a posição estimada para a cadeia de carácter CC_j na trama T_k deve ser inferior a um dado limiar Th_{pos} . A posição de uma cadeia de carácter pode variar ao longo do tempo, ou seja, pode assumir um valor diferente para cada trama onde ela existe e é dada em cada trama pela posição do carácter que a representa nessa trama ou seja mais precisamente pela posição (x_c, y_c) correspondente ao centro da *bounding box* do carácter, C , na matriz $T(x,y)$.

Desta forma, o carácter C_i de coordenadas (x_{C_i}, y_{C_i}) pertencente à trama T_k faz parte da cadeia de carácter CC_j de coordenadas estimadas (x_{CC_j}, y_{CC_j}) na trama T_k , se $d < Th_{pos}$, onde a distância, d , é dada pela expressão (4.4):

$$d = \sqrt{(x_{C_i} - x_{CC_j})^2 + (y_{C_i} - y_{CC_j})^2} \quad (4.4)$$

e a posição estimada $(x_{CC_{j_k}}, y_{CC_{j_k}})$ para a cadeia de carácter CC_j , na trama T_k , é dada por:

$$(x_{CC_{j_k}}, y_{CC_{j_k}}) = (x_{CC_{j_{k-1}}}, y_{CC_{j_{k-1}}}) + (dx, dy) \quad (4.5)$$

Onde (dx, dy) é o valor médio do deslocamento da cadeia de carácter CC_j , entre duas tramas, dado pela expressão (4.6):

$$(dx, dy) = \left(\frac{x_{CC_{j_1}} - x_{CC_{j_{k-1}}}}{n}, \frac{y_{CC_{j_1}} - y_{CC_{j_{k-1}}}}{n} \right) \quad (4.6)$$

onde n representa a duração da cadeia de carácter CC_j , i.e. o número de tramas onde a cadeia de caracteres existe e $x_{CC_{j_1}}, y_{CC_{j_1}}, x_{CC_{j_{k-1}}}, y_{CC_{j_{k-1}}}$ representam as coordenadas da cadeia de carácter, CC_j , na trama onde ela teve origem e na trama T_k , respectivamente. Nesta Tese, usa-se $Th_{pos} = 3$ (*pixels*), valor que foi obtido empiricamente através de testes exaustivos.

Este critério só é aplicado a partir da terceira região adicionada à cadeia de carácter, por se considerar que só a partir dessa altura é que o valor médio do deslocamento da cadeia de carácter possui a precisão suficiente para estimar uma posição para a cadeia com fiabilidade;

- **Forma** – Para que um dado carácter C_i faça parte de uma cadeia de carácter, CC_j , este deve possuir uma forma semelhante à da cadeia de carácter, i.e. a distribuição espacial do carácter C_i , dentro da sua *bounding box* deve ser semelhante à distribuição espacial da cadeia de carácter CC_j , dentro da sua *bounding box*. De modo a verificar se a distribuição espacial do carácter C_i é semelhante à da cadeia de carácter CC_j , as *bounding boxes* de ambos são divididas em quatro sectores idênticos através de duas linhas, uma horizontal e outra vertical, ver Figura 4.10.



Figura 4.10 – Exemplo da divisão da *bounding box* de um carácter em quatro sectores.

Assim, para que um dado carácter, C_i , faça parte de uma cadeia de carácter, CC_j , a sua dimensão (número de *pixels*) em cada sector deve ser aproximadamente a mesma que a dimensão média dos sectores que formam a cadeia de carácter. Para que isso se verifique, são definidos quatro valores de limiar, Th_{s_i} , um para cada sector, para a variação aceitável da dimensão de cada sector. Assim, o carácter C_i faz parte da cadeia de carácter CC_j se:

$$\left\{ \begin{array}{l} |S1_{C_i} - S1_{CC_j}| < Th_{s_1} \\ |S2_{C_i} - S2_{CC_j}| < Th_{s_2} \\ |S3_{C_i} - S3_{CC_j}| < Th_{s_3} \\ |S4_{C_i} - S4_{CC_j}| < Th_{s_4} \end{array} \right. \quad (4.7)$$

Onde $| |$ representa os valores absolutos da diferença entre a dimensão dos sectores do carácter C_i e a área média dos sectores da cadeia de carácter CC_j , medidas em *pixels*. O valor de limiar Th_{s_i} para dada sector é dado pela seguinte expressão:

$$Th_{s_i} = Si_{CC_j} \times 0.25 \quad (4.8)$$

ou seja, Th_{s_i} é igual a 25% da dimensão média (em *pixels*) dos sectores dos caracteres que já fazem parte da cadeia de carácter CC_j .

O processo de seguimento do texto consiste na formação de cadeias de carácter. A estrutura adoptada na presente Tese para representar as cadeias de carácter é formada por três tipos de informação ($A, [i, f], D$):

- A , armazena para cada trama a assinatura da cadeia de carácter, i.e. as características do carácter que representa a cadeia de carácter nessa trama: luminância média, dimensão média (número de *pixels* da região) posição e forma;
- $[i, f]$, armazena o intervalo de tramas em que o carácter que originou a cadeia de carácter está presente no vídeo ou seja a duração da cadeia de carácter;
- D , armazena a direcção e o valor médio do deslocamento entre tramas da cadeia de carácter. A direcção, q , é dada pela expressão:

$$q = \tan^{-1} \left(\frac{dy}{dx} \right) \quad (4.9)$$

e o deslocamento, d , é dado pela expressão:

$$d = \sqrt{dx^2 + dy^2} \quad (4.10)$$

Onde dx e dy são os valores médios das duas componentes do deslocamento da cadeia de carácter entre duas tramas, medidos em *pixels/trama*, segundo o eixo do xx e yy , respectivamente.

Deste modo, a formação de cadeias de carácter pode ser descrita da seguinte forma:

1. Cada carácter C_i , pertencente à trama n , é comparado com todas as cadeias de carácter CC_j , $j \in \{1, \dots, t\}$ que existam nas tramas $n-8$ a $n-1$, ou seja, aquelas que ainda se podem propagar para a trama actual;

2. Se o carácter C_i cumprir todos os critérios de semelhança anteriormente definidos em relação a uma dada cadeia de carácter, então esse carácter é incluído nessa cadeia de carácter. No caso de cumprir todos os critérios de semelhança com mais de uma cadeia de carácter, o carácter é incluído naquela com a qual a sua semelhança é maior, i.e. aquela cuja diferença entre os parâmetros da sua assinatura e os parâmetros da assinatura da cadeia for menor. Para possibilitar medir a semelhança entre um dado carácter C_i e a cadeia de carácter CC_j em termos da sua assinatura, foi definida uma métrica de semelhança S_g que permite avaliar a semelhança existente entre o carácter C_i e a cadeia de carácter CC_j . A semelhança, S_g , é dada pela seguinte expressão:

$$S_g = \frac{S_l + S_d + S_p + S_f}{4} \quad (4.11)$$

Onde S_l , S_d , S_p e S_f representam a semelhança entre os parâmetros luminância, dimensão, posição e forma, para as assinaturas do carácter C_i e da cadeia de carácter CC_j .

As medidas de semelhança para cada parâmetro que faz parte da assinatura, variam entre 0 e 1, assumindo o valor 1 quando o valor do parâmetro (luminância, dimensão, posição e forma) do carácter e o valor médio do mesmo parâmetro da cadeia de carácter são idênticas e vão diminuindo à medida que a diferença entre ambos aumenta. Assim, a métrica de semelhança, S_i , entre o parâmetro P_i do carácter C_i e o mesmo parâmetro da cadeia de carácter CC_j é dada pela expressão:

$$S_i = \frac{\left| P_{iC_i} - P_{iC_j} \right| - P_{iC_j}}{P_{iC_j}} \quad (4.12)$$

Os parâmetros de luminância, dimensão, posição e forma da cadeia de carácter são actualizados com os valores médios obtidos usando a cadeia já com a região recém adicionada;

3. Se a região R_i não for semelhante a nenhuma das cadeias de carácter existentes, uma nova cadeia de carácter é formada e inicializada com as características dessa região;
4. No final do processamento de cada trama, são verificadas quais as cadeias de carácter que terminaram e aquelas que se podem propagar para a trama seguinte. São consideradas cadeias de carácter que terminaram todas aquelas que não cumpram um dos seguintes critérios:
 - ♦ Tenham sido criadas na trama $n-1$ e não continuem na trama n ;
 - ♦ Não tenham sido seguidas nos últimos 0.32 segundos, i.e. 8 tramas para uma resolução temporal de 25 tps;
 - ♦ A sua posição estimada seja fora da imagem.

De seguida verifica-se para as cadeias de carácter que terminaram, aquelas que são válidas. São consideradas cadeias de carácter válidas todas aquelas que cumprirem os seguintes critérios:

- ♦ Tenham uma duração superior a 0.25 segundos, i.e. 6 tramas para uma resolução temporal de 25 tps e tenham ocorrido em mais de 3 tramas;
- ♦ Tenham um movimento inferior a 250 *pixels* por segundo, i.e. 10 *pixels/trama* para uma resolução temporal de 25 tps.

Os valores usados nestes critérios foram obtidos empiricamente através de testes exaustivos. As tramas classificadas como inválidas são eliminadas. A eliminação destas cadeias de carácter permite acabar com as falsas detecções em tramas individuais, uma vez que estas não são consistentes no tempo.

Na Figura 4.11 é ilustrado graficamente o processo de formação de cadeias de carácter. CC_1 e CC_2 representam cadeias de caracteres válidas pois cumprem as condições anteriormente definidas. CC_3 e CC_4 representam cadeias de caracteres inválidas e como tal são eliminadas; a cadeia CC_3 ocorreu em menos de 4 tramas e a cadeia CC_4 tem um intervalo de duração inferior a 6 tramas.

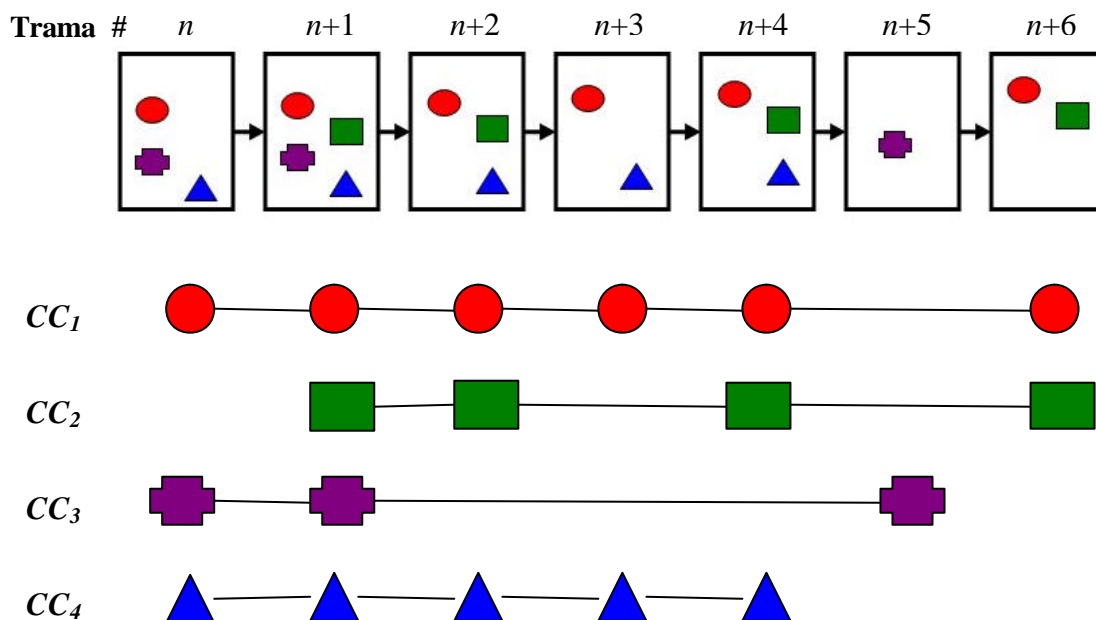


Figura 4.11 – Exemplo da formação de cadeias de carácter: as figuras geométricas correspondem a regiões classificadas como caracteres. CC_1 e CC_2 representam cadeias de caracteres válidos; CC_3 e CC_4 representam cadeias de caracteres inválidos.

Deste modo, no final da fase de seguimento do texto, foram construídas as várias cadeias de caracteres correspondentes aos caracteres detectados, uma para cada carácter, as quais representam a evolução dos vários caracteres ao longo do tempo. As regiões conexas classificadas como caracteres na fase de monitorização do texto e que não são consistentes no tempo são eliminadas. Seguidamente, é efectuada a integração do texto, ou seja, procura-se agrupar as várias cadeias de caracteres de modo a formar palavras.

4.2.3.2 Integração de Texto

A integração do texto visa o agrupamento das cadeias de caracteres de modo a formar palavras. As cadeias que não forem incluídas nas palavras formadas nesta fase são eliminadas por se considerar que correspondem a regiões que não fazem parte do texto. Deste modo, consegue-se, também, o refinamento das cadeias de caracteres atrás efectuada. Assim, para formar palavras e linhas, procede-se ao agrupamento de todas as cadeias de carácter detectadas ao longo das várias tramas da sequência de vídeo, provenientes das fases anteriores e que foram classificadas como texto [Sato99, Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Estes sistemas foram concebidos para formar palavras escritas segundo a direcção horizontal.

O algoritmo de integração desenvolvido no âmbito da presente Tese para efectuar a formação de palavras em sequências de vídeo tem por base o algoritmo proposto para a formação de palavras em imagens apresentado no Capítulo 3. Para explorar a redundância temporal existente no vídeo, foi adicionada a este algoritmo a componente temporal. Este algoritmo apresenta como grande vantagem em relação aos algoritmos estudados no Capítulo 2 a capacidade de agrupar texto escrito em qualquer direcção com um pequeno número de falsas detecções. Assim, de forma semelhante ao que foi efectuado para a extracção de texto em imagens, assume-se que o texto consiste em grupos de mais de duas regiões, alinhadas em qualquer direcção e que ocorrem nas mesmas tramas. Assume-se, também, que essas regiões estão relativamente próximas umas das outras quando se encontrarem sobre a recta que passa pelo centro das regiões de início e de fim da palavra. Desta forma, o algoritmo proposto para a detecção de palavras em vídeo considera cinco fases distintas que serão de seguida apresentadas em detalhe: criação de palavras, selecção de caracteres de referência, recuperação de regiões, eliminação de palavras sobrepostos e rotação do texto.

1ª Fase – Criação de palavras

Esta fase visa o agrupamento das várias cadeias de carácter (correspondentes a um carácter) resultantes do seguimento do texto de modo a formar palavras. Para efectuar os agrupamentos, há que estabelecer os critérios que permitem definir, com exactidão, o modo como as palavras vão ser constituídas. Tais critérios permitem não só definir os agrupamentos de cadeias de carácter, mas também quais desses agrupamentos correspondem a palavras. Assim, os critérios para a formação de agrupamentos de cadeias de carácter são os seguintes:

- **Coexistência temporal** – As cadeias de carácter têm que existir nas mesmas tramas para que façam parte da mesma palavra. Na presente Tese, considera-se que as cadeias de carácter possuem uma coexistência temporal válida se coexistirem em pelo menos quatro tramas, que podem ser contíguas ou não;
- **Proximidade espacial** – Para que façam parte da mesma palavra, as cadeias de carácter devem estar suficientemente próximas umas das outras no contexto da trama. Assim, ao considerar as cadeias de carácter CC_1, \dots, CC_n , que foram classificadas como possíveis caracteres nas fases anteriores, deve aplicar-se o critério de proximidade espacial para que estas possam ser agrupadas numa palavra. A distância entre duas cadeias de carácter CC_1 e CC_2 , numa dada trama T_k , corresponde à distância entre os centros das *bounding boxes* correspondentes aos caracteres que representam as duas cadeias na trama T_k . Deste modo, para que a cadeia de carácter CC_i faça parte da palavra P_j nas tramas onde ambas existem, a distância d entre o ponto correspondente ao centro da *bounding box* da cadeia de carácter e o centro da *bounding box* de pelo menos uma cadeia de carácter que faça

parte da palavra P_j tem que ser inferior a um determinado limiar, Th_{dist} . Assim, a cadeia de carácter CC_i de coordenadas (x_{CC_i}, y_{CC_i}) na trama T_k , faz parte da palavra $P_j = \{CC_1, \dots, CC_n\}$ de coordenadas $\{(x_{CC_1}, y_{CC_1}), \dots, (x_{CC_n}, y_{CC_n})\}$ na trama T_k , se $d < Th_{dist}$, onde a distância, d , é dada pela expressão (4.9).

$$d_r = \sqrt{(x_{CC_i} - x_{CC_r})^2 + (y_{CC_i} - y_{CC_r})^2} \text{ com } r = 1, 2, \dots, n \quad (4.13)$$

Com vista à definição prévia do valor de Th_{dist} , há que ter em consideração o facto da distância entre palavras ser tipicamente superior à distância entre caracteres que façam parte da mesma palavra. Assim, testes exaustivos permitiram definir Th_{dist} da seguinte forma:

$$Th_{dist} = 3 \times h_{\min} \text{ onde } h_{\min} = \min \{h_1, h_2, h_3\} \quad (4.14)$$

Em que $\{h_1, h_2, h_3\}$ representa a altura das *bounding boxes* das regiões correspondentes às três cadeias de caracteres que deram origem à palavra.

Quando a distância entre a cadeia de carácter CC_i e pelo menos uma das cadeias de carácter que formam a palavra P_j é inferior a Th_{dist} , em mais de quatro tramas, considera-se que a cadeia de carácter CC_i faz parte da palavra P_j ;

- **Alinhamento** – As cadeias de carácter devem estar alinhadas ao longo de uma dada direcção para que façam parte da mesma palavra. Assim, definiu-se um intervalo de tolerância para o alinhamento dessas mesmas cadeias de carácter ao longo de uma direcção, dependendo dessa tolerância da altura das cadeias de carácter em questão. Assim, ao considerar as cadeias de carácter CC_1, \dots, CC_n , que foram classificadas como possíveis caracteres nas fases anteriores, deve aplicar-se o critério de alinhamento nas tramas onde elas existem para que estas possam ser agrupadas numa palavra. Deste modo, para que a cadeia de carácter CC_i faça parte da palavra P_j nas tramas onde ambas existem, a distância d entre o ponto correspondente ao centro da *bounding box* da cadeia de carácter e a recta que passa pelos centros das *bounding boxes* das cadeias de caracteres de início e de fim da palavra P_j tem que ser inferior a um determinado limiar, Th_{alin} , sendo o valor de Th_{alin} definido por:

$$Th_{alin} = \frac{h_{\min}}{3} \quad (4.15)$$

Assim, a cadeia de carácter CC_i de coordenadas (x_{CC_i}, y_{CC_i}) na trama T_k , faz parte da palavra $P_j = \{CC_1, \dots, CC_n\}$ de coordenadas $\{(x_{CC_1}, y_{CC_1}), \dots, (x_{CC_n}, y_{CC_n})\}$ na trama T_k , se $d < Th_{alin}$, onde a distância, d , é dada pela expressão (4.12).

$$d = \left| \frac{x_{CC_i} - m \times y_{CC_i} - b}{\sqrt{1 + b^2}} \right| \quad (4.16)$$

As coordenadas (x_{CC}, y_{CC}) da cadeia de carácter, CC , correspondem à posição do centro da sua *bounding box*. O valor de b é a ordenada na origem da recta e o m é o declive da recta que passa no centro das cadeias de carácter CC_1 e CC_n .

Quando a distância entre o centro da cadeia de carácter em questão e a recta que passa pelos centros das cadeias de caracteres de início e de fim da palavra é inferior a Th_{alin} , em mais de quatro tramas considera-se que a cadeia de carácter CC_i faz parte da palavra P_j ;

- **Altura** – As cadeias de carácter devem ter uma diferença mínima de altura para que façam parte da mesma palavra. Para que isso se verifique, definem-se dois limiares: um para a altura mínima da palavra (Th_{hmin}) e outro para a altura máxima da palavra (Th_{hmax}). A necessidade de definir dois limites de altura para as palavras deve-se ao facto de, na mesma palavra, existirem normalmente letras com vários tamanhos. Assim, a cadeia de carácter CC_i faz parte da palavra P_j se em mais de quatro tramas onde ambas existam se verificar a condição (4.13):

$$Th_{hmin} < h(CC_i) < Th_{hmax} \text{ onde } Th_{hmin} = h_{min}(P_j) \times 0.9 \quad (4.17)$$

onde $h(CC_i)$ e $h_{min}(P_j)$ são, respectivamente, a altura da cadeia (altura do carácter que representa a cadeia na trama) e a altura mínima da palavra numa trama onde ambas existam;

- **Movimento** – Para que façam parte da mesma palavra, as cadeias de carácter a agrupar devem possuir um movimento semelhante. Para que isso se verifique, é definido um valor de limiar para a diferença de deslocamento das cadeias entre tramas sucessivas, Th_{desl} . Assim, a cadeia de carácter CC_i faz parte da palavra P_j se durante pelo menos quatro tramas onde ambas existam:

$$\left| D_{(CC_i)} - D_{(P_j)} \right| < Th_{desl} \quad (4.18)$$

Onde $\left| \right|$ representa o valor absoluto da diferença entre a média do deslocamento das cadeias que formam a palavra P_j e o movimento médio entre tramas da cadeia de carácter CC_i . Foi utilizado $Th_{desl} = 1$ (pixels/trama), valor que foi obtido empiricamente através de testes exaustivos;

- **Luminância** – Para que façam parte da mesma palavra, as várias cadeias de carácter devem possuir uma luminância média semelhante. Para que isso se verifique, é definido um valor de limiar para a diferença entre luminâncias médias, Th_{lum} . Assim, a cadeia de carácter CC_i faz parte da palavra P_j se em mais de quatro tramas onde ambas existam se verificar a condição:

$$\left| Y_{(CC_i)} - Y_{(P_j)} \right| < Th_{lum} \quad (4.19)$$

Onde $\left| \right|$ representa o valor absoluto da diferença entre a luminância média da cadeia de carácter CC_i e a luminância média da palavra P_j nas tramas onde ambas existem. Este critério é importante para evitar a formação de falsas palavras, por exemplo, devido a palavras com sombra ou palavras sobre fundo muito texturado. Nesta Tese, usa-se $Th_{lum} = 25$, valor que foi obtido empiricamente através de testes exaustivos;

- **Dimensão da palavra** – Os agrupamentos formados por três ou mais cadeias de carácter são classificados como palavras, sendo os restantes eliminados. A eliminação de palavras

com menos de três cadeias de carácter ocorre por se considerar que estas palavras possuem pouco valor semântico.

O processo de agrupamento das cadeias de carácter consiste na formação de palavras ou mesmo linhas de texto. Em conclusão, uma palavra válida $P_i = \{CC_{i_1}, \dots, CC_{i_n}\}$ é formada, no mínimo, por três cadeias de carácter com as seguintes características:

- Existem simultaneamente em pelo menos 4 tramas;
- São vizinhas próximas;
- Estão orientadas segundo a mesma direcção;
- Possuem uma diferença mínima de altura;
- Possuem movimento semelhante;
- Possuem luminâncias médias semelhantes.

Assim, a formação de palavras consiste em dois passos:

1º Inicialização de palavras – Neste passo, inicializam-se novas palavras, ou seja, procuram-se combinações de três cadeias de carácter que representem uma palavra válida, i.e. que cumpra os critérios supracitados em pelo menos 4 tramas. Para tal, as cadeias de caracteres são combinadas 3 a 3 e, para todas as tramas onde estas cadeias existem simultaneamente, verifica-se se cumprem, ou não, os critérios de formação de palavras anteriormente descritos. As combinações que cumprirem os critérios em pelo menos 4 tramas são consideradas válidas e dão origem a uma nova palavra. Logo que seja encontrado um conjunto de três cadeias válido, as cadeias de carácter correspondentes são removidas do agrupamento de cadeias de carácter e adicionados à nova palavra recém formada e a 2ª etapa é iniciada;

2º Preenchimento das palavras – Nesta etapa, pretende-se completar as palavras iniciadas na etapa anterior. Desta forma, para todas as cadeias de carácter restantes no conjunto de cadeias de carácter, verifica-se se estas cumprem os critérios, anteriormente estabelecidos, em relação à palavra que se criou na etapa anterior. As cadeias de carácter que cumprirem os critérios de pertença a uma palavra são removidas do agrupamento e adicionadas à nova palavra.

Este processo de procura da próxima palavra válida e da integração de cadeias de carácter que cumpram os critérios de formação de palavras, repete-se até que não seja possível formar mais palavras válidas, ou então, que não existam mais cadeias de carácter para agrupar. As cadeias de carácter que não forem agrupadas são eliminadas. O início e o fim da palavra coincide com o início da primeira cadeia de carácter a ocorrer nessa palavra e o seu fim com o fim da última cadeia de carácter a ocorrer na mesma palavra. O início e o fim de uma palavra podem ser observados na Figura 4.12, onde coincidem com o início e o fim das cadeias de carácter CC_3 e CC_7 , respectivamente.

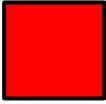


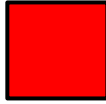



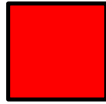





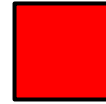
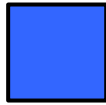
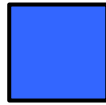








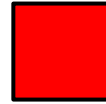
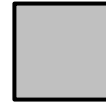


Trama #	n	$n+1$	$n+2$	$n+3$	$n+4$	$n+5$	$n+6$
CC_1							
CC_2							
CC_3							
P_1							

Figura 4.12 – Exemplo da formação de uma palavra P_1 formada a partir de três cadeias de caracter, CC_1 , CC_2 e CC_3 . Os quadrados azuis representam as tramas onde as cadeias de caracter e a palavra são detectadas, os quadrados vermelhos representam tramas onde as cadeias de caracter e a palavra não são detectadas e os quadrados cinzentos representam tramas onde a palavra está incompleta.

2ª Fase – Selecção de caracteres de referência

Esta fase visa efectuar a selecção dos caracteres que melhor representam cada uma das cadeias de caracteres incluídas numa dada palavra, i.e. seleccionar de entre os caracteres que fazem parte de uma cadeia de caracter aquele que apresenta a melhor semelhança com essa cadeia de caracter caracterizada através dos seus valores médios. Esta necessidade de procurar o caracter que melhor representa cada cadeia de caracter surge da necessidade de possuir o caracter que melhor a representa tanto para efectuar a interpolação com o objectivo de recuperar caracteres em falta como para representar na imagem final cada cadeia de caracter com um único caracter. Assim, para definir qual o caracter associado à cadeia de caracter que apresenta a maior semelhança com os valores médios das características da cadeia (dimensão e luminância), há que estabelecer métricas que permitam definir, com exactidão, a qualidade da comparação. Desta forma, as métricas de comparação entre a cadeia de caracter e os caracteres que a formam são as seguintes:

- **Métrica de dimensão** – Permite medir a semelhança entre o caracter C_i e a cadeia de caracter CC_j em termos da sua dimensão. Este parâmetro varia entre 0 e 1, assumindo o valor 1 quando a dimensão do caracter e a dimensão média da cadeia de caracter são idênticas e diminuindo à medida que a diferença entre ambos aumenta. A métrica de dimensão, D_i , entre o caracter C_i e a cadeia de caracter CC_j é dada pela expressão:

$$D_i = \frac{\left| AC_i - ACC_j \right| - ACC_j}{ACC_j} \quad (4.20)$$

Onde AC_i representa a área do carácter C_i e ACC_j representa a área média da cadeia de carácter CC_j , medidas em número de *pixels*;

- **Métrica de luminância** – Permite medir a semelhança entre a luminância média do carácter C_i e a luminância média da palavra CC_j . Este parâmetro varia entre 0 e 1, assumindo o valor 1 quando a luminância média do carácter e a luminância média da cadeia de carácter são idênticas e diminuindo à medida que a diferença entre ambos aumenta. A métrica de luminância, L_i , entre o carácter C_i e a cadeia de carácter CC_j é dada pela expressão:

$$L_i = \frac{\left| YC_i - YCC_j \right|}{YCC_j} \quad (4.21)$$

Onde YC_i representa o valor médio da luminância do carácter C_i e YCC_j representa o valor médio da luminância da cadeia de carácter CC_j ;

- **Métrica global** – Permite integrar a informação das métricas de dimensão e de luminância. Para cada carácter C_i associado à cadeia de carácter CC_j , é calculada uma métrica global de semelhança baseada nas métricas de tamanho e luminância, dada pela expressão:

$$S_i = \frac{DC_i + LC_i}{2} \quad (4.22)$$

Desta forma, a cadeia de carácter CC_j , formado pelo conjunto de caracteres $\{C_1, \dots, C_n\}$, é representado pelo carácter de referência para o qual $C_{ref} = \max\{S_{C_1}, \dots, S_{C_n}\}$, onde $\{S_{C_1}, \dots, S_{C_n}\}$ representa o conjunto com todas as métricas globais de semelhança, calculadas para todos os caracteres que formam a cadeia de caracteres CC_j .

3ª Fase – Recuperação de caracteres

Esta fase visa completar as palavras incompletas, i.e. as palavras formadas por cadeias de carácter com durações diferentes, dando assim, origem a palavras incompletas para algumas das tramas.

Na Figura 4.12 representa-se uma palavra, P_I , formada por três cadeias de carácter, CC_1 , CC_2 e CC_3 de dimensões diferentes. Os quadrados a azul representam as tramas onde as cadeias de carácter e a palavra foram detectadas, os quadrados a vermelho representam tramas onde as cadeias de carácter e a palavra não foram detectadas e os quadrados cinzentos representam tramas onde a palavra está incompleta. Assim, através da observação da Figura 4.12, facilmente se verifica que a palavra na trama n só é representado pela cadeia de carácter C_3 , na trama $n+3$ não possui qualquer representação e na trama $n+4$ é representada apenas por duas das três cadeias de carácter. Esta situação em termos de extracção de texto origina a falta da palavra na trama $n+3$ e origina palavras incompletas nas tramas n e $n+4$, devido à falta de caracteres nessas tramas.

Assim, de forma a completar as palavras, estas são estendidas para o tamanho da sua maior cadeia de carácter e os caracteres em falta nas cadeias de carácter, são recuperados com

recurso à interpolação. O efeito pretendido com a recuperação de caracteres, i.e. a recuperação dos caracteres em falta nas cadeias de carácter, é ilustrado na Figura 4.13: a Figura 4.13 (a) representa a cadeia de carácter CC_2 antes da recuperação de caracteres, a Figura 4.13 (b) ilustra a expansão da cadeia de carácter CC_2 e a Figura 4.13 (c) e (d) ilustram a interpolação dos caracteres em falta baseada na trama de trás e na trama da frente, respectivamente.

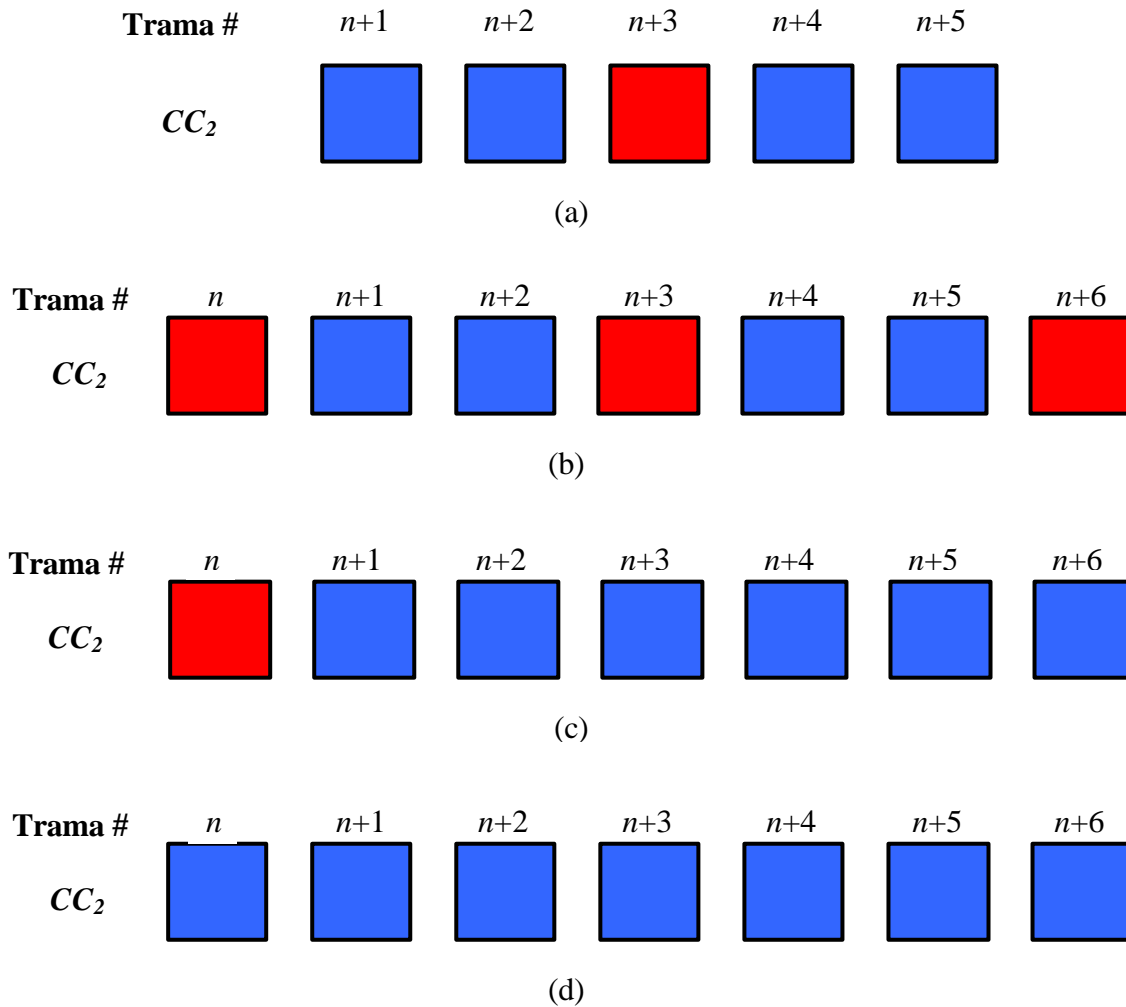


Figura 4.13 – Exemplo da interpolação das regiões em falta numa cadeia de carácter. A imagem (a) ilustra a cadeia de carácter antes da recuperação de caracteres; a imagem (b) ilustra o efeito da expansão da cadeia de carácter e as imagens (c) e (d) ilustram a interpolação dos caracteres em falta, baseada na trama de trás e na trama da frente, respectivamente. Os quadrados a azul e a vermelho, representam as tramas onde os caracteres foram detectados e onde a sua detecção falhou, respectivamente.

A informação temporal, ou seja, o intervalo de tempo onde a cadeia de carácter CC_i foi detectada $[i, f]$ pode ser diferente para as várias cadeias de carácter que constituem a palavra, P_j , devido a falhas no processo de detecção. Para além desta situação, as cadeias de carácter que formam a palavra podem ainda possuir ‘buracos’, i.e. tramas pertencentes ao intervalo $[i, f]$ nas quais a cadeia de carácter não foi detectada. Estas situações podem ocorrer quer por falhas na detecção de caracteres, quer pela falta de caracteres em algumas tramas.

Para efectuar a recuperação das regiões em falta na palavra, procede-se à interpolação das regiões em falta nas cadeias de carácter. O processo de recuperação das regiões em falta divide-se em duas etapas, as quais podem ser descritas da seguinte forma:

1ª Extensão da duração das cadeias de carácter – Nesta etapa, procede-se unicamente à extensão de todas as cadeias de carácter pertencentes à palavra. Estas são estendidas através da redefinição do seu intervalo de duração (trama de início e de fim), para a duração temporal da palavra, i.e. número de tramas em que a palavra existe ou seja foi detectada. A duração temporal da palavra é representada por $[\min \{i_{C_1}, \dots, i_{C_n}\}, \max \{f_{C_1}, \dots, f_{C_n}\}]$, em que $\{i_{C_1}, \dots, i_{C_n}\}$ e $\{f_{C_1}, \dots, f_{C_n}\}$ representam o início e o fim das várias cadeias de carácter, respectivamente. A extensão da duração das cadeias de caracteres é ilustrada na Figura 4.13 (b);

2ª Interpolação dos caracteres em falta nas cadeias de carácter – Nesta etapa, para cada cadeia de carácter procede-se à estimativa das posições dos caracteres que se pretende recuperar nas várias tramas. Para tal, é utilizada quer a informação de deslocamento da cadeia de carácter, quer a posição da cadeia de carácter na trama mais próxima onde o carácter tenha sido detectado e ainda a informação de deslocamento das cadeias de carácter adjacentes. De modo a possibilitar a recuperação das regiões perdidas, quer no início, quer no fim da cadeia de carácter, a interpolação é efectuada de dois modos distintos (aplicados sequencialmente se o primeiro modo não resultar): um baseado na trama anterior, o outro na trama posterior:

1º Interpolação baseada na trama anterior

A posição da cadeia de carácter que se pretende recuperar é estimada utilizando a informação referente à posição da palavra à qual ela pertence na trama anterior, i.e. utiliza-se a posição da palavra na trama $n-1$ para estimar a posição da palavra na trama n . Para efectuar a descrição do processo de interpolação vai-se recorrer ao exemplo de uma palavra formada por três cadeias de carácter CC_1 , CC_2 e CC_3 ilustrado na Figura 4.14; nesta figura os quadrados a azul representam as tramas onde as cadeias de carácter são detectadas e os quadrados vermelhos representam tramas onde as cadeias de carácter não são detectadas. A interpolação baseada na trama anterior é feita utilizando duas técnicas: uma que utiliza a informação das cadeias de carácter adjacentes desde que estas possuam detecções nas tramas onde se pretende efectuar a recuperação de caracteres e outra que utiliza a informação do deslocamento da cadeia de carácter para estimar a posição do carácter em falta.

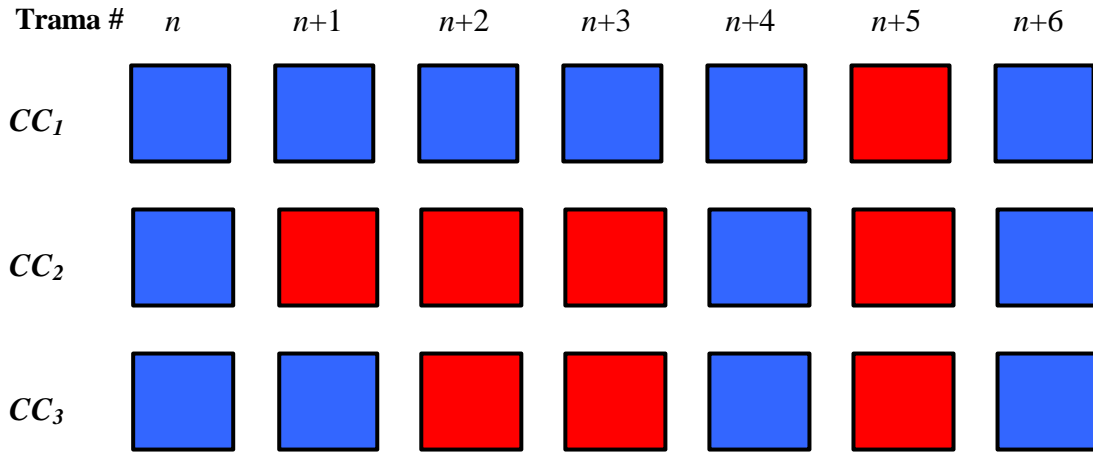


Figura 4.14 – Exemplo da recuperação de caracteres

Os dois tipos de interpolação serão descritos de seguida:

- ♦ **Interpolação baseada em cadeias de caracter adjacentes** – Este tipo de interpolação parte do pressuposto que as cadeias de caracter que fazem parte da mesma palavra possuem deslocamentos idênticos. Para recuperar os caracteres da cadeia de caracter CC_2 que não foram detectados nas tramas $n+1$ a $n+3$, utilizando a interpolação baseada em cadeias de caracter adjacentes, é necessário que pelo menos uma das cadeias de caracter (CC_1 e CC_3) adjacentes à cadeia de caracter CC_2 possua detecções nas tramas n a $n+3$. No exemplo ilustrado na Figura 4.14 a cadeia de caracter CC_1 cumpre este requisito. Assim, as coordenadas (x_{C_i}, y_{C_i}) do caracter C_i em falta na cadeia de caracter CC_2 na trama $n+2$, podem ser estimadas com base na última detecção (trama n) na cadeia CC_2 e nas detecções da cadeia CC_1 nas tramas n e $n+2$, através da seguinte expressão:

$$\begin{cases} (x_{C_i})_{n+2} = (x_{CC_1})_{n+2} + (x_{CC_2})_n - (x_{CC_1})_n \\ (y_{C_i})_{n+2} = (y_{CC_1})_{n+2} + (y_{CC_2})_n - (y_{CC_1})_n \end{cases} \quad (4.23)$$

As coordenadas (x_C, y_C) e (x_{CC}, y_{CC}) do caracter, C , e da cadeia de caracter, CC , correspondem à posição dos centros das suas *bounding boxes*, respectivamente;

- ♦ **Interpolação baseada no deslocamento da própria cadeia de caracter** – Este tipo de interpolação utiliza a informação do deslocamento da cadeia de caracter para estimar a posição do caracter em falta. Assim, as coordenadas (x_{C_i}, y_{C_i}) do caracter C_i em falta na cadeia de caracter CC_2 na trama $n+2$, podem ser estimadas com base na informação do deslocamento calculada até à última detecção (trama n) na cadeia CC_2 , através da seguinte expressão:

$$(x_{C_i}, y_{C_i})_{n+2} = (x_{CC_2}, y_{CC_2})_n + (dx, dy) \times k \quad (4.24)$$

Onde k representa o número de tramas entre a trama onde ocorreu a última detecção em CC_2 e a trama onde se pretende recuperar o carácter e (dx, dy) é o valor médio do deslocamento da cadeia de carácter CC_2 , entre duas tramas sucessivas, dado pela expressão (4.6).

Primeiro, tenta-se recuperar os caracteres utilizando a interpolação baseada em cadeias de carácter adjacentes. Só quando não é possível recuperar os caracteres utilizando esta técnica é que se recorre à interpolação baseada no deslocamento da própria cadeia de carácter. A primeira técnica permite recuperar com rigor caracteres cuja detecção falhou durante várias tramas sucessivas.

2º Interpolação baseada na trama posterior

Neste caso, a posição da cadeia de carácter que se pretende recuperar é estimada utilizando a informação referente à posição da palavra à qual ela pertence na trama posterior, i.e. utiliza-se a posição da palavra na trama $n+1$ para estimar a posição da palavra na trama n . Este passo é em tudo idêntico ao anterior com excepção das tramas utilizadas como referência.

Para efectuar a recuperação de caracteres, aplica-se primeiro a interpolação baseada na trama anterior. Aos caracteres que não for possível recuperar com a interpolação baseada na trama anterior, por exemplo caracteres em falta no início das cadeias de carácter, é aplicada a interpolação baseada na trama posterior.

Na recuperação dos caracteres em falta em cada cadeia de carácter, são utilizados para repor os caracteres em falta, cópias dos caracteres de referência.

O efeito da recuperação de caracteres em falta numa cadeia de caracteres pode ser observado na Figura 4.15. Na Figura 4.15 (a) é ilustrada uma sequência de tramas; na Figura 4.15 (b) são ilustrados os resultados da detecção de texto sem efectuar a recuperação de caracteres, i.e. como se de imagens independentes se tratasse; finalmente, na Figura 4.15 (c) ilustra-se o resultado da detecção de texto efectuando a recuperação de caracteres explorando o facto das imagens serem efectivamente tramas de vídeo e haver correlação temporal. Através da observação da Figura 4.15, pode observar-se o efeito das duas etapas da recuperação de caracteres: extensão da duração das cadeias de caracteres e interpolação dos caracteres em falta nas cadeias de carácter. O efeito da extensão das cadeias de caracteres pode ser observado nas cadeias de caracteres correspondentes aos caracteres T, I e M da palavra TIMOTHY, aos quais foi alterado o intervalo de duração, antecipando o seu início. A sua recuperação foi efectuada com recurso à interpolação baseada na trama posterior.

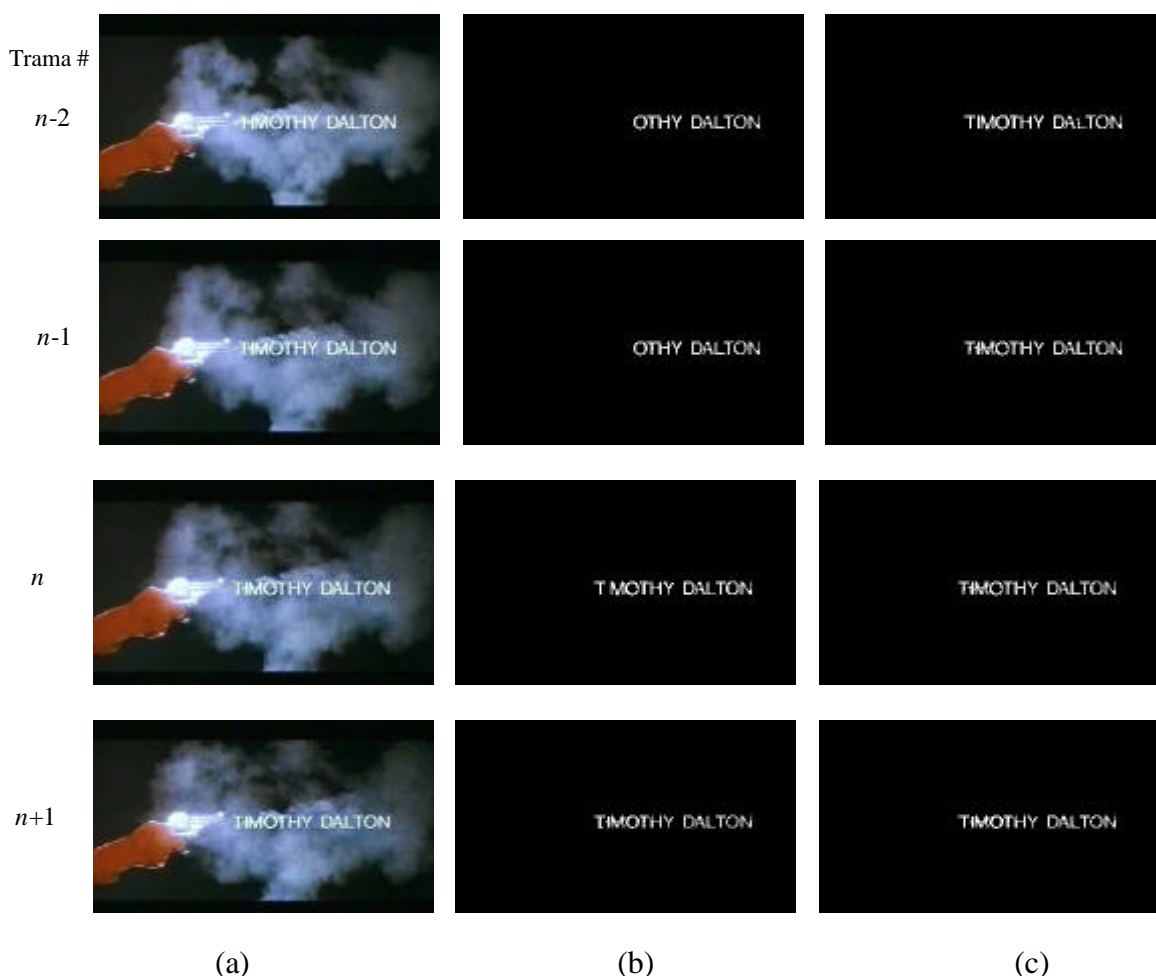


Figura 4.15 – Exemplo da recuperação de regiões: (a) imagens originais; (b) resultados da detecção de texto aplicada às tramas individualmente; e (c) resultado da detecção de texto com análise de movimento e recuperação de caracteres perdidos.

3ª Fase – Eliminação de palavras sobrepostas

Esta fase, tal como para a extracção de texto em imagens apresentada no Capítulo 3, consiste na eliminação de eventuais palavras que se sobreponham, i.e. palavras cujas *bounding boxes* se intersectem. As sobreposições de palavras devem-se normalmente à formação de palavras falsas. Estas palavras falsas resultam tipicamente do agrupamento de caracteres que não fazem parte do texto mas que pela sua forma, dimensões, posição e movimento tenham sido confundidos com texto e classificadas como tal. Um exemplo deste tipo de regiões foi apresentado na Secção 3.2.4.

Sempre que duas ou mais palavras se sobreponham, i.e. as respectivas *bounding boxes* se intersectem em pelo menos uma trama, aquelas que possuírem menor área são eliminadas restando apenas a maior. A área de uma palavra, AP_i , na trama T_k , é definida da seguinte forma:

$$AP_i = \sum_{j=1}^n AC_{i_j} \quad (4.25)$$

Onde AC_i representa a área de cada cadeia de carácter na palavra, i.e. o número de *pixels* do carácter C_i na trama T_k , e n o número total de caracteres existentes na palavra.

4º Fase – Rotação do texto

Esta fase consiste na rotação do texto para a posição horizontal de forma a permitir o seu reconhecimento com maior desempenho por parte dos sistemas OCR. Para tal, e de forma idêntica à extracção de texto em imagens, são considerados dois tipos de texto:

- **Texto vertical** – Este tipo de texto caracteriza-se pela existência de palavras com ângulos de inclinação superiores a 70 graus, formadas por caracteres horizontais (com ângulos de inclinação iguais a 0 graus);
- **Texto inclinado** – Este tipo de texto caracteriza-se pela existência de palavras com ângulos de inclinação inferiores a 70 graus, formadas por caracteres com uma inclinação idêntica à das palavras.

O caso simples do texto horizontal aparece aqui como um caso particular do texto inclinado. Exemplos dos dois tipos de texto, vertical e inclinado, foram apresentados na Secção 3.2.4.

Para o cálculo da rotação do texto, assume-se que este se encontra escrito da esquerda para a direita. No caso particular de texto com uma inclinação de 90 graus, assume-se que este está escrito de cima para baixo. Assim, a rotação do texto é efectuada para todas as tramas onde este existe ao longo do vídeo e é conseguida em dois passos:

1. **Cálculo do ângulo de rotação** – O ângulo de rotação, θ , é definido como o ângulo entre o eixo dos xx e a recta que passa pelo centro das regiões de início e de fim de cada palavra e é calculado para todas as tramas onde a palavra existe. Assim, se se considerar a palavra, P , existente na trama T , formada pelo conjunto de cadeias de caracteres $\{CC_1, \dots, CC_n\}$ com *bounding boxes* de coordenadas $\{(x_{CC_1}, y_{CC_1}), \dots, (x_{CC_n}, y_{CC_n})\}$, o seu ângulo de rotação, θ_T , é dado por:

$$q_T = \tan^{-1}\left(\frac{b}{a}\right) \quad (4.26)$$

onde

$$a = \max\{x_{CC_1}, \dots, x_{CC_n}\} - \min\{x_{CC_1}, \dots, x_{CC_n}\} \quad (4.27)$$

$$b = y_{\max\{x_{CC_1}, \dots, x_{CC_n}\}} - y_{\min\{x_{CC_1}, \dots, x_{CC_n}\}} \quad (4.28)$$

2. **Rotação do texto** – A rotação do texto efectua-se para cada trama onde este existe e é realizada utilizando o ângulo anteriormente calculado. Esta é efectuada de forma diferenciada para o texto vertical e inclinado, do seguinte modo:
 - **Texto vertical** – No texto vertical, a rotação da palavra consiste na translação dos caracteres para a posição horizontal sem a rotação dos mesmos;
 - **Texto inclinado** – No texto inclinado, a rotação da palavra consiste na translação dos caracteres para a posição horizontal acompanhada da sua rotação. Para efectuar a

rotação do texto foi utilizado tal como no capítulo anterior o algoritmo de rotação proposto por Alan Paeth [Paeth86].

Esta necessidade de calcular o ângulo de rotação para todas as tramas onde o texto existe, justifica-se com a possibilidade do texto variar a sua inclinação ao longo do tempo.

4.2.3.3 Representação de Resultados

A representação de resultados tem como objectivo a conversão dos resultados da extracção de texto no vídeo para um formato que seja facilmente interpretado pelo utilizador. A análise de movimento fornece informação detalhada sobre como, onde e quando o texto ocorre. Assim, de forma a tornar o resultado da extracção de texto mais flexível para futuras utilizações, são disponibilizados três formatos com a informação de texto:

1. Uma imagem binária por trama com a representação de cada palavra extraída, na sua localização original, e mantendo a sua inclinação;
2. Uma imagem binária por trama com a representação de cada palavra de texto extraída, segundo a direcção horizontal;
3. Uma imagem binária global com a representação de todas as palavras extraídas da sequência de texto em análise, segundo a direcção horizontal e sequencialmente de acordo com o instante em que surgiram no vídeo. Estas imagens podem ser muito grandes, especialmente quando as sequências de texto têm origem no final de filmes (podem chegar a conter mais de mil caracteres); todavia, são importantes como mapa de bits de entrada para os sistemas OCR.

A imagem de saída é formada pelos caracteres de referência, i.e. pelos caracteres que melhor representam cada cadeia de carácter.

Na Figura 4.16 ilustram-se os três formatos utilizados para fazer o posterior processamento de resultados. A Figura 4.16 (a) ilustra a imagem binária, por trama, com a representação de cada palavra na sua localização original. Este tipo de representação permite visualizar a forma como o texto surge nas várias tramas. A Figura 4.16 (b) ilustra a imagem binária, por trama, com a representação de cada palavra extraída, representada numa linha individual, segundo a direcção horizontal. Esta imagem possibilita o reconhecimento do texto por parte do sistema OCR ao nível da trama. Por último, a Figura 4.16 (c) ilustra a imagem binária com todos os caracteres extraídos da sequência de vídeo, representadas numa linha individual segundo a direcção horizontal. Esta imagem é necessária para permitir o reconhecimento ao nível da sequência de texto.



Figura 4.16 – Exemplo do tipo de imagens utilizadas para visualizar os resultados: (a) imagem binária com a representação de cada palavra na sua localização original; (b) imagem binária com a representação de cada palavra extraída depois da rotação para a direcção horizontal; (c) imagem binária com a representação de todas as palavras extraídas da sequência de vídeo, segundo a direcção horizontal.

Na Figura 4.17 ilustra-se o efeito da análise de movimento na extracção de texto em sequências de vídeo. Na Figura 4.17 (b) apresenta-se o resultado da extracção de texto para uma trama individual; na Figura 4.17 (c) apresenta-se o resultado da extracção de texto para a mesma trama mas desta vez inserida numa sequência de vídeo onde foi aplicada a análise de movimento. Como se pode ver, a análise de movimento possibilita a recuperação de caracteres perdidos na detecção de texto em tramas individuais, contribuindo desta forma para aumentar o desempenho dos sistemas de extracção de texto.

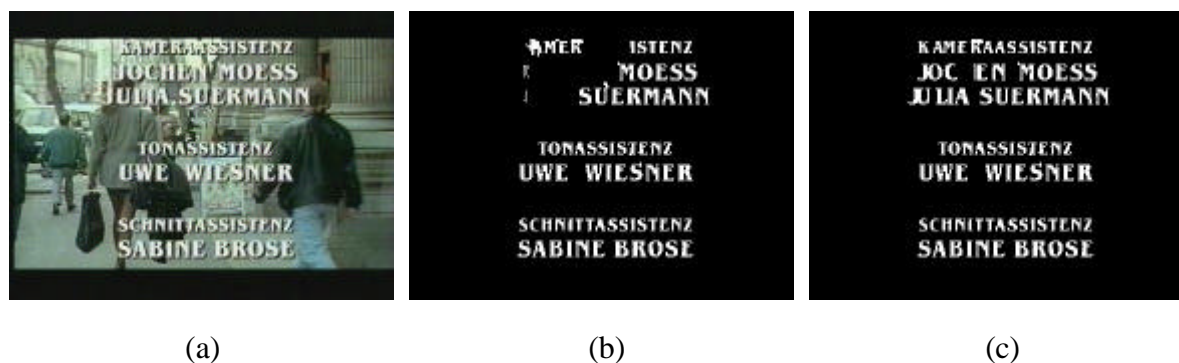


Figura 4.17 – Exemplo do efeito da recuperação de regiões: (a) imagem original; (b) resultado da detecção de texto efectuada sobre uma trama individualmente; (c) resultado da detecção de texto com análise de movimento e recuperação de caracteres perdidos.

4.3 Reconhecimento de Texto

Esta fase visa o reconhecimento do texto detectado na sequência de vídeo utilizando as palavras formadas na fase anterior, usando um sistema OCR. Para efectuar o reconhecimento

do texto nas sequências de vídeo, são utilizados os mesmos sistemas OCR utilizados na extracção de texto em imagens e que foram já descritos no Capítulo 3 (Secção 3.3): um sistema OCR comercial denominado por OmniPage Pro 12.0 [ScanSoft] e um sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95]. Note-se que o desenvolvimento do sistema OCR propriamente dito não fazia parte dos objectivos desta Tese e, por isso, são usados sistemas OCR disponíveis.

4.4 Avaliação de Desempenho

Nesta secção vai efectuar-se a avaliação do desempenho do algoritmo de extracção de texto em sequências de vídeo proposto neste capítulo. Como o objectivo final do algoritmo é o reconhecimento dos caracteres, torna-se natural a utilização de um sistema OCR para efectuar o reconhecimento dos mesmos.

4.4.1 Métricas de Desempenho

De forma semelhante ao que foi efectuado na avaliação do desempenho do algoritmo de extracção de texto em imagens, também aqui se adoptam as métricas *recall* e precisão para avaliar o desempenho do algoritmo proposto para a extracção de texto em sequências de vídeo. Esta escolha justifica-se, essencialmente, por permitir avaliar o desempenho do algoritmo de uma forma mais objectiva uma vez que são aquelas cuja utilização é mais comum [Li00, Lienhart00, Li02, Lienhart02, Wolf02]. Nesta Tese foram definidas métricas quer para a avaliação da detecção de texto, quer para a avaliação do reconhecimento do mesmo:

- **Avaliação da detecção de texto** – A avaliação do desempenho em termos da detecção do texto exprime a capacidade do algoritmo proposto em detectar correctamente caracteres de texto. Para isso, foram utilizadas as métricas *recall* e precisão definidas da seguinte forma:
 - ♦ **Recall** – Relação entre o número de caracteres correctamente detectados e o número de caracteres da *ground truth* na sequência de vídeo ou seja o número total de caracteres que deveria idealmente ser detectado:

$$Recall = \frac{CCD}{CGT} \quad (4.29)$$

Onde *CCD* é o número de caracteres correctamente detectados e *CGT* é o número de caracteres da *ground truth* na sequência de vídeo;

- ♦ **Precisão** – Relação entre o número de caracteres correctamente detectados e o número total de caracteres detectados:

$$Precisão = \frac{CCD}{TCD} \quad (4.30)$$

Onde *TCD* é o número total de caracteres detectados na sequência de vídeo (correcta e erradamente).

- **Avaliação do reconhecimento de texto** – A avaliação do desempenho em termos de reconhecimento do texto exprime a capacidade do sistema desenvolvido para reconhecer correctamente o texto existente nas sequências de vídeo. Este tipo de desempenho é determinado não só pela capacidade do algoritmo proposto para detectar texto mas também pela capacidade dos OCRs usados em reconhecer o texto detectado. Para fazer este tipo de avaliação, usaram-se as métricas tradicionais, precisão e *recall*, definidas do seguinte modo:

- ♦ **Recall** – Relação entre o número de caracteres que foram correctamente reconhecidos e o número de caracteres da *ground truth* na sequência de vídeo:

$$Recall = \frac{CCR}{CGT} \quad (4.31)$$

Onde *CCR* representa o número de caracteres que foram correctamente reconhecidos pelo sistema OCR;

- ♦ **Precisão** – Relação entre o número de caracteres que foram correctamente reconhecidos e o número total de caracteres reconhecidos pelo sistema OCR na sequência de vídeo.

$$Precisão = \frac{CCR}{CSO} \quad (4.32)$$

Onde *CSO* corresponde ao número total de caracteres na saída do OCR para a sequência de vídeo em questão.

Tal como foi referido no Capítulo 3 para as imagens, também para o caso do vídeo é tipicamente difícil obter simultaneamente valores muito elevados de precisão e *recall* (quando um aumenta muito, o outro tende a diminuir), devendo por isso procurar-se o melhor compromisso entre os dois valores. Este melhor compromisso depende do tipo de aplicações: enquanto para certas aplicações ter falsos alarmes não implica, necessariamente, um problema grave já que o importante é mesmo não perder nenhum alarme verdadeiro (por exemplo o seguimento da trajectória de determinada viatura através da verificação da sua matrícula ou em sistemas de vigilância), para outras aplicações a existência de falsos alarmes é gravosa, podendo mesmo admitir-se ‘perder’ alguns casos da *ground truth* se isso permitir diminuir o número de falsos alarmes, por exemplo em navegação automática para confirmar automaticamente determinado percurso.

4.4.2 Condições e Metodologia de Avaliação do Desempenho

Com o objectivo de avaliar o desempenho do algoritmo proposto para a extracção de texto em sequências de vídeo, foi utilizado um conjunto de 13 sequências de vídeo com bastante texto num total de cerca de 14.2 minutos. Estas sequências de vídeo foram seleccionadas de entre vários canais TV e foram capturadas com o recurso à placa de captura de vídeo Pinnacle Linx Video Input Cable, usando resoluções espaciais (luminância) compreendidas entre 352×208 e 384×288 *pixels*.

As sequências de vídeo contêm texto de cena e texto gráfico, alinhado em qualquer direcção, com múltiplas fontes e tamanhos, num total de cerca de 5852 caracteres. As sequências de

vídeo seleccionadas para os testes foram retiradas de títulos e apresentações de filmes onde predomina o texto gráfico, bem como de anúncios e programas de informação onde predomina o texto de cena. No caso de vídeos onde predomina o texto gráfico, foram seleccionados quer vídeos onde o texto se movimenta (movimento de *scroll*), quer vídeos onde o texto está fixo. A avaliação do desempenho do algoritmo proposto será, portanto, efectuada tanto para o texto gráfico, como para texto de cena. Na Figura 4.18 são ilustrados alguns exemplos de vídeo com casos relevantes: a Figura 4.18 (a) ilustra texto de cena; a Figura 4.18 (b) ilustra texto gráfico com movimento e, por último, a Figura 4.18 (c) ilustra texto gráfico fixo.

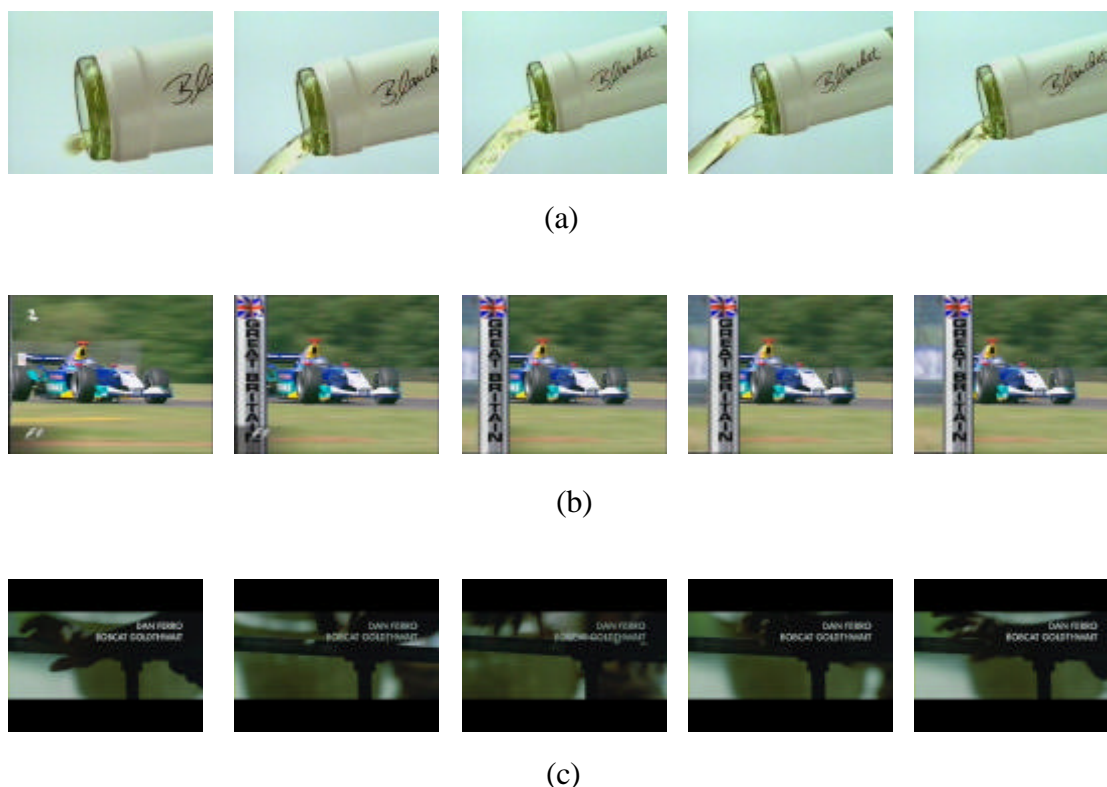


Figura 4.18 – Exemplos com vários tipos de texto em vídeo: (a) texto de cena; (b) texto gráfico com movimento e (c) texto gráfico fixo.

Na avaliação do desempenho em termos de detecção do texto, e antes de processar cada sequência de vídeo com o algoritmo de detecção de texto proposto, é definida para cada sequência de vídeo a sua *ground truth* em termos de texto, ou seja, determinam-se manualmente quais os caracteres de texto existente em cada sequência de vídeo, bem como, a primeira e a última trama onde cada carácter existe, ou seja, é visível. Para tal, efectua-se para cada sequência de vídeo um levantamento manual dos caracteres existentes na mesma e que são relevantes para a detecção de texto, i.e. aqueles que formam palavras. São consideradas palavras a detectar todos os conjuntos de caracteres que possuam as seguintes características:

- Formados por mais de dois caracteres da mesma cor que não se toquem entre si;
- Alinhados segundo uma dada direcção;

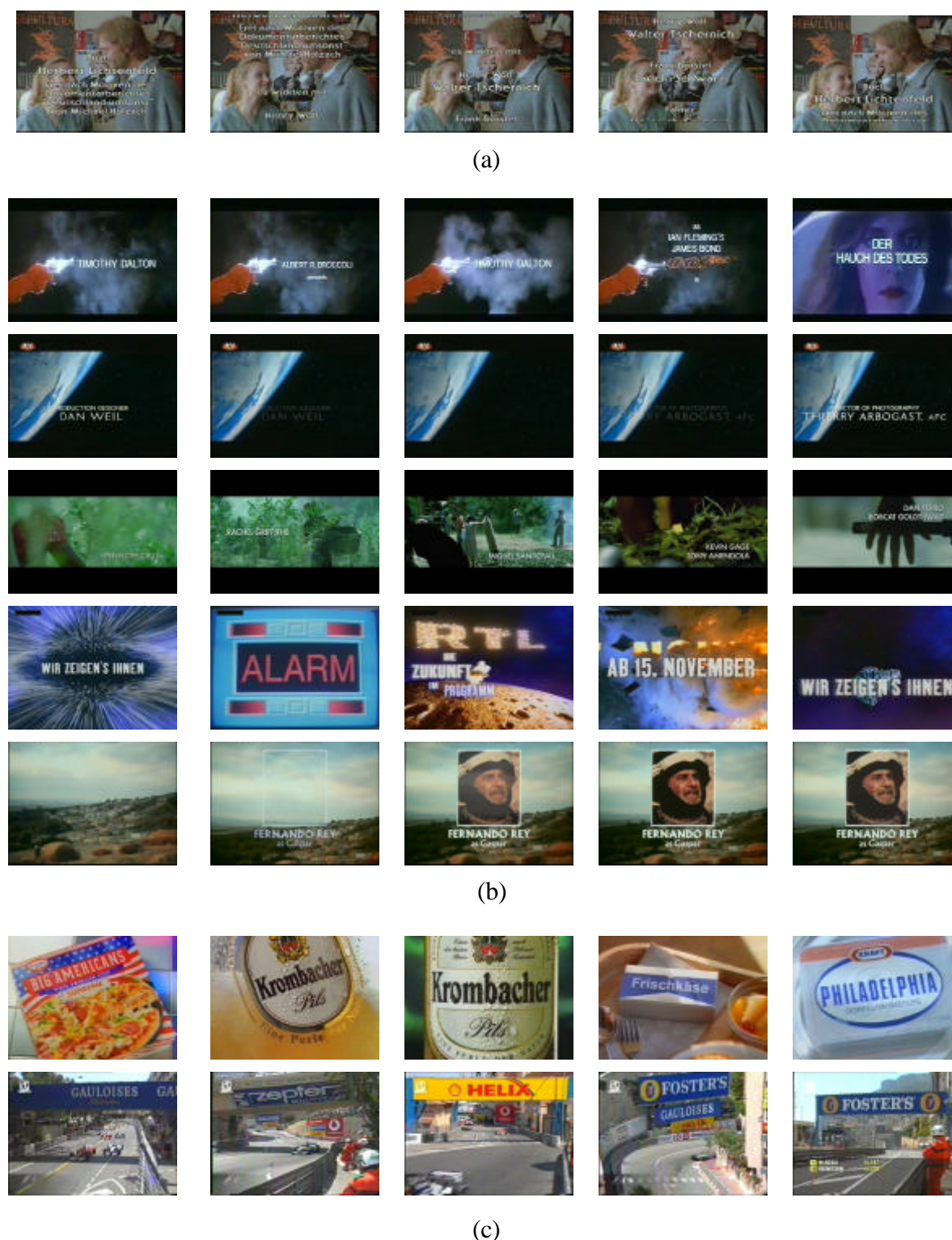


Figura 4.19 – Exemplos de sequências de vídeo que fazem parte do conjunto de teste: (a) sequências onde o texto possui movimento; (b) sequências onde o texto está fixo e o fundo da imagem se movimenta; e (c) sequências onde se movimenta quer o texto, quer o fundo da imagem, com movimentos semelhantes.

4.4.3 Resultados e Análise

Nesta secção são apresentados os resultados obtidos para a avaliação de desempenho efectuada para o algoritmo de extracção de texto em sequências de vídeo proposto neste capítulo. Na Tabela 4.1 apresentam-se os valores dos vários parâmetros utilizados para a configuração do algoritmo de extracção de texto proposto. Estes valores foram aqueles que se revelaram mais eficazes para o conjunto de vídeos utilizados na avaliação do desempenho; uma vez que o conjunto de vídeos utilizado é extremamente variado, espera-se que estes valores para os parâmetros seja adequado para um vasto conjunto de vídeos, não apresentando ‘sintonização’ para qualquer tipo de vídeos em especial.

Tabela 4.1 – Parâmetros utilizados para a avaliação do desempenho.

Limiares para a segmentação	Th_{GD}	
Fase de <i>Split</i>	30	
Fase de <i>Merge</i>	35	
Limiar para a análise de contraste	Th_{cont}	
Contraste entre regiões	10	
Restrições geométricas das regiões	Min	Max
Largura	1	0,25×(largura imagem)
Altura	4	0,25×(altura imagem)
Relação altura/largura	0,4	10
Solidez	0,15	1
Parâmetros para a análise do movimento	Min	Max
Variação da luminância	0	30 (0 – 255)
Nº de detecções (tramas)	4	–
Extensão	7	– (tramas)
Movimento	0	9 (<i>pixels</i> /trama)

4.4.3.1 Avaliação da Detecção de Texto

Na avaliação do desempenho em termos de detecção de texto, foram processados 13 vídeos constituídos por 21298 tramas. Para cada trama determinou-se, manualmente, se cada

caracter relevante ou seja incluído na *ground truth*, tal como definido na secção anterior, foi detectado correctamente ou não. A detecção correcta dos caracteres é determinada com recurso à inspecção visual das imagens binárias criadas pelo algoritmo. A avaliação da detecção de texto foi efectuada para todo o texto tomado como *ground truth* nos vídeos de teste. Os resultados obtidos na avaliação do desempenho do algoritmo na detecção do texto que faz parte da *ground truth*, i.e. texto horizontal, inclinado e vertical, para a totalidade dos vídeos podem ser observados na Tabela 4.2.

Tabela 4.2 – Resultados médios obtidos para a detecção de todo o texto que faz parte da *ground truth* para a totalidade dos vídeos.

Tipos de texto	Nº Tramas	Nº caracteres	Recall	Precisão
Texto de cena	8319	907	0.879	0.836
Texto gráfico	12979	4926	0.953	0.958
Totalidade do texto na <i>ground truth</i>	21298	5833	0.941	0.938

O desempenho do algoritmo proposto pode considerar-se bastante bom já que se obtiveram valores para as métricas de *recall* e precisão que são considerados, por outros investigadores, como sendo muito bons resultados. Na detecção de texto gráfico Li, Lienhart e Wolf em [Li02, Lienhart02 e Wolf02] obtiveram para a métrica *recall* valores da ordem dos 88, 94.7 e 93.5%, respectivamente. Para os sistemas propostos por Li e Lienhart, não são apresentados valores para a precisão da detecção; contudo, é referido que foram obtidos valores elevados. No sistema proposto por Wolf [Wolf02], a precisão para a detecção foi da ordem dos 34.4%, valor considerado baixo pelo autor e justificado com o facto de existirem muitas estruturas com propriedades semelhantes às do texto que só podem ser distinguidas utilizando as técnicas de reconhecimento, i.e. só podem ser distinguidas pelo algoritmo numa fase posterior à detecção. Os valores anteriormente referidos foram classificados pelos investigadores como muito bons resultados e portanto servirão de referência nesta secção. É de referir que estes valores foram obtidos com conjuntos de teste onde predomina o texto gráfico e não o texto de cena o que, em princípio, facilita a tarefa de extracção de texto.

Nos testes efectuados, obtiveram-se, em termos de detecção de texto, valores para a *recall* na ordem de 94%, factor indicativo de que apenas cerca de 6% do texto não foi detectado. No que respeita aos valores da precisão em termos de detecção de texto, estes andaram também na ordem de 94%, indicando que somente cerca de 6% dos caracteres detectados foram erradamente detectados (falsas detecções).

Fazendo uma análise separada para o texto gráfico e para o texto de cena, pode dizer-se que:

- **Texto gráfico** – Nos vídeos onde predomina o texto gráfico, os resultados obtidos apresentam valores mais elevados para ambas as métricas. Estes são da ordem dos 95 e 96% para a *recall* e precisão, respectivamente. Tais valores indicam que, para texto gráfico, apenas cerca de 5% dos caracteres não foram detectados ou foram falsamente detectados. De modo a efectuar uma análise mais detalhada do desempenho da detecção para texto gráfico apresentam-se na Tabela 4.3 os resultados obtidos considerando separadamente texto gráfico com movimento e texto gráfico fixo.

Tabela 4.3 – Resultados médios obtidos para a detecção de texto gráfico com movimento e texto gráfico fixo.

Tipos de texto	<i>Nº</i> <i>Tramas</i>	<i>Nº</i> <i>caracteres</i>	<i>Recall</i>	<i>Precisão</i>
Texto gráfico com movimento	4110	3986	0.961	0.982
Texto gráfico fixo	8869	940	0.917	0.866
Totalidade do texto gráfico na <i>ground truth</i>	12979	4926	0.953	0.958

Analisando então de uma forma mais detalhada os resultados obtidos para o texto gráfico (ver Tabela 4.3), pode constatar-se que o desempenho é melhor para os vídeos onde predomina o texto com movimento, 96 e 98% para a *recall* e precisão, respectivamente. Este melhor desempenho fica a dever-se ao facto de o movimento diferenciado entre o texto e o fundo facilitar a identificação de falsas detecções. No entanto, nos vídeos onde o texto é fixo, para que as falsas detecções sejam detectadas é necessário que o fundo se movimente, facto que nem sempre acontece;

- **Texto de cena** – Nos vídeos onde o texto de cena predomina, verificou-se um decréscimo significativo dos valores de *recall* e precisão (88 e 84%, respectivamente) em relação aos vídeo onde predomina o texto gráfico. Estes resultados menos bons na métrica *recall* para o texto de cena devem-se, essencialmente, às características do texto de cena que apresenta uma maior variedade de fontes e tamanhos, o que o torna mais difícil de classificar e por consequência de detectar. Os valores mais baixos da métrica precisão, nos vídeos onde predomina o texto de cena, ficam a dever-se ao aumento do número de estruturas que possuem uma forma, posição espacial e movimento semelhantes aos caracteres, o que torna difícil a sua classificação como não texto por parte do algoritmo proposto. O movimento do texto de cena resulta essencialmente do movimento da câmara. Assim, o movimento do texto é semelhante ao do fundo do vídeo o que também dificulta a sua detecção, pois não se consegue tirar partido, como no texto gráfico, do movimento diferenciado entre o texto e o fundo do vídeo para identificar as falsas detecções.

Com o objectivo de avaliar a melhoria de desempenho do algoritmo quando aplicado a sequências de vídeo (onde pode ser feita a análise de movimento) em comparação com o seu desempenho aplicado a imagens, são apresentados na

Tabela 4.4 os resultados da detecção de texto efectuada sobre o conjunto de 60 imagens utilizado para efectuar a avaliação do desempenho do algoritmo de extracção de texto em imagens no Capítulo 3, bem como os resultados obtidos para a detecção de texto efectuada sobre o conjunto de teste de 13 sequências de vídeo. Esta comparação tem interesse, uma vez que a maioria das 60 imagens utilizadas para avaliar o desempenho no Capítulo 3, foram retiradas das 13 sequências de vídeo utilizadas neste Capítulo para efectuar a análise de desempenho para vídeo.

Tabela 4.4 – Resultados médios obtidos para a detecção de todo o texto que faz parte da *ground truth*, quer para o conjunto de teste de 60 imagens, quer para o conjunto de teste das 13 sequências de vídeo.

Tipos de texto	Imagens		Vídeos	
	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão
Texto de cena	0.791	0.895	0.879	0.836
Texto gráfico	0.913	0.909	0.953	0.958
Totalidade do texto na <i>ground truth</i>	0.868	0.904	0.941	0.938

Feita a comparação do desempenho em termos de detecção de texto para as imagens e para o vídeo, para a totalidade do texto na *ground truth*, verifica-se um aumento no caso da detecção efectuada sobre sequências de vídeo tanto para a *recall*, como para a precisão da ordem dos 7.3 e 3.4%, respectivamente. Analisando de forma separada o texto gráfico e o texto de cena, verifica-se o seguinte:

- **Texto gráfico** – Quando o texto é gráfico, constata-se na detecção de texto efectuada em vídeos, um aumento em relação à detecção efectuada em imagens da ordem dos 4.8 e 7.3%, para a *recall* e precisão, respectivamente. O aumento do desempenho do algoritmo na detecção de texto gráfico nos vídeos, resulta essencialmente da exploração da redundância temporal existente no vídeo. Nos vídeos, sobretudo naqueles onde existe movimento, as condições de detecção para um determinado carácter variam de trama para trama, devido ao seu movimento, aumentando a probabilidade do texto ser detectado e logo contribuir deste modo para o aumento da *recall*. Para além disso, também se torna mais fácil diferenciar o texto de outras estruturas com características semelhantes às do texto, pois estas possuem um movimento diferente do movimento do texto, contribuindo assim para o aumento da precisão. Nos vídeos onde o texto está fixo e o fundo não possui movimento, torna-se mais difícil efectuar a detecção, i.e. quando esta falha para uma trama falha para todas, o que coloca os vídeos em pé de igualdade com as imagens em termos de *recall*. A precisão tende a diminuir nesta situação (fundo e texto fixos), pois a diferenciação entre os caracteres e outras estruturas com características semelhantes também é mais difícil de efectuar, uma vez que ambos possuem o mesmo movimento; para além disso, nas tramas de vídeo onde não existe texto acabam sempre por ocorrer falsas detecções as quais contribuem também para a diminuição dos valores de precisão;
- **Texto de cena** – Quando o texto é de cena, verifica-se uma melhoria do desempenho da ordem de 8.8% para a *recall* quando a detecção de texto é efectuada em vídeos, em relação à detecção de texto em imagens; porém, para a precisão tem-se uma diminuição da ordem de 5.9%. O aumento na *recall*, tal como foi referido para o texto gráfico, deve-se essencialmente à exploração da redundância temporal existente no vídeo. A diminuição da precisão, tal como para o texto gráfico, está associada em grande parte às falsas detecções que ocorrem nas tramas sem texto.

No capítulo anterior foram apresentados alguns exemplos de imagens onde a detecção de texto era particularmente difícil para o algoritmo proposto. Neste capítulo serão também

apresentados exemplos de situações onde o algoritmo proposto tem dificuldade em efectuar o seguimento do texto, o que vai condicionar a sua detecção. Assim, durante a análise de resultados foi identificada uma situação crítica em termos do seguimento correcto do texto e que está relacionada com o movimento do texto: a Figura 4.20 ilustra esta dificuldade. Sempre que o movimento do texto sofre variações grandes, quer na sua direcção, quer na sua velocidade, o algoritmo apresenta dificuldades em efectuar o seu seguimento correctamente. Estas dificuldades derivam do facto do algoritmo estimar a posição do texto para a trama $n+1$ com base na informação recolhida sobre o movimento do texto, direcção e velocidade, até à trama n . O exemplo apresentado na Figura 4.20 corresponde a uma sequência de vídeo onde o texto varia a direcção do seu movimento de forma substancial. O texto possui movimento de *scroll* com deslocamento para a direita até à trama n e a partir dessa trama passa a deslocar-se para a esquerda. Nesta situação, o algoritmo gera duas cadeias de caracteres para o mesmo carácter, uma para quando o texto se desloca para a esquerda e outra quando o texto se desloca para a direita, dando origem a uma dupla detecção, como ilustrado na Figura 4.20 b).

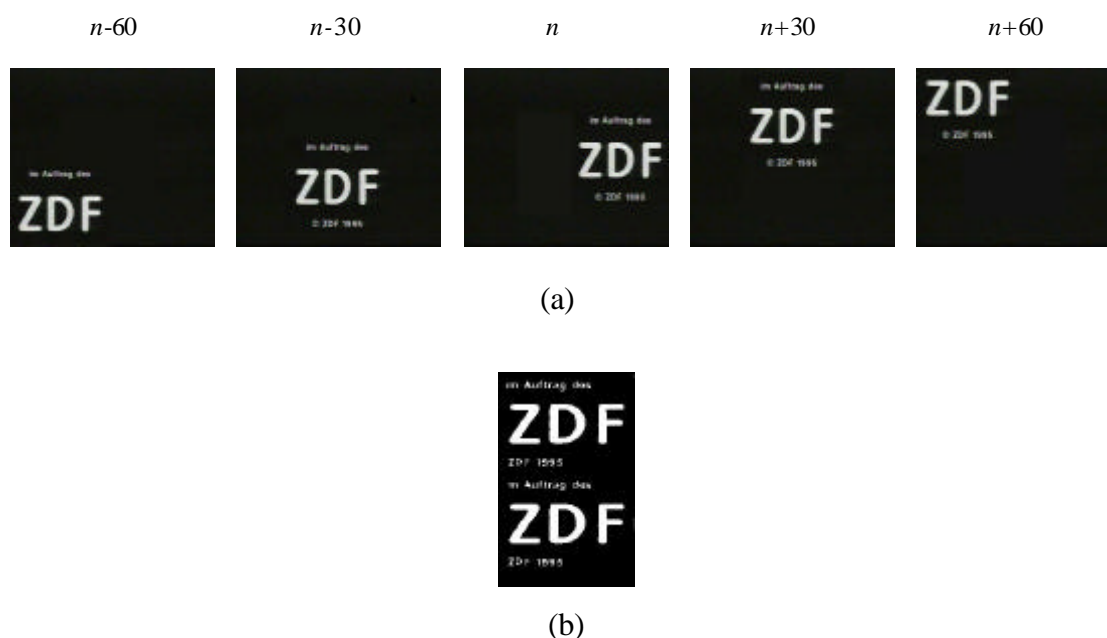


Figura 4.20 – Exemplo de falha no seguimento do texto devido a alterações fortes na direcção do movimento do texto: (a) sequência de texto; (b) resultado da detecção de texto.

Os algoritmos estudados no Capítulo 2 também apresentam dificuldades para efectuar o seguimento de texto sempre que este apresenta movimento com variações grandes, quer na sua direcção, quer na sua velocidade. Os seus autores minimizam esta desvantagem considerando que o texto mais relevante para os vários tipos de aplicações é maioritariamente gráfico; no texto gráfico, o movimento predominante é o movimento rectilíneo com velocidade constante.

4.4.3.2 Avaliação do Reconhecimento de Texto

Na avaliação do desempenho em termos de reconhecimento de texto são utilizadas as imagens resultantes da integração do texto detectado nas várias tramas que compõem cada

sequência de texto do vídeo. Por exemplo, na Figura 4.21 (b) ilustra-se a imagem resultante da integração do texto existente na sequência apresentada na Figura 4.21 (a).

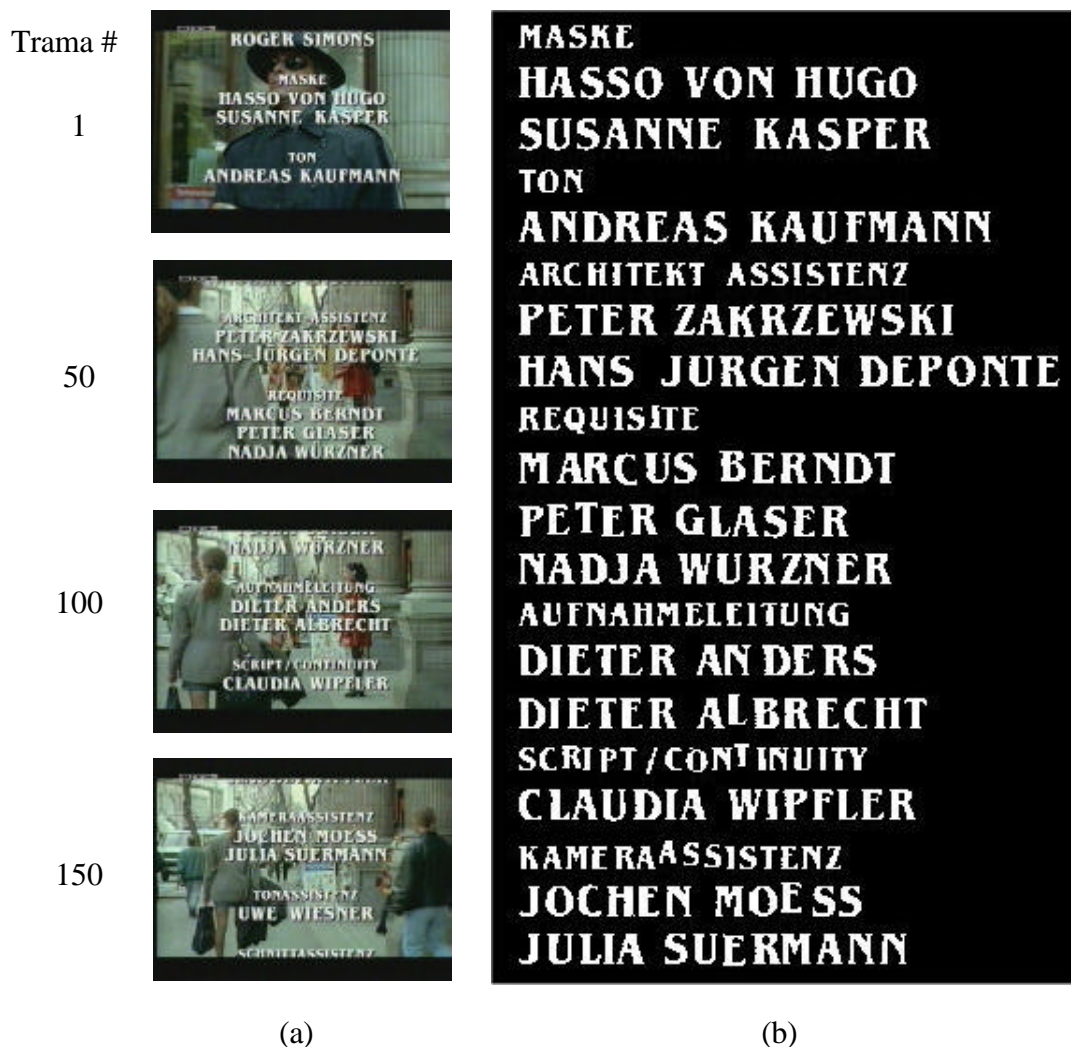


Figura 4.21 – Exemplo da imagem resultante da integração do texto existente numa sequência de vídeo: (a) sequência de vídeo; (b) imagem resultante da integração do texto existente na sequência de vídeo.

Antes de processar as imagens com o sistema OCR comercial escolhido, a sua resolução é aumentada para 100 ppi (*pixels per inch*), uma vez que os sistemas OCR comerciais estão preparados para efectuar o reconhecimento de caracteres em imagens com elevada resolução (superior a 75 ppi no caso do OmniPage Pro 12.0). No caso do sistema OCR desenvolvido por Lienhart [Lienhart95], não é necessário aumentar a resolução. De seguida, as imagens resultantes da integração são processadas e, para cada uma delas, determina-se por inspecção humana se cada carácter detectado é reconhecido correctamente ou não. Os resultados obtidos na avaliação do desempenho de reconhecimento de todo o texto que faz parte da *ground truth*, i.e. texto horizontal, inclinado e vertical, para os vários tipos de texto e para a totalidade das imagens, podem ser observados na Tabela 4.5.

Tabela 4.5 – Resultados médios obtidos para o reconhecimento de todo o texto na *ground truth* para a totalidade dos vídeos.

Tipos de texto	OCR [Lienhart95]		OCR OmniPage Pro 12.0	
	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão
Texto de cena	0.653	0.670	0.803	0.811
Texto gráfico	0.755	0.767	0.920	0.922
Totalidade do texto na <i>ground truth</i>	0.739	0.749	0.902	0.905

De forma semelhante ao que se verificou para a detecção, também para o reconhecimento o desempenho do algoritmo proposto pode considerar-se bastante bom já que se obtiveram para as métricas de *recall* e precisão valores que são considerados, por outros investigadores, como sendo muito bons. No reconhecimento de texto gráfico em sequências de vídeo Li, Lienhart e Wolf em [Li02, Lienhart02, Wolf02] obtiveram para a métrica *recall* valores da ordem dos 88, 79.6 e 86.4%, respectivamente. Para a métrica precisão, Lienhart e Wolf apresentam valores da ordem dos 88.0 e 89.6%, respectivamente. Li [Li02] não disponibiliza valores para a precisão, mas refere que esta foi elevada.

No reconhecimento do texto que faz parte da *ground truth*, tal como na detecção, também se verificou um melhor desempenho para os vídeos cujo texto predominante é gráfico, tanto para a *recall* como para a precisão. Nos testes efectuados com o sistema OCR OmniPage Pro 12.0 obtiveram-se, em termos de reconhecimento de texto, valores para *recall* da ordem de 90%, valor indicativo de que apenas cerca de 10% do texto considerado relevante, segundo as condições atrás definidas, não foi reconhecido. No que respeita aos valores da precisão, estes andaram também na ordem de 90%, indicando que somente cerca de 10% dos caracteres foram falsamente reconhecidos. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], obtiveram-se valores inferiores tanto para *recall* como para a precisão, da ordem dos 74 e 75%, respectivamente. Estes valores indicam que cerca de 26% do texto considerado relevante, segundo as condições atrás definidas, não foi reconhecido e cerca de 25% dos caracteres foram falsamente reconhecidos.

Tal como já foi referido no capítulo anterior, a diferença de desempenho evidenciada pelos dois sistemas OCR resulta, essencialmente, do reino limitado efectuado à base de dados utilizada pelo sistema OCR desenvolvido por Lienhart e Stuber [Lienhart95]. Deste modo, o treino da base de dados com poucos tipos de fontes penaliza muito este sistema OCR, sobretudo no reconhecimento de texto de cena onde o texto apresenta uma maior diversidade de fontes, estilos e tamanhos.

Efectuada uma análise mais detalhada para o desempenho do reconhecimento, verifica-se que os resultados obtidos são melhores para os vídeos onde o texto gráfico predomina comparativamente aos vídeos onde predomina o texto de cena, o que era expectável. Assim, fazendo uma análise separada para o texto gráfico e para o texto de cena, pode dizer-se que:

- **Texto gráfico** – Nos vídeos onde o texto gráfico predomina, a *recall* e a precisão apresentam para o OCR OmniPage Pro 12.0 valores de cerca 92%. Tais valores indicam que, para texto gráfico apenas, cerca de 8% dos caracteres não foram reconhecidos ou

foram erradamente identificados. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], a *recall* e a precisão apresentam valores de cerca 76 e 77%, respectivamente. Tais valores indicam que, para texto gráfico, cerca de 24-23% dos caracteres não foram reconhecidos ou foram erradamente identificados;

- **Texto de cena** – Nos vídeos onde o texto de cena é o texto predominante, verificou-se um decréscimo dos valores de *recall* e precisão em relação ao vídeos onde predomina o texto gráfico, sendo estes valores da ordem de 80 % utilizando o OCR OmniPage Pro 12.0. Estes resultados indicam que, para texto de cena, cerca de 20% dos caracteres não foram reconhecidos ou foram falsamente identificados. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], verificou-se igualmente um decréscimo da *recall* e da precisão em relação ao texto gráfico. Estas apresentam agora valores de cerca de 65-67%, respectivamente; tais valores indicam que cerca de 35% dos caracteres não foram reconhecidos ou foram erradamente identificados.

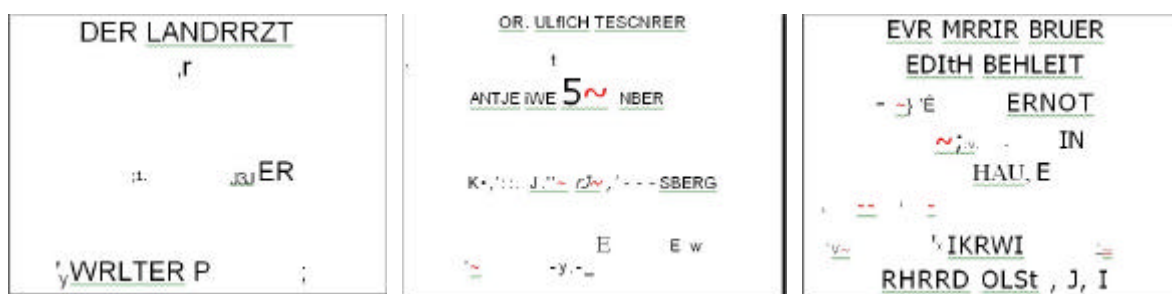
Para ambos os sistemas OCR utilizados, o desempenho em termos de reconhecimento apresenta melhores valores para o texto gráfico, quer para a *recall*, quer para a precisão. Este melhor desempenho de reconhecimento nos vídeos onde o texto gráfico predomina, deve-se, tal como para a detecção de texto em imagens apresentado no capítulo anterior, ao facto das fontes utilizadas corresponderem, usualmente, a fontes bem conhecidas dos sistemas OCR, tais como Arial, Courier e Times New Roman. O reconhecimento do texto de cena é penalizado, essencialmente devido ao elevado número de fontes diferentes que existem no mesmo ou fontes que foram mais ou menos alteradas.

De forma semelhante ao que foi efectuado para o algoritmo de extracção de texto em imagens e com vista a avaliar o desempenho do algoritmo desenvolvido para extrair texto em sequências de vídeo de forma mais objectiva, foram efectuados testes que comparam o desempenho do OCR OmniPage Pro 12.0 usado isoladamente e em conjunto com o algoritmo de detecção de texto proposto neste capítulo. Para isso, foi efectuado o reconhecimento de todo o texto que faz parte da *ground truth* do conjunto de vídeos de teste, utilizando unicamente o OCR OmniPage Pro 12.0 ou seja dando-lhe directamente as imagens originais sem qualquer processamento e utilizando o algoritmo de detecção de texto proposto em conjunto com o OCR OmniPage Pro 12.0.

Para efectuar o reconhecimento do texto nos vídeos com o OmniPage Pro 12.0, foram analisadas individualmente um conjunto de tramas que incluem todo o texto existente nesse vídeo. Os resultados obtidos podem ser observados na Tabela 4.6. Na Figura 4.22 é apresentado um exemplo que ilustra todo este processo. A Figura 4.22 (a) ilustra as imagens que incluem todo o texto da sequência e a Figura 4.22 (b) ilustra o resultado da detecção efectuada pelo OmniPage Pro 12.0.



(a)



(b)

Figura 4.22 – Exemplo de reconhecimento de texto com o OmniPage Pro 12.0: (a) tramas representativas do texto existente no vídeo; (b) resultados do reconhecimento efectuada pelo OmniPage Pro 12.0 para as tramas em (a)⁴.

Tabela 4.6 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da *ground truth*, utilizando unicamente o OCR OmniPage Pro 12.0 e utilizando o algoritmo de detecção de texto proposto em conjunto com o OCR OmniPage Pro 12.0.

Tipos de texto	OCR OmniPage Pro 12.0		Algoritmo de Detecção de Texto + OCR OmniPage Pro 12.0	
	Recall	Precisão	Recall	Precisão
Texto de cena	0.544	0.561	0.803	0.811
Texto gráfico	0.440	0.643	0.920	0.922
Totalidade do texto	0.456	0.626	0.902	0.905

⁴ Os “~” na Figura 4.22 (b) correspondem a regiões que verificam os requisitos para serem reconhecidas pelo sistema OCR como sendo um carácter, mas que o OCR não reconhece porque não fazem parte da sua base de dados. O sistema OCR poderá ser treinado para reconhecer esse tipo de regiões.

A utilização conjunta do algoritmo de detecção de texto proposto e do OCR OmniPage Pro 12.0 permite realizar uma melhor separação entre o texto e o fundo complexo da imagem. Como se pode ver na Tabela 4.6, esta separação permite um aumento muito significativo da *recall* do sistema OCR de 54.4% para 80.3%, nos vídeos onde predomina o texto de cena, e de 44% para 92%, nos vídeos onde predomina o texto gráfico. Em termos globais, a *recall* do OmniPage Pro 12.0 aumenta também substancialmente de 45.6% para 90.2%. Em termos de precisão, o aumento de desempenho é menor mas ainda assim muito significativo uma vez que há um aumento global de 62.6% para 90.5%. O aumento da precisão é menor quando comparado com o aumento da *recall*, porque se parte de um valor mais elevado. Note-se no entanto que o valor final é mais ou menos idêntico.

De forma semelhante ao que se fez na avaliação da detecção, também para a avaliação do reconhecimento é feita uma comparação do desempenho do algoritmo para imagens e vídeos. Com esta comparação pretende-se avaliar o aumento de desempenho do algoritmo quando aplicado a sequências de vídeo (onde pode ser feita a análise de movimento) em comparação com o seu desempenho aplicado a imagens. Na Tabela 4.7 são apresentados os resultados do reconhecimento de texto, efectuada sobre o conjunto de 60 imagens utilizadas para avaliar o desempenho do algoritmo no Capítulo 3, bem como os resultados obtidos para o reconhecimento de texto efectuada sobre o conjunto de teste de 13 sequências de vídeo.

Tabela 4.7 – Resultados médios obtidos para o reconhecimento de todo o texto que faz parte da *ground truth*, quer para o conjunto de teste de 60 imagens, quer para o conjunto de teste das 13 sequências de vídeo.

Tipos de texto	OCR [Lienhart95]				OCR OmniPage Pro 12.0			
	Imagens		Vídeos		Imagens		Vídeos	
	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão	<i>Recall</i>	Precisão
Texto de cena	0.509	0.590	0.653	0.670	0.698	0.815	0.803	0.811
Texto gráfico	0.728	0.721	0.755	0.767	0.902	0.919	0.920	0.922
Totalidade do texto	0.647	0.677	0.739	0.749	0.826	0.884	0.902	0.905

Comparando os resultados obtidos no reconhecimento de texto em imagens e em vídeo efectuada sobre a totalidade do texto na *ground truth*, verificou-se um aumento do desempenho no caso do reconhecimento em sequências de vídeo, tanto para a *recall*, como para a precisão da ordem dos 7.6 e 2.1% utilizando o OCR OmniPage Pro 12.0, e de 9.2 e 7.2% utilizando o OCR desenvolvido por Lienhart [Lienhart95], respectivamente. Analisando de forma separada o texto gráfico e o texto de cena, verificou-se o seguinte:

- **Texto gráfico** – Para texto gráfico e utilizando o OCR OmniPage Pro 12.0, constata-se no reconhecimento de texto em vídeos, um aumento em relação ao reconhecimento em imagens da ordem dos 2.0 e 0.3%, para a *recall* e precisão, respectivamente. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], existe igualmente um aumento para os vídeos da *recall* e da precisão da ordem dos 2.7 e 4.6%, respectivamente. O aumento do desempenho do algoritmo no reconhecimento de texto gráfico nos vídeos, resulta essencialmente da exploração da redundância temporal

existente no vídeo efectuada na fase de detecção para eliminar falsas detecções que originam normalmente falsos reconhecimentos. O aumento menos significativo da precisão sobretudo para o OCR OmniPage Pro 12.0 deve-se sobretudo às falsas detecções nas tramas onde não existe texto;

- **Texto de cena** – Para texto de cena e utilizando o OCR OmniPage Pro 12.0 para efectuar o reconhecimento em vídeos, verifica-se um aumento na *recall* da ordem de 8.8% e uma diminuição da precisão da ordem de 0.4%, quando comparado com o reconhecimento em imagens. O aumento na *recall*, tal como foi referido para o texto gráfico, deve-se essencialmente à exploração da redundância temporal existente no vídeo. A diminuição da precisão, está associada em grande parte às falsas detecções que ocorrem nas tramas sem texto; é de referir que no conjunto das 60 imagens não existem imagens sem texto. Com a utilização do OCR desenvolvido por Lienhart [Lienhart95], existe um aumento para os vídeos da *recall* e da precisão da ordem dos 14.4 e 8.0%, respectivamente.

Este aumento nos valores do reconhecimento para ambas as métricas era expectável uma vez que no processamento do vídeo se consegue explorar o movimento do texto e/ou do fundo para melhorar o desempenho do algoritmo de extracção de texto. Contudo, também devido às características dos conteúdos usados, este aumento não é tão elevado como se poderia esperar.

4.5 Comentários Finais

Ao longo deste capítulo foi proposto um algoritmo que permite fazer a extracção de texto em vídeo, tanto para texto gráfico, como para texto de cena. O texto pode ser constituído por caracteres de vários tamanhos, fontes e cores e aparecer segundo qualquer direcção. O método proposto é uma evolução do método proposto no capítulo anterior para efectuar a extracção de texto em imagens ou tramas de vídeo, i.e. foi considerada neste último a componente temporal, a qual permite explorar a redundância existente no vídeo para melhorar a detecção do texto. Esta melhoria pode efectuar-se, quer através da eliminação de regiões que não sejam consistentes no tempo ou que possuam um movimento diferente do movimento do texto, quer através da recuperação por interpolação de caracteres que fazem parte do texto mas não foram detectados. Assim, o método de extracção de texto proposto começa por efectuar a detecção das sequências de texto existentes nos vídeos. Para tal, algumas tramas seleccionadas de forma periódica são segmentadas em regiões conexas que são, posteriormente, filtradas de acordo com várias restrições com vista a detectar a existência, ou não, de texto. As restrições impostas actuam ao nível do contraste, da forma e da localização espacial e o seu objectivo prende-se com a eliminação de regiões que não correspondam a texto. Uma vez detectado o início e o fim das sequências de texto, é aplicada a análise de movimento a cada uma das sequências de texto anteriormente identificadas. Na análise do movimento é explorada a redundância temporal existente no vídeo com vista a melhorar a detecção de texto através da eliminação dos caracteres que não sejam consistentes no tempo, bem como da recuperação de caracteres que não foram detectados. Esta análise permite ainda determinar, com precisão, o início e o fim de cada palavra.

Após a implementação do método proposto para a extracção de texto em sequências de vídeo, e de forma semelhante ao efectuado para o algoritmo de extracção de texto em imagens, fez-se o seu teste utilizando vários tipos de vídeos retirados de genéricos de filmes, noticiários, anúncios comerciais e eventos desportivos. Os resultados foram analisados, tendo-se

verificado que nos vídeos onde o texto possui um movimento diferente do movimento do fundo da imagem e onde tipicamente predomina o texto gráfico, o algoritmo tem melhor desempenho, tanto para a *recall* como para a precisão. No reconhecimento do texto, tal como na detecção, também se verificou para o texto gráfico um melhor desempenho tanto para a *recall* como para a precisão.

O menor desempenho do algoritmo na detecção de texto de cena deve-se às características deste tipo de texto: grande diversidade de fontes, estilos, tamanhos e orientações. Essas características tornam-no mais difícil de detectar pelo algoritmo proposto, tanto na fase de segmentação, como na de classificação. Na fase de seguimento também se torna mais difícil fazer a eliminação das falsas detecções pois o movimento do texto de cena é tipicamente semelhante ao movimento do fundo. Em termos de reconhecimento, também o texto de cena é mais difícil de processar por parte dos sistemas OCR utilizados devido à sua complexidade estrutural. Desta forma, os resultados obtidos pelo algoritmo proposto para sequências de vídeo onde predomina o texto de cena tornam-se menos bons quando comparados com os resultados para os vídeos onde o texto predominante é o gráfico. Pode, contudo, concluir-se que o algoritmo proposto apresenta resultados muito satisfatórios, tanto para a detecção como para o reconhecimento, quer para texto gráfico, quer para texto de cena, nomeadamente semelhantes aos que foram obtidos em [Li02, Lienhart02, Wolf02]. Este facto parece indicar que os valores adoptados para os parâmetros de configuração, nomeadamente na fase de segmentação e de análise de contraste e de geometria, são adequados; como é evidente, poderiam usar-se valores mais adequados se se conhecessem à partida algumas características particulares do texto a processar.

Capítulo 5

Sumário e Trabalho Futuro

Como foi dito no Capítulo 1, a maior facilidade em adquirir, processar, armazenar e transmitir informação audiovisual veio acentuar a necessidade de desenvolver ferramentas para fazer o processamento da mesma, com vários objectivos, consoante o ambiente de aplicação. Para além disso, o facto dos conteúdos criados terem começado a ser enriquecidos com componentes multimédia, tais como imagens, vídeos e áudio, originou o aumento substancial da dimensão das bibliotecas para os guardar. Neste contexto, a informação textual, existente nas imagens e nos vídeos, é uma fonte de informação com um elevado valor semântico em termos de pesquisa, desde que esse mesmo texto esteja disponível. Para isso, o texto deve poder ser detectado e reconhecido automaticamente de modo a que possa ser utilizado para indexação e procura nas bibliotecas de imagem e vídeo. Actualmente, a tecnologia utilizada pelos sistemas OCR disponíveis no mercado apresenta dificuldades no reconhecimento de texto impresso sobre fundos com texturas complexas que ocorrem com muita frequência nos vídeos e nas imagens, sobretudo quando a cores.

A extracção de texto em imagens e sequências de vídeo é um problema complexo para o qual não existe uma técnica perfeita para todos os tipos de conteúdos e situações e cuja solução passa, muitas vezes, pela combinação de várias técnicas, aproveitando as vantagens de cada uma, de modo a obter uma solução adequada às necessidades das várias aplicações. Neste contexto, o objectivo desta Tese, ou seja a extracção automática e eficaz de texto, quer em imagens, quer em sequências de vídeo, visa contribuir para o desenvolvimento de técnicas que permitam implementar sistemas automáticos e eficientes, de descrição, indexação e procura de conteúdos multimédia. Neste sentido, e de acordo com os grandes objectivos desta Tese definidos no Capítulo 1, cada um dos capítulos da mesma contribuiu para rever, estudar, aplicar e desenvolver conceitos e métodos relacionados com a extracção automática de texto em sequências de vídeo, de modo a superar as dificuldades evidenciadas pelos sistemas OCR no reconhecimento do texto nas situações descritas.

A Tese foi organizada em 5 capítulos que são, de seguida, sumariados.

No Capítulo 1 foi efectuada a contextualização do tema, analisada a sua importância no universo onde se insere e ainda definidos os objectivos a atingir com esta Tese.

O Capítulo 2 começou por definir uma arquitectura genérica para o mecanismo de extracção de texto em imagens e vídeo. Ao longo do mesmo fez-se, ainda que sumariamente, a apresentação de vários sistemas e técnicas disponíveis e relevantes e que oferecem soluções para os vários módulos da arquitectura básica de extracção de texto apresentada. Estas técnicas permitem efectuar a segmentação, a classificação, o seguimento e o reconhecimento das várias regiões de uma imagem ou vídeo como texto ou não. Para as várias técnicas foi ainda efectuada uma análise comparativa e apresentadas as vantagens e desvantagens de cada uma delas. Ainda neste capítulo, foram descritos os sistemas mais representativos disponíveis na literatura para a extracção de texto em imagens e sequências de vídeo, bem como as várias técnicas que cada um utiliza para efectuar a extracção de texto. Para tal, e para cada um dos sistemas apresentados, foi analisada a sua arquitectura básica, as suas vantagens e desvantagens, e ainda as limitações impostas aos conteúdos a analisar e ao texto a extrair.

O Capítulo 3 apresentou o mecanismo de extracção automática de texto em imagens desenvolvido no âmbito desta Tese. Este mecanismo explora, principalmente, o contraste existente entre o texto e o fundo da imagem, bem como a forma e a distribuição espacial dos caracteres. Foram propostas soluções melhoradas tanto para a segmentação, como para a detecção de caracteres: para ambos estes módulos, partiu-se de técnicas conhecidas, tendo-se introduzido melhorias de modo a alargar a sua gama de aplicação e a melhorar o seu desempenho na detecção de texto. De entre estas melhorias, fazem parte técnicas para melhorar a precisão das fronteiras das regiões conexas detectadas na segmentação e técnicas que permitem melhorar a eficiência da detecção de caracteres com base na análise do contraste, nomeadamente em imagens pouco contrastadas. Foi também proposto um filtro que combina a detecção de fronteiras com um filtro de mediana de modo a diminuir a influência de alguns efeitos indesejáveis nas imagens ao mesmo tempo que preserva as zonas de elevado contraste (normalmente correspondentes a regiões de texto). Foram ainda propostas neste capítulo técnicas que permitem tanto detectar palavras com inclinações compreendidas entre 0 – 90°, como efectuar a sua rotação para a horizontal de modo a poderem ser reconhecidas por sistemas OCR. A aplicação desenvolvida permitiu avaliar o desempenho do mecanismo de extracção de texto proposto. Permitiu também efectuar a avaliação do desempenho de sistemas OCR comerciais no reconhecimento de texto em imagens com fundos complexos de uma forma isolada e em conjunto com o mecanismo proposto.

O Capítulo 4 apresentou o mecanismo de extracção automática de texto em vídeos desenvolvido no contexto desta Tese. Neste mecanismo adicionou-se ao algoritmo desenvolvido no capítulo anterior, a dimensão temporal, que é uma característica intrínseca do vídeo. Para tal, foi explorada a redundância temporal existente nos vídeos com vista a aumentar a probabilidade de detecção de texto, remover detecções falsas em tramas individuais e recuperar palavras que não foram ‘acidentalmente’ detectadas em algumas tramas individuais. Assim, foram propostas soluções que permitem efectuar o seguimento dos caracteres de texto ao longo do tempo e recuperar falhas de detecção em tramas individuais. As técnicas de seguimento de caracteres propostas baseiam-se na comparação de tramas sucessivas, i.e. relacionam o resultado do momento anterior com o do momento actual, através da comparação de uma assinatura calculada para cada carácter, formada por características tais como a cor, o tamanho, o deslocamento, etc.. Para recuperar os caracteres em falta, foram propostas técnicas que utilizam a informação de deslocamento dos caracteres para fazerem a sua interpolação. De forma semelhante ao que foi efectuada no capítulo anterior para a detecção de texto em imagens, também para o vídeo a aplicação desenvolvida

permitiu avaliar o desempenho do mecanismo de extracção de texto, assim como de sistemas OCR no reconhecimento de texto em vídeos com fundos complexos de uma forma isolada e em conjunto com o mecanismo de extracção proposto.

Apesar de todo o trabalho desenvolvido para esta Tese no âmbito da extracção de texto, muito resta, ainda, para ser feito nesta área. Em termos de trabalho futuro, os métodos propostos poderiam ainda ser melhorado através da inclusão de:

- Técnicas que permitam extrair texto de pequenas dimensões, eventualmente através de um pré-processamento que permita identificar ainda que grosseiramente as zonas onde existe texto, seguido de um aumento da resolução, somente das regiões de texto;
- Aperfeiçoamento de técnicas para a detecção do texto de cena, nomeadamente técnicas de classificação que possibilitem classificar texto ornamentado com efeitos especiais típicos do texto de cena, como por exemplo, texto com sombra, texto tridimensional e texto com os mais variados formatos numa única palavra;
- Aperfeiçoamento da técnica para efectuar a detecção de texto inclinado de modo a diminuir o número de falsas detecções para este tipo de texto;
- Aperfeiçoamento das técnicas de seguimento de modo a permitirem efectuar o seguimento de texto com variações fortes no seu movimento, tanto em velocidade como em direcção.

Espera-se que estas sugestões possam futuramente vir a ser implementadas, contribuindo assim para mais um passo no desenvolvimento de métodos de extracção de texto mais potentes, flexíveis e robustos.

Bibliografia

- [Aach93] T. Aach e A. Kaup, “Statistical Model-Based Change Detection in Moving Video”, *Signal Processing*, Vol. 31, N° 2, pp. 165 – 180, Março 1993.
- [Adiv85] G. Adiv, “Determining Three-Dimensional Motion and Structure from Optical Flow Generated by Several Moving Objects”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 7, N° 4, pp. 384 – 401, Julho 1985.
- [Antonini94] M. Antonini, T. Gaidon, P. Mathieu e M. Barlaud, “Wavelet Transform and Image Coding”, *Advances in Image Communication*, Elsevier, Vol. 5, 1994.
- [Belongie97] S. Belongie, C. Carson, H. Greenspan e J. Malik, “Recognition of Images in Large Databases Using a Learning Framework” *Computer Science Division, University of California at Berkeley, Technical Report 94720* – 1997.
- [Bertini01] M. Bertini, C. Colombo e A. Del Bimbo, “Automatic Caption Localization in Videos Using Salient Points”, *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, Tóquio – Japão, Agosto 2001.
- [Bimbo99] A. D. Bimbo, “Visual Information Retrieval”, *Morgan Kaufmann Publishers, Inc.*, 1999.
- [Bober99] M. Bober, “Performance Evaluation of the CSS Shape Descriptor”, Doc. *ISO/IEC JTC1/SC29/WG11/M4731*, Vancouver MPEG Meeting – Canadá, Julho 1999.
- [Broida86] T. J. Broida, R. Chellappa, “Estimation of Object Motion Parameters from Noisy Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.8, N° 1, pp. 90 – 99, Janeiro 1996.

- [Canny86] J. Canny, “A Computacional Approach to Edge Detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 8, Nº 6, pp. 679 – 698, Novembro 1986.
- [Chalom95] E. Chalom e V. Bove, “Segmentation of Frames in a Video Sequence Using Motion and Other Attributes”, in “Digital Video Compression: Algorithms and Technologies”, SPIE Vol. 2419, pp. 230 – 241, 1995.
- [Chen01] X. Chen e H. Zhang, “Text Area Detection from Video Frames”, *Microsoft Research China*, 2001.
- [Choi97] J. G. Choi, S. Lee e S. Kim, “Spatio-Temporal Segmentation Using a Joint Similarity Measure”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, Nº 2, pp. 279 - 286, Abril 1997.
- [Correia02] P. Correia, “Video Analysis for Object-Based Coding and Description”, Instituto Superior Técnico, Lisboa, Tese de Doutoramento, Dezembro 2002.
- [Cortez95] D. Cortez, P. Nunes, M. Sequeira e F. Pereira, “Image Segmentation Towards New Image Representation Methods”, *Signal Processing: Image Communication*, Vol. 6, Nº 6, pp. 485 – 498, Fevereiro 1995.
- [Cover67] T. M. Cover e P. E. Hart. “Nearest Neighbor Pattern Classification” *IEEE Trans. on Information Theory*, Vol. IT-13, Nº 1, pp. 21 – 27, Janeiro 1967.
- [Crandall01] D. Crandall e R. Kasturi, “Robust Detection of Stylized Text Events in Digital Video”, *Proceedings of the International Conference on Document Analysis and Recognition*, Seattle – EUA, pp. 865 – 869, Setembro 2001.
- [Eikvil93] L. Eikvil, “Optical Character Recognition”, <http://www.nr.no/bild/DocOnline.html>, Dezembro 1993.
- [Etemad97] K. Etemad, D. Doermann e R. Chellappa, “Multiscale Segmentation of Unstructured Document Pages Using Soft Decision Integration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, Nº 1, pp. 92 – 96, Janeiro 1997.
- [Fan01] J. Fan e D. K. Y. Yau, “Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing”, *IEEE Transactions on Image Processing*, Vol. 10, Nº 10, pp. 1454 – 1466, Outubro 2001.
- [Fletcher88] L. A. Fletcher e R. Kasturi, “A Robust Algorithm for Text String Separation from Mixed Text/Graphics Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, Nº 6, pp. 910 – 918, Novembro 1988.
- [Gagalowicz85] A. Gagalowicz e S. Ma, “Sequential of Nature Textures”, *Computer Vision Graphics and Image Processing*, Vol. 30, Nº 9, pp. 289 – 315, Setembro 1985.
- [Garduno94] V. G. Garduño e C. Labit, “On the Tracking of Regions Over Time for Very Low Bit Rate Image Sequence Coding”, *Picture Coding Symposium*, Sacramento – EUA, 1994.

- [Gil96] S. Gil e R. Milanese, Thierry Pun, “Combining Multiple Motion Estimates for Vehicle Tracking”, *4th European Conference on Computer Vision*, Cambridge – Reino Unido, Abril 1996.
- [Gonzalez93] R. C. Gonzalez e R. E Woods, “Digital Image Processing”, *Addison-Wesley Publishing Company*, 1993.
- [Gu01] L. Gu, “Text Detection and Extraction in MPEG Video Sequences”, *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, Brescia – Itália, pp. 233 – 240, Setembro 2001.
- [Haralick85] R. M. Haralick e L. G. Shapiro “Survey: Image Segmentation Techniques”, *Computer Vision Graphics and Image Processing*, Vol. 29, pp 100 - 132, 1985.
- [Haralick92] R. Haralick e L. Sharpiro, “Computer and Robot Vision”, *Vol. I, Addison-Wesley Pub. Company*, 1992.
- [Haralick94] R. M. Haralick e L. G. Shapiro “Glossary of Computer Vision Terms”, in “Digital Image Processing Methods”, editado por *E. Dougherty, Dekker*, pp. 415 – 467, 1994.
- [Hasan00] Y. M. Y. Hasan e L. J. Karam, “Morphological Text Extraction from Images”, *IEEE Transactions on Image Processing*, Vol. 9, Nº 11, pp. 1978 – 1983, Novembro 2000.
- [Hase01] H. Hase, T. Shinokawa, M. Yoneda e C. Y. Suen, “Character String Extraction from Color Documents”, *Pattern Recognition*, Vol. 34, Nº 7, pp. 1349 – 1365, Julho 2001.
- [Heidegger29] M. Heidegger, “Da Essência do Fundamento”, Halle, 1929.
- [Horowitz72] S.L. Horowitz e T. Pavlidis, “Picture Segmentation by a Transversal Algorithm”, *Computer Graphics and Image Processing*, Vol. 1, pp 360 - 372, 1972.
- [Hötter88] M. Hötter e R. Thoma, “Image Segmentation Based on Object Oriented Mapping Parameter Estimation”, *Signal Processing*, Vol. 15, Nº 3, pp. 315 – 348, Outubro 1988.
- [IBMRe99] IBM, “Technical Summary of Turning Angle Shape Descriptors Proposed by IBM”, Doc. *ISO/IEC JTC1/SC29/WG11/P162*, Lancaster *MPEG Meeting* – Reino Unido, Fevereiro 1999.
- [Jähne97] B. Jähne, “Digital Image Processing. Concepts, Algorithms, and Scientific Applications”, *Springer*, 1997.
- [Jain89] A. K. Jain, “Fundamentals of Digital Image Processing”, *Prentice-Hall*, 1989.
- [Jain98] A. K. Jain e B. Yu “Automatic Text Location in Images and Video Frames”, *Pattern Recognition*, Vol. 31, Nº 12, pp. 2055 – 2076, Dezembro 1998.
- [Li98] H. Li e D. Doermann, “Automatic Text Tracking in Digital Video”, <http://documents.cfar.umd.edu/LAMP/Media/Projects/TextTrack/>

- [Li00] H. Li, D. Doermann e O. Kia, “Automatic Text Detection and Tracking in Digital Video”, *IEEE Transactions on Image Processing*, Vol. 1, Nº 1, pp. 147 – 156, Janeiro 2000.
- [Li02] H. Li e D. Doermann “Text Enhancement in Digital Video Using Multiple Frame Integration”, <http://documents.cfar.umd.edu/LAMP/Media/Publications/>, 2002
- [Lienhart95] Rainer Lienhart e F. Stuber, “Automatic Text Recognition for Video Indexing”, http://Eratosthenes.informatik.uni-mannheim.de/informatik/pi4/projects/MoCA/MoCA_TextRecognition/
- [Lienhart00] R. Lienhart e W. Effelsberg, “Automatic Text Segmentation and Text Recognition for Video Indexing”, *Multimedia Systems*, Vol. 8, , Nº 1, pp. 69 – 81, Janeiro 2000.
- [Lienhart02] R. Lienhart e A. Wernicke, “Localizing and Segmenting Text in Images and Videos”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 12, Nº 4, pp. 256 – 268, Abril 2002.
- [Lindsay91] P. H. Lindsay e D. A. Norman, “Introduction into Psychology – Human Information Reception and Processing”, *Springer – Verlag*, 1991.
- [Liu94] J. Liu e Y. H. Yang, “Multiresolution Color Image Segmentation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Nº 7, pp. 689 - 700, Julho 1994.
- [Liu02] F. Liu, X. Song, Y. Luo e D. Hu, “Unsupervised Mumford-Shah Energy Based Hybrid of Texture and NonTexture Image Segmentation”, *IEEE International Conference on Image Processing, Rochester, Nova Iorque – EUA*, Setembro 2002.
- [Khotanzad90] A. Khotanzad e Y. H. Hong, “Invariant Image Recognition by Zernike Moments”, *IEEE Transactions on Pattern and Machine Intelligence*, Vol. 12, Nº 5, pp. 489 – 498, Julho 1990.
- [Kim99] M. Kim, J. Choi, D. Kim, H. Lee, M. Lee, C. Ahn e Y. Ho, “A VOP Generation Tool Automatic Segmentation of Moving Objects in Image Sequences Based on Spatio-Temporal Information”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, Nº 8, pp. 1216 – 1226, Dezembro 1999.
- [Kim99a] W. Kim e Y. Kim, “A Region-Based Shape Descriptor using Zernike Moments”, *Signal Processing: Image Communication*, vol. 16, Nº 1-2, pp. 95 – 102, Setembro 2000.
- [Kim99b] W. Kim, “A Rotation Invariant Geometric Shape Descriptor using Zernike Moments”, Doc. *ISO/IEC JTC1/SC29/WG11/P687*, Lancaster *MPEG Meeting – Reino Unido*, Fevereiro 1999.
- [Kim99c] W. Y. Kim e Y. S. Kim, “Shape Descriptor Based on Multi-Layer Eigen Vector”, Doc. *ISO/IEC JTC1/SC29/WG11/P517*, Lancaster *MPEG Meeting – Reino Unido*, Fevereiro 1999.

- [Kim99d] H. K. Kim e J. D. Kim, “Region-Based Shape Descriptor Invariant to Rotation, Scale and Translation”, Número Especial sobre MPEG-7, *Signal Processing: Image Communication*, Vol. 16, Nº 1-2, pp. 87-93, Setembro 2000.
- [Kim00a] M. Kim, “Report for Cross-verification Results on a Region-based Shape Descriptor”, Doc. *ISO/IEC JTC1/SC29/WG11/M5862*, Noordwijkerhout *MPEG Meeting* – Holanda, Março 2000.
- [Kim00b] M. Kim, “Cross-verification Results of Region-based Shape Descriptors”, Doc. *ISO/IEC JTC1/SC29/WG11/M6068*, Genebra *MPEG Meeting* – Suíça, Maio 2000.
- [Kruse96] S. M. Kruse, “Scene Segmentation from Dense Displacement Vector Fields Using Randomised Hough Transform”, *Signal Processing: Image Communication*, Vol. 9, Nº 1, pp. 29 - 41, Novembro 1996.
- [Kruse99] S. Kruse, A. Graffunder e S. Askar: “A New Tracking System for Semi- Automatic Video Object Segmentation”, *Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'99*, Berlim – Alemanha, Maio 1999.
- [Manjunath02] B. S. Manjunath, P. Salembier e T. Sikora, “Introduction MPEG-7 Multimedia Content Description Interface”, *John Wiley & Sons*, 2002.
- [Marques96] F. Marqués, M. Pardàs e P. Salembier, “Coding-Oriented Segmentation of Video Sequences”, in “Video Coding: the Second Generation Approach”, *Edited by L. Torres e M. Kunt, Kluwer*, pp. 79 – 123, 1996.
- [Marques97] F. Marqués e C. Molina, “Object Tracking for Content-Based Functionalities”, *Visual Communications and Image Processing, VCIP '97*, São José – EUA, Fevereiro 1997.
- [Mech98] R. Mech e M. Wollborn, “A Noise Robust Method for 2D Shape Estimation of Moving Objects in Video Sequences Considering a Moving Camera”, *Signal Processing*, Número Especial sobre “Video Sequence segmentation for Content-Based Processing and Manipulation”, Vol. 66, Nº 2, pp. 203 – 217, 1998.
- [Messelodi99] S. Messelodi e C. M. Modena, “Automatic Identification and Skew Estimation of Text Lines in Real Scene Images”, *Pattern Recognition*, Vol. 32, Nº 5, pp. 791 – 810, Maio1999.
- [Mokhtarian99] F. Mokhtarian e S. Abbasi, “Shape-Based Indexing using Curvature Scale Space with Affine Curvature”, *Proc. of the First European Workshop on Content-Based Multimedia Indexing, IRIT, Toulouse* – França, pp. 255-262, Outubro 1999.
- [Moghaddamzadeh97] A. Moghaddamzadeh e N. Bourbakis, “A Fuzzy Region Growing Approach for Segmentation on Color Images”, *Pattern Recognition*, Vol. 30, Nº 6, pp. 867 – 881, 1997.
- [Montoya00] M. G. Montoya, C. Gil e I. Garcia, “Implementation of a Region Growing Algorithm on Multicomputers: Analysis of the Work Load Balance”, *Dept. de Arquitectura de Computadores y Electronica, Universidad de Almeria, Technical Report* – 2000.

- [Moscheni96] F. Moscheni, F. Dufaux e M. Kunt, "Object Tracking Based on Temporal and Spatial Information", *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'96*, Atlanta – EUA, pp.1914 – 1917, Maio 1996.
- [Moscheni98] F. Moscheni, S. Bhattacharjee e M. Kunt, "Spatio-Temporal Segmentation Based on Region Merging", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, Nº 9, pp. 897 - 915, Setembro 1998.
- [MPEG7-Req02] Requirements Group, "MPEG-7 Requirements", Doc. *ISO/IEC JTC1/SC29/WG11/N4981*, Klagenfurt MPEG Meeting – Áustria, Julho 2002.
- [MPEG7-Visual01] ISO/IEC 15938-3/FCD, "Information Technology – Multimedia Content Description Interface – Part 3: Visual", Singapura *MPEG Meeting* – Singapura, Março 2001.
- [Muller99] K. Muller e J. Ohm, "Descriptor for Arbitrarily Shaped Objects", Doc. *ISO/IEC JTC1/SC29/WG11/P568*, Lancaster *MPEG Meeting* – Reino Unido, Fevereiro 1999.
- [Murray87] D. H. Murray e H. Buxton, "Scene Segmentation from Visual Motion Using Global Optimisation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 9, Nº 2, pp. 220 – 228, Março 1987.
- [Musmann89] H. Musmann, M. Hotter e J. Ostermann, "Object-Oriented Analysis-Synthesis Coding of Moving Images", *Signal Processing: Image Communication*, Vol. 1, Nº 2, pp. 117 – 138, Outubro 1989.
- [Niblack95] W. Niblack e J. Yin, "A Pseudo-Distance measure for 2D Shapes Based on Turning Angle", *Proc. of ICIP-95*, pp. 352 – 355, Outubro 1995.
- [Niblack86] W. Niblack. "An Introduction to Image Processing", *Prentice-Hall, Englewood Cliff, NJ*, pp. 115-116, 1986.
- [Nunes95] P. Nunes, "Detecção de Fronteiras em Imagens Texturadas", Instituto Superior Técnico, Lisboa, Tese de Mestrado, Agosto 1995.
- [Ohya94] J. Ohya, A. Shio e S. Akamatsu, "Reorganizing Characters in Scene Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, Nº 2, pp. 214 – 220, Fevereiro 1994.
- [Otsu79] N. Otsu, "A Threshold Selection Method from Gray-level Histograms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 9, Nº 1, pp. 62 – 66, Janeiro 1979.
- [Paeth86] A. Paeth, "A Fast Algorithm for General Raster Rotation" *Graphics Interface*, pp. 77 – 81, Maio 1986
- [Pavlidis77] T. Pavlidis, "Structural Pattern Recognition", *Springer-Verlag*, 1977.
- [Pavlidis90] T. Pavlidis e Y. Liow, "Integrating Region Growing and Edge Detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, Nº 3, pp. 225 - 233, Março 1990.
- [Pereira02] F. Pereira e T. Ebrahimi "The MPEG-4 Book", *IMSC Press*, 2002.

- [Pham02] D. L. Pham, “Fuzzy Clustering with Spatial Constrains”, *IEEE International Conference on Image Processing, Rochester, Nova Iorque – EUA*, Setembro 2002.
- [Pratt91] W. Pratt, “Digital Image Processing”, *2nd Edition, John Wiley & Sons*, 1991.
- [Raghu96] P. P. Raghu e B. Yegnanarayana, “Segmentation of Gabor-Filter Textures Using Deterministic Relaxation”, *IEEE Transactions on Image Processing*, Vol. 5, Nº 12, pp. 1625 – 1635, Dezembro 1996.
- [Rui98] Y. Rui, T. Huang e S. Chang, “Image Retrieval: Past, Present and Future”, *Journal of Visual Communication and Image Representation*, 1998.
- [Russ95] J. Russ, “The Image Processing Handbook”, *IEEE Press, Inc.*, 2^a edição, 1995.
- [Sahoo88] P. K. Sahoo, S. Soltani e A. K. C. Wong, “A Survey of Thresholding Techniques”, *Computer Vision Graphics and Image Processing*, Vol. 41, Nº 2, pp. 233 – 260, Fevereiro 1988.
- [Salembier94] P. Salembier e M. Pardás, “Hierarchical Morphological Segmentation for Image Sequence Coding”, *IEEE Transactions on Image Processing*, Vol. 3, Nº 5, pp. 639 – 651, Setembro 1994.
- [Salembier97] P. Salembier, L. Garrido e D. Garcia, “Image Sequence Analysis and Merging Algorithms”, *International Workshop on Coding Techniques for Very Low Bit-rate Video*, Linköping – Suécia, pp. 1 – 8, Julho 1997.
- [Salembier99] P. Salembier e F. Marqués, “Region-Based Representations of Image and Video: Segmentation Tools for Multimedia Services”, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, Nº 8, pp. 1147 – 1169, Dezembro 1999.
- [Salton83] G. Salton e McGill MJ, “Introduction to Modern Information Retrieval”, *McGraw-Hill*, Nova Iorque, 1983.
- [Sato99] T. Sato, T. Kanade, E. K. Hughes, M. A. Smith e S. Satoh, “Video OCR: Indexing Digital News Libraries by Recognition of Superimposed Captions”, *Multimedia Systems*, Vol. 7, Nº 5, pp. 385 – 395, Setembro 1999.
- [ScanSoft] “OmniPage Pro 12.0 Office”, <http://www.omnipage.com/omnipage/ocr/>.
- [Serra93] J. Serra, “Image Analysis and Mathematical Morphology – Volume 1”, *Academic Press*, 1993.
- [Stephen94] G. A. Stephen, “String Searching Algorithms”, *World Scientific Publishing*, Singapura, 1994.
- [Tabatabai99] A. Tabatabai, “Normalized Contour as a Shape Descriptor for Visual Objects”, *Doc. ISO/IEC JTC1/SC29/WG11/P579, Lancaster MPEG Meeting – Reino Unido*, Fevereiro 1999.
- [Teh88] C. Teh e R. T. Chin, “On Image Analysis by the Methods of Moments”, *IEEE Transactions on Pattern and Machine Intelligence*, Vol. 10, Nº 4, pp. 496 – 514, Julho 1988.

- [Thoma89] R. Thoma e M. Bierling, “Motion Compensating Interpolation Considering Covered and Uncovered Background”, *Signal Processing: Image Communication*, Vol. 1, pp. 191 – 212, 1989.
- [Trier95] O. D. Trier e A. K. Jain, “Goal-Direct Evaluation of Binarization Methods”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, Nº 12, pp. 1191 – 1201, Dezembro 1995.
- [Wang94] J. Wang e E. Adelson, “Representing Moving Images with Layers”, *IEEE Transactions on Image Processing*, Vol. 3, Nº 5, pp. 625 – 638, Setembro 1994.
- [Wolf02] C. Wolf e J. M. Jolion, “Extraction and Recognition of Artificial text in Multimedia Documents”, *Laboratoire Reconnaissance de Formes et Vision INSA de Lyon, Technical Report* – 2002.
- [Wu93] S. Wu e J. Kittler, “A Gradient-based Method for General Motion Estimation and Segmentation”, *Journal of Visual Communication and Image Representation*, Vol. 4, Nº 1, pp. 25 – 38, Março 1993.
- [Wu96] X. Wu, “YIQ Vector Quantization in a New Color Palette Architecture,” *IEEE Transactions on Image Processing*, Vol. 5, Nº 2, pp. 321–329, Fevereiro 1996.
- [Wu99] V. Wu, R. Manmatha e E. M. Riseman, “Text Finder an Automatic System to Detect and Recognize Text in Images”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, Nº 11, pp. 1224 – 1229, Novembro 1999.
- [Yanowitz89] S. D. Yanowitz e A. M. Bruckstein, “A New Method for Image Segmentation”, *Computer Vision Graphics and Image Processing*, Vol. 46, Nº1, pp. 82 – 95, Abril 1989.
- [Zhang02] D. Zhang, R. K. Rajendran e S. Chang, “General and Domain-Specific Techniques for Detecting and Recognizing Superimposed Text in Video”, *IEEE International Conference on Image Processing, Rochester, Nova Iorque – EUA*, Setembro 2002.
- [Zhong95] Y. Zhong, K. Karu e A. K. Jain, “Location Text in Complex Color Images”, *Pattern Recognition*, Vol. 28, Nº 10, pp.1523 – 1535, Outubro1995.
- [Zhong98] D. Zhong e S. F. Chang, “AMOS: An Active System for MPEG-4 Video ObjectSegmentation”, *IEEE International Conference on Image Processing, ICIP’98, Chicago – EUA*, Outubro 1998.
- [Zhong00] Y. Zhong, H. Zhang e A. K. Jain “Automatic Caption Localizing in Compressed Video”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, Nº 4, pp. 385 – 392, Abril 2000.
- [Zhou97] J. Zhou e D. Lopresti “OCR for World Wide Web Images”, *Proceedings of SPIE, Document Recognition IV*, pp. 58 – 66, 1997.
- [Zibreira00] C. Zibreira, “Descrição e Procura de Vídeo Baseada na Forma”, Instituto Superior Técnico, Lisboa, Tese de Mestrado, Dezembro 2000.
- [Zucker76] S. Zucker, “Region Growing: Childhood and Adolescence”, *Computer Graphics and Image Processing*, Vol. 5, Nº3, pp 382 - 399, Setembro1976.