Universidade de Coimbra

Faculdade de Ciências e Tecnologia

Departamento de Engenharia Electrotécnica e de Computadores

Mestrado de Engenharia Electrotécnica e de Computadores

Detecção Automática de Texto em Sequências de Vídeo

Inês Daniela Cunha Nereu

Júri:

Prof. Fernando Santos Perdigão
Prof. Paulo José Monteiro Peixoto
Prof. Nuno Miguel Mendonça da Silva Gonçalves

Outubro de 2012

Agradecimentos

Agradeço ao meu melhor amigo, que tive a infelicidade de assistir ao seu falecimento durante o período de realização da dissertação, a quem devo parte do que sou, de quem sou e do que sou capaz de fazer. E porque ele foi a melhor das fontes de inspiração que poderia ter, por ser tão forte, tão leal, tão justo e tão inteligente.

Agradeço à minha mãe e à Eva que juntas contribuíram para o meu crescimento e progresso acreditando sempre em mim tanto nos momentos altos como nos baixos.

Agradeço também ao meu namorado porque toda a amizade, a paciência, o apoio e pela partilha de todas as celebrações a cada detecção de texto bem sucedida.

Por último, agradeço ao meu orientador, Dr. Paulo Peixoto, por todo o apoio, flexibilidade e compreensão que manifestou durante o período de desenvolvimento desta dissertação.

Resumo

A detecção de texto é importante uma vez que permite obter informação relevante em imagens digitais, video, bases de dados e páginas web. Contudo, a sua detecção é bastante desafiante uma vez que o texto está frequentemente integrado num fundo complexo. São propostos três métodos para detectar tanto texto gráfico como texto de cena em frames de video. O primeiro baseia-se na Transformada Wavelet de Haar com a decomposição nível um nas sub-bandas LL, HL e HH onde são computadas um conjunto de características que vão alimentar o algoritmo k-médias para classificar as zonas de texto e as sem texto. A média das sub-bandas da Wavelet de Haar e a imagem binária resultante do k-médias permitem a classificação dos pixeis de texto. Os blocos de texto são segmentados com base na análise das projecções horizontais e verticais. Finalmente é introduzido um método para eliminação dos falsos positivos baseado nos contornos intrínsecos e extrínsecos. O segundo método é baseado na Transformada de Fourier no espaço de cores RGB. Neste método as características são calculadas sobre a FT nas sub-bandas R, G e B as quais são passadas enquanto argumento para o algoritmo k-médias e as restantes fases são iguais às do método anterior. O terceiro método detecta contornos para obter o mapa dos contornos nas direcções horizontal, vertical, diagonal direita para cima e diagonal esquerda para cima. Em seguida, as características são calculadas sobre os quatro mapas de contornos para serem o argumento do algoritmo de classificação k-médias. As restantes fases são iguais às dos dois métodos resumidos anteriormente. Por último foi ainda implementado um método de classificação de frames através de três regras formuladas com base no comportamento dos contornos para identificar frames de texto. Todos os métodos foram testados para uma variedade de imagens incluindo imagens com baixo contraste, diferentes fontes e distintos tamanhos de caracteres. Os resultados experimentais demonstram um melhor desempenho para o primeiro método.

Palavras-chave: Transformada Wavelet de Haar, Transformada de Fourier, detecção de contornos, algoritmo de classificação K-médias, eliminação de falsos positivos, classificação de frames.

Abstract

Text detection is important in the retrieval of texts from digital pictures, video, databases and WebPages. However, it can be very challenging since the text is often embedded in a complex background. I propose three methods for both graphics and scene text detection in video frames. The first is based on Haar Wavelet Transform, this method uses Wavelet single level decomposition LL, HL and HH sub bands for computing features and the computed features are fed to k-means clustering to classify the text pixel from the background of the image. The average of the Wavelet sub bands and the output of k-means clustering helps in classifying true text pixel in the image. The text blocks are detected based on analysis of projection profiles. Finally I introduce a method based on intrinsic and extrinsic edges to eliminate the false positives. The second method is based on Fourier Transform (FT) in RGB space and the features are computed over Fourier Transform on R,G and B sub bands to be fed to k-means, the rest of the steps are like the ones of the first method. The third method applies edge detection to get four edge maps in horizontal, vertical, up-right and up-left direction. Secondly the features are extracted from four edge maps to be fed to k-means. The rest of the steps are also like the ones of the first and second methods. Moreover a text frame classification is proposed based on three visual rules of the edges to indentify a true text frame. The robustness of all the methods is tested by conducting experiments on a variety of images of low contrast, different fonts and size of text in the image. The experimental results show that Haar Wavelet Transform outperforms the other methods.

Keywords: Haar Wavelet Transform, Fourier Transform, edge analysis, k-means clustering, false positive elimination, text frames classification.

Nomenclatura

- HWT: *Haar Wavelet Transform*, Transformada Wavelet de Haar
- L: *Low*, baixo
- H: *High*, alto
- FV: *Feature vecto*r, vector de características .
- NFV: *Normalized Feature Vector*, vector de características normalizado
- T: *Threshold*, limiar
- FC: *Frequency Coefficients*, coeficientes de frequência
- MFC: *Mean Frequency Coefficients*, media dos coeficientes de frequência
- HFC: *High Frequency Coefficients*, coeficientes de frequência elevados
- SFC: *Sum Frequency Coefficients*, soma dos coeficientes de frequência
- NHFC: *Number of High Frequency Coefficients*, número de coeficientes de frequência elevados
- BB: *Bounding Boxes*, caixa menor que contem um elemento conectado
- FT: *Fourier Transform*, Transformada de Fourier
- IFT: *Inverse Fourier Transform*, Transformada Inversa de Fourier
- AIFT: *Absolute Inverse Fourier Transform*, Transformada Inversa de Fourier Absoluta
- NAIFT: *Normalized Absolute Inverse Fourier Transform*, Transformada Inversa de Fourier Absoluta Normalizada
- RGB: *Red*, *Green*, *Blue*, vermelho, verde, azul
- R1: Regra 1
- R2: Regra 2
- R3: Regra 3
- AF: *Arithmetic Filter*, filtro médio aritmético
- MF: *Median Filter*, filtro mediano
- Std: *Standard Desviation*, desvio padrão
- BD: *Block Detected*, blocos detectados
- TD: *Truly Detected*, realmente detectados
- MD: *Miss Detection*, detecção incompleta
- FD: *False Detection*, falsa detecção
- DR: *Detection Rate*, taxa de detecção realmente detectados
- MDR: *Miss Detection Rate*, taxa de detecção incompleta
- FDR: *False Detection Rate*, taxa de falsa detecção

ÍNDICE

Índice de Tabelas ..... **ii**

Índice de Figuras ..... **iii**

1. Introdução **1**

2. Detecção de texto pela Transformada Wavelet de Haar **3**

2.1. Transformada Wavelet de Haar 4

2.2. Vector de características 6

2.3. Algoritmo de agrupamento K-Médias 8

2.4. Operações Morfológicas de abertura e dilatação 9

2.5. Mapeamento sobre a imagem média 11

2.6. Segmentação 13

2.7. Eliminação de falsos positivos 16

3. Detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB **18**

4. Detecção de texto pela baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB **23**

5. Pré-Processamento **28**

5.1. Método Máx-Min 29

5.2. R1 34

5.3. R2 37

5.4. R3 39

5.4. Classificação final 41

6. Resultados e Conclusões **43**

6.1. Resultados do Pré-Processamento 43

6.2. Resultados dos algoritmos propostos 45

7. Limitações dos algoritmos propostos **47**

8. Referências Bibliográficas **48**

ANEXOS ..... **A**

Índice de Tabelas

Tabela		Pág.
1	Resultados da classificação pelo Máx-Min	44
2	Resultados da classificação por R1,R2, e R3	44
3	Resultados da classificação pelo Máx-Min e R1,R2, e R3	44
4	Resultados para as 100 frames de texto para os algoritmos	45
5	Desempenho para as 100 frames de texto para todos os algoritmos	45

Índice de Figuras

Figura		Pág.
1	Fluxograma do algoritmo para detecção de texto pela Transformada Wavelet de Haar	3
2	Histogramas das sub-bandas HH, HL, LH e LL	5
3	Passos intermediários no processo de detecção de texto pela Wavelet de Haar	6
4	Passos de aplicação do algoritmo K-médias	8
5	Operação morfológica de erosão	10
6	Operação morfológica de dilatação	10
7	**(a)** Coeficientes da Wavelet de Haar na imagem média na Figura 3(f), **(b)** Perfil na linha 217 de (a), linha de texto e **(c)** Perfil na linha 54 de (a), linha sem texto	12
8	**(a)** Projecção horizontal das *bounding boxes* de (b), **(b)** Imagem mapeada com representação dos rectángulos para cada componente conectado e **(c)**, Projecção vettical das bouding boxes de (b)	14
9	Esquema representativo da projecção horizontal	15
10	**(a)** G(x,y) com caixas resultantes do processo de segmentação, **(b)** Projecção de uma caixa sem texto, **(c)** Eliminação dos contornos extrínsecos e intrínsecos de (b), **(d)** Projecção de parte de uma caixa de texto e **(e)** Eliminação dos contornos intrínsecos e extrínsecos de (d)	18
11	Fluxograma do algoritmo para a detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB	19
12	Passos intermediários no processo de detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB	21
13	Representação 3D dos coeficientes NAIFT **(a)** sub-banda R, **(b)** da sub-banda G, **(c)** da sub-banda B e **(d)** da banda banda média Avg	22
14	**(a)** Coeficientes de NAIFT na imagem média na Figura 12(h), **(b)** Perfil na linha 217 de (a), **(c)** Perfil na linha 54 de (a), linha sem texto	23
15	Fluxograma do algoritmo para detecção de texto baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB	24
16	Passos intermediários no processo de detecção de texto baseado no	26

	comportamento dos contornos de Sobel sobre o espaço de cores RGB	
17	Representação 3D para θ=135º **(a)** na sub-banda R, **(b)** na sub-banda G, **(c)** na sub-banda B	27
18	**(a)** Valores do gradiente na imagem média da Figura 16(f), **(b)** Perfil na linha 217 de (a), linha de texto, **(c)** Perfil na linha 54 de (a), linha sem texto	28
19	Fluxograma do Pré-Processamento	29
20	Divisão em blocos	30
21	Blocos 5 e 15	31
22	Classificação Máx-Min	32
23	**(a)** Frame sem texto, **(b)** Blocos e **(c)** Classificação Máx-Min	33
24	**(a)** Frame com texto, **(b)** Blocos e **(c)** Classificação Máx-Min	33
25	Fluxograma representativo de R1	35
26	Passos de classificação com base em R1	36
27	Classificação com base em R1 (a)	36
28	$NSobel_{AF}$ e $NCanny_{Diff}$ para os blocos 13, 14 e 16	37
29	**(a)-(d)** Bloco 5 sem texto e **(c)-(h)** Bloco 15 de texto	38
30	**C**lassificação com base em R2 **(a)** para os blocos classficados pelo Máx-Min **(b)** para os restantes blocos	38
31	S_S e S_C para os blocos 13, 14 e 16	38
32	Proximidade entre os centróides dos contornos para os blocos 5 e 15	40
33	**C**lassificação de acordo com R3 **(a)** para os blocos classficados pelo Máx-Min **(b)** para os restantes blocos	40
34	Proximidade entre os centróides dos contornos para os blocos 13, 14 e 16	41
35	Classificação final da frame	42
36	**(a)** Frame sem texto e **(b)** Classificação final da frame (a)	42
37	Exemplo de frame com texto multi-orientado	47
A-1	Resultados do Algoritmo de Detecção pela Transformada Wavelet de Haar	B
A-2	Resultados do Algoritmo de Detecção pela Transformada de Fourier sobre o espaço de cores RGB	D
A-3	Resultados do Algoritmo de Detecção de texto baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB	F

1. Introdução

Com o rápido avanço na tecnologia digital e os preços cada vez mais acessíveis ao público em geral, o vídeo tornou-se vulgar no quotidiano da maioria das pessoas. Praticamente toda a gente hoje em dia tem um ou mais telemóveis com câmaras integradas que permitem a gravação de vídeos, além das câmaras fotográficas e de vídeo. O texto presente nos vídeos apresenta elevada carga semântica. Denote-se por exemplo que em muitas páginas web o título das mesmas está contido numa imagem. A identificação de texto nas capas dos livros e de jornais pode ser importante para permitir utilizar essa informação posteriormente em formato digital. O texto presente nos noticiários usualmente transmite informação importante relativamente a um dado evento permitindo muitas vezes identificar a data da sua ocorrência e quem esteve envolvido. Em vídeos desportivos, o texto contém informação relativa aos atletas e classificações.

Actualmente os sistemas de OCR (*Optical Character Recognition)* reconhecem caracteres com elevada precisão nos casos em que o texto se encontra separado do fundo. Contudo, quando testados para imagens com fundo complexo apresentam normalmente um baixo desempenho. Acrescido a este facto as frames dos vídeos apresentam problemas adicionais tais como baixo contraste, baixa resolução, desfasamento de cor e o texto surge muitas vezes com os contornos mal definidos. Os algoritmos de detecção de texto podem então extrair as zonas de texto as quais podem posteriormente alimentar módulos OCR para o seu reconhecimento. Inúmeros algoritmos para a detecção de texto têm sido apresentados ao longo dos últimos anos, no entanto, devido à natureza peculiar das imagens de vídeo, pelos motivos apresentados anteriormente, a detecção e extracção de texto em frames de vídeo é ainda desafiante, [1-3].

Consideram-se essencialmente quatro grandes categorias na detecção e extracção de texto: a baseada em componentes conectados, a baseada na textura, a baseada em gradiente e contornos e a baseada na cor. Os algoritmos baseados em componentes conectados são bastante simples e baseiam-se nas propriedades geométricas dos componentes para a detecção de texto em vídeos. Adicionalmente assumem que os pixeis de texto que pertencem à mesma região conectada partilham algumas características em comum tal como a cor ou o mesmo nível na escala de cinzentos. Devido a estas suposições estes métodos não apresentam um bom desempenho em imagens de vídeo com fundo complexo nem em frames que contém simultaneamente texto gráfico e texto de cena [4-5].

Os métodos baseados na textura consideram que o texto apresenta um padrão com uma propriedade de textura diferente, desta forma necessitam um elevado contraste entre as zonas de texto e as zonas sem texto o que nem sempre é conseguido essencialmente para o texto de cena [6-8].

A terceira categoria baseada no gradiente e em contornos permite a detecção tanto de texto de cena como de texto gráfico com um baixo esforço computacional, como tal são bastante populares para a detecção de texto. No entanto, apresentam problemas essencialmente quanto ao elevado número de falsos positivos e quanto à selecção do limiar para a classificação dos pixeis de texto [9-13].

A categoria final considera a cor para a detecção de texto em frames de vídeo no espaço de cores L*a*b e com contornos de Sobel nos canais Y U V [14-16]. Estas aproximações falham quando o texto numa frame contém múltiplas cores numa linha de texto ou numa palavra. Assim, aproximações baseadas na cor não apresentam bom desempenho para frames com texto de cena e fundo complexo.

Em contraste com as aproximações anteriores foram também propostos métodos baseados em SVM (*Supervised Machine Learning*) e redes neuronais os quais apresentam elevado desempenho para texto gráfico mas requerem um grande número de características e um extenso treino com o classificador [17].

Com base nestas observações são propostos três métodos o primeiro baseado na Transformada Wavelet de Haar, o segundo baseado na Transformada de Fourier sobre o espaço de cores RGB e o terceiro baseado no comportamento dos contornos no espaço de cores RGB. Apesar do anteriormente exposto relativamente aos contornos optou-se por implementar na mesma este método, por permitir a detecção tanto de texto de cena como de texto gráfico, introduzindo um cálculo automático do limiar. Em acréscimo e para combater o problema dos falsos positivos foi implementado um método baseado nos contornos intrínsecos e extrínsecos. Foi também observado que os algoritmos de detecção retornam imensos falsos positivos quando testados para frames sem texto, como tal foi proposta uma abordagem designada por Pré-Processamento que permite uma pré classificação das frames determinando se contém ou não texto antes de passarem para os algoritmos de detecção.

2. Detecção de texto pela Transformada Wavelet de Haar

O algoritmo de reconhecimento de texto com base na Transformada Wavelet de Haar encontra-se representado no fluxograma da figura seguinte, Figura 1:

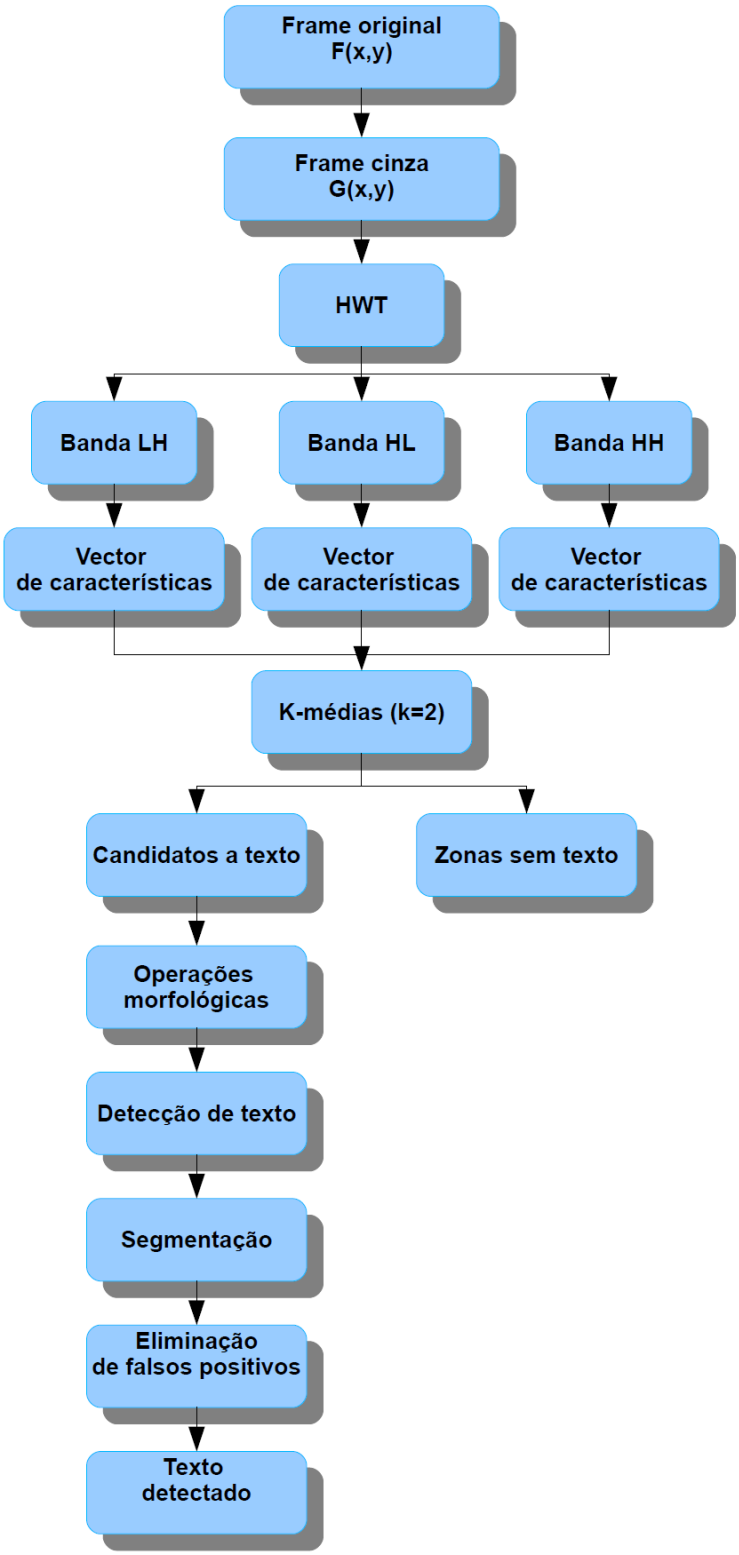
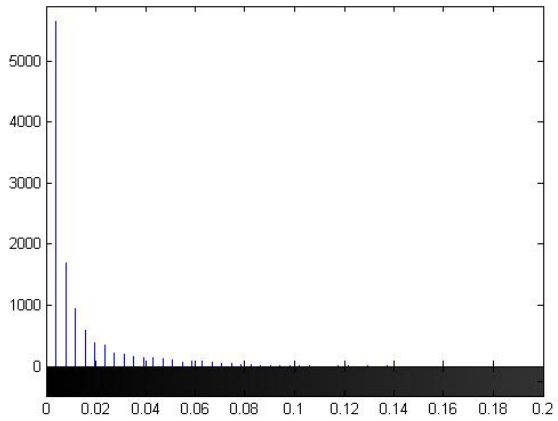


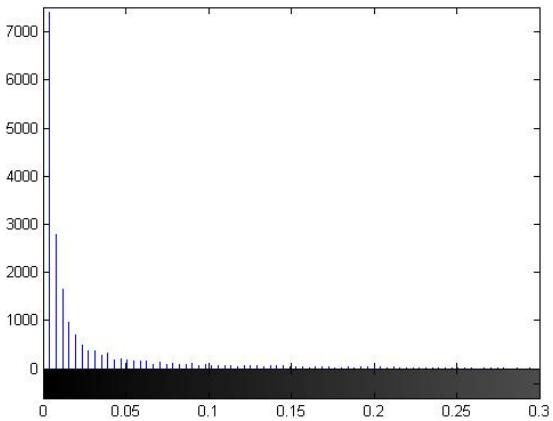
Figura 1. Fluxograma do algoritmo para detecção de texto pela Transformada Wavelet de Haar

2.1 Transformada Wavelet de Haar

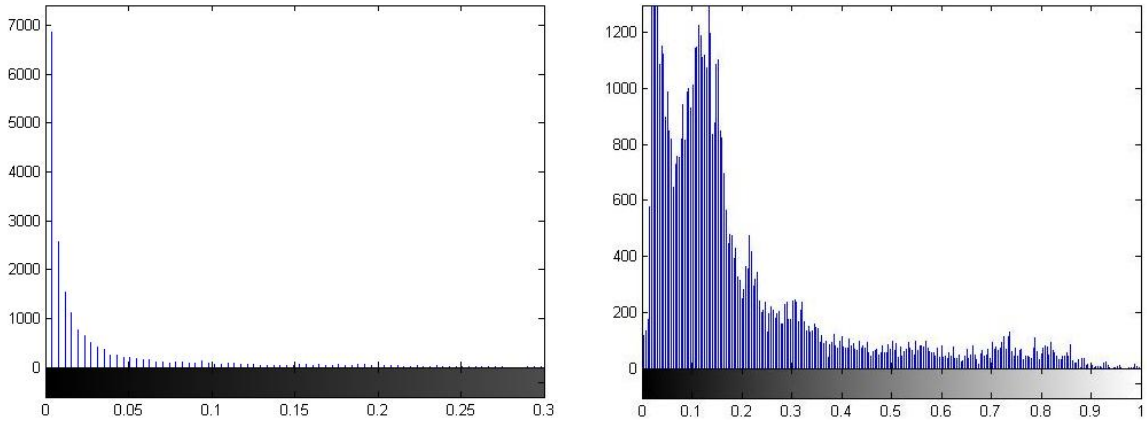
Considere-se a frame original de dimensões 256x256, F(x,y), e a correspondente imagem em tons de cinza, G(x,y), representadas nas Figuras 2(a) e 2(b), respectivamente. Para detectar texto nas frames de vídeo utiliza-se a decomposição 2D para a Transformada Wavelet de Haar (2D-HWT), nível um [18]. Esta decomposição é aplicada sobre cada linha de G(x,y) e posteriormente sobre cada coluna da imagem resultante da primeira operação, permitindo desta forma a detecção individualizada dos contornos horizontais e verticais. A imagem resultante é decomposta em quatro sub-bandas: LL, HL, LH e HH (L=*Low*, H=*High*). A sub-banda LL consiste numa aproximação de G(x,y), LH preserva os contornos (*edges*) horizontais, HL preserva os contornos verticais e HH preserva os detalhes diagonais, de acordo com os filtros usados para gerar cada sub-banda. Por exemplo, HL significa que é utilizado um filtro passa-alto ao longo das linhas e um filtro passa-baixo ao longo das colunas. Em seguida, para cada sub-banda, é aplicada a Inversa da Transformada de Haar (2D-IHWT) em cada coluna e depois em cada linha da imagem resultante da primeira operação. As sub-bandas obtidas encontram-se representadas na Figura 3(c)-(e). Denote-se que a sub-banda LL não é considerada pois são as propriedades estatísticas das restantes sub-bandas, a distribuição Laplaciana dos coeficientes da Wavelet, que contribuem para a detecção de texto, conforme se pode visualizar nos histogramas da Figura 2.



(a) Histograma da sub-banda HH



(b) Histograma da sub-banda HL

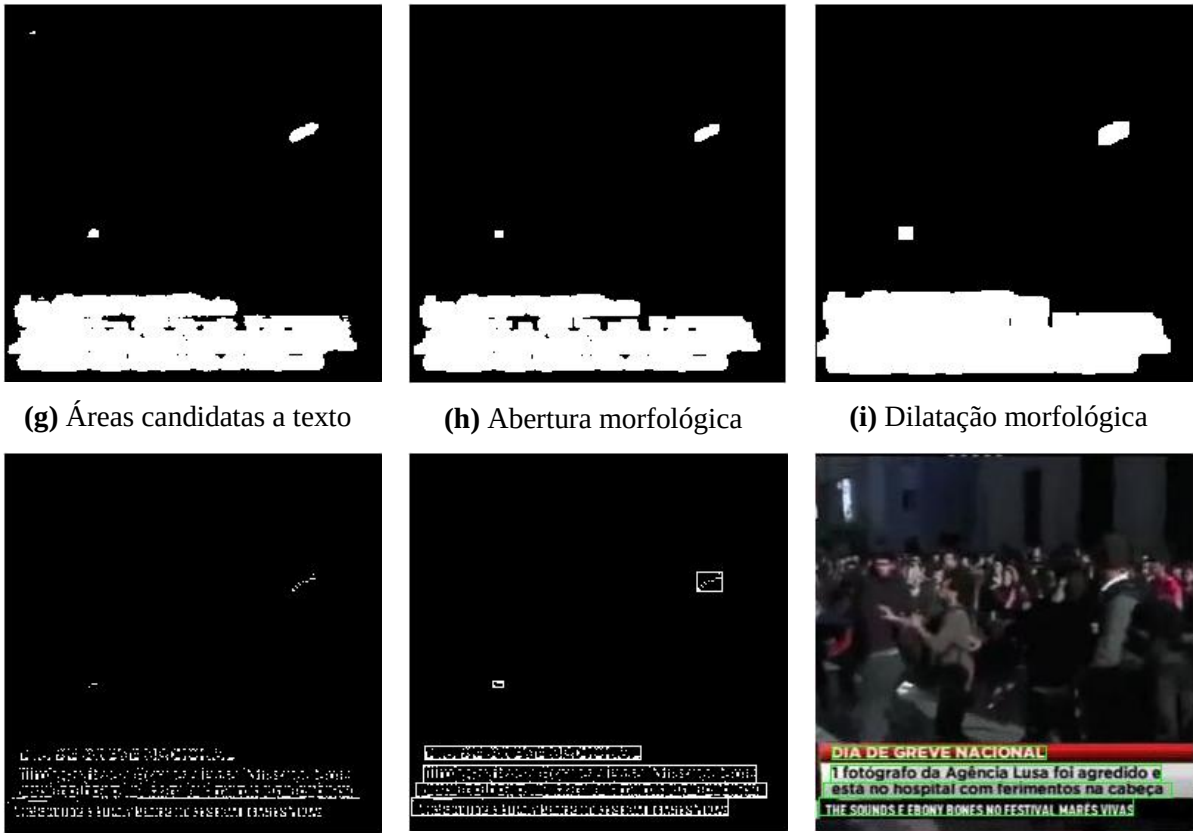


(c) Histograma da sub-banda LH **(d)** Histograma da sub-banda LL

Figura 2. Histogramas das sub-bandas HH,HL,LH e LL

(a) Imagem original **(b)** Imagem em escala cinza **(c)** Horizontal(LH)

(d) Vertical(HL) **(e)** Diagonal(HH) **(f)** Imagem média



(g) Áreas candidatas a texto **(h)** Abertura morfológica **(i)** Dilatação morfológica

(j) Imagem mapeada **(k)** Segmentação **(l)** Texto extraído

Figura 3. Passos intermediários no processo de detecção de texto pela Wavelet de Haar

2.2 Vector de características

Para cada uma das sub-bandas, Figura 3(c)-(d), são calculados um conjunto de características (*features*) estatísticas para capturar informação relativa à textura presente nas imagens. Com base no facto de que o texto apresenta textura distinta, é possível distinguir as zonas de texto das sem texto. As características propostas incluem a energia (E), entropia (Et), inércia (I), homogeneidade (Hm), média (M), momento centrado de segunda ordem (μ_2) e momento centrado de terceira ordem (μ_3). Uma janela de dimensões N x N (N=8) percorre cada uma das sub-bandas, centrada em cada pixel. A escolha da dimensão da janela é determinante para a detecção, como tal foram efectuados testes para janelas de dimensões 2x2, 4x4, 8x8 e 12x12, obtendo-se melhores resultados para N=8. Para cada uma das sub-bandas e para cada uma das posições da janela considerada, as características são calculadas de acordo com as equações seguintes:

$$E = \sum_{x=1}^{N}\sum_{y=1}^{N} W^2(x,y) \qquad \textbf{(1)}$$

$$Et = \sum_{x=1}^{N}\sum_{y=1}^{N} W(x,y).\log W(x,y) \qquad \textbf{(2)}$$

$$I = \sum_{x=1}^{N}\sum_{y=1}^{N} (x-y)^2.W(x,y) \qquad \textbf{(3)}$$

$$Hm = \sum_{x=1}^{N}\sum_{y=1}^{N} \frac{1}{1+(x-y)^2} W(x,y) \qquad \textbf{(4)}$$

$$M = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N} W(x,y) \qquad \textbf{(5)}$$

$$\mu_2 = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N} (W(x,y)-M)^2 \qquad \textbf{(6)}$$

$$\mu_3 = \frac{1}{N^2}\sum_{x=1}^{N}\sum_{y=1}^{N} (W(x,y)-M)^3 \qquad \textbf{(7)}$$

Onde W(x,y) corresponde a cada sub-banda, Figura 3(c)-(e), para cada pixel na posição (x,y), na janela de dimensões NxN.

As características computadas perfazem um total de vinte e uma, o que corresponde a sete características multiplicadas por cada uma das três sub-bandas LL, HL e HH. Um vector de características, FV, com dimensões 65536x21, é formado com cada linha correspondente a cada pixel, para frames de 256x256 pixeis. O vector FV é depois normalizado para o intervalo [0,1], designando-se por NFV. A normalização do vector de características antes de o passar enquanto argumento para o algoritmo de classificação K-médias é importante uma vez que contribui para a desempenho do algoritmo. A normalização é especialmente necessária já que métricas de distância, como a distância euclidiana utilizada no algoritmo K-médias, são sensíveis a diferenças de magnitude nos atributos. Há vários métodos para a normalização de dados, no entanto, e de acordo com resultados inerentes da comparação entre os diferentes métodos [19], foi utilizada a normalização Min-Máx. Para cada atributo de FV, correspondente a cada coluna do vector, determinar:

$$NFV = \frac{FV(i) - Min}{Max - Min} \qquad \textbf{(8)}$$

Com FV(i) os valores de FV na coluna i (i=1..21), Min o valor mínimo e Máx o valor máximo da coluna i de FV.

2.3 Algoritmo de agrupamento K-Médias

O algoritmo K-médias (*K-means algorithm*) permite uma classificação de informações de acordo com os próprios dados. Esta classificação é baseada em análises e comparações entre os valores numéricos dos dados. Desta maneira, o algoritmo vai fornecer uma classificação automática sem a necessidade de nenhuma supervisão humana, ou seja, sem nenhuma pré-classificação existente. Por causa desta característica, o K-médias é considerado como um algoritmo não supervisionado. O objectivo deste algoritmo é encontrar a melhor divisão de um conjunto P de dados em k grupos (*clusters*) de maneira a que a distância total entre os dados de um grupo e o seu respectivo centro seja minimizada. Este método consiste em utilizar os valores dos primeiros n casos como estimativas temporárias das médias dos k grupos, onde k é o número de grupos previamente especificado. Assim, o centro do *cluster* inicial é formado para cada caso em torno dos dados mais próximos e, então, comparados com os pontos mais distantes permitindo a formação dos restantes grupos. A partir daí, dentro de um processo de actualização contínua e de um processo interactivo [23] encontram-se os centros dos grupos (centróides). Em outras palavras, o algoritmo atribui aleatoriamente os P pontos a k grupos e calcula as médias dos vectores de cada grupo, os centróides. Em seguida, cada ponto é deslocado para o grupo correspondente ao vector médio do qual ele está mais próximo, de acordo com uma determinada métrica de distância, e novos vectores médios são calculados. O processo de re-alocação de pontos a novos grupos cujos vectores médios são os mais próximos deles continua até que se chegue a uma situação em que todos os pontos já estejam nos grupos dos seus vectores médios mais próximos. O esquema da Figura 4 resume o comportamento do algoritmo K-médias.

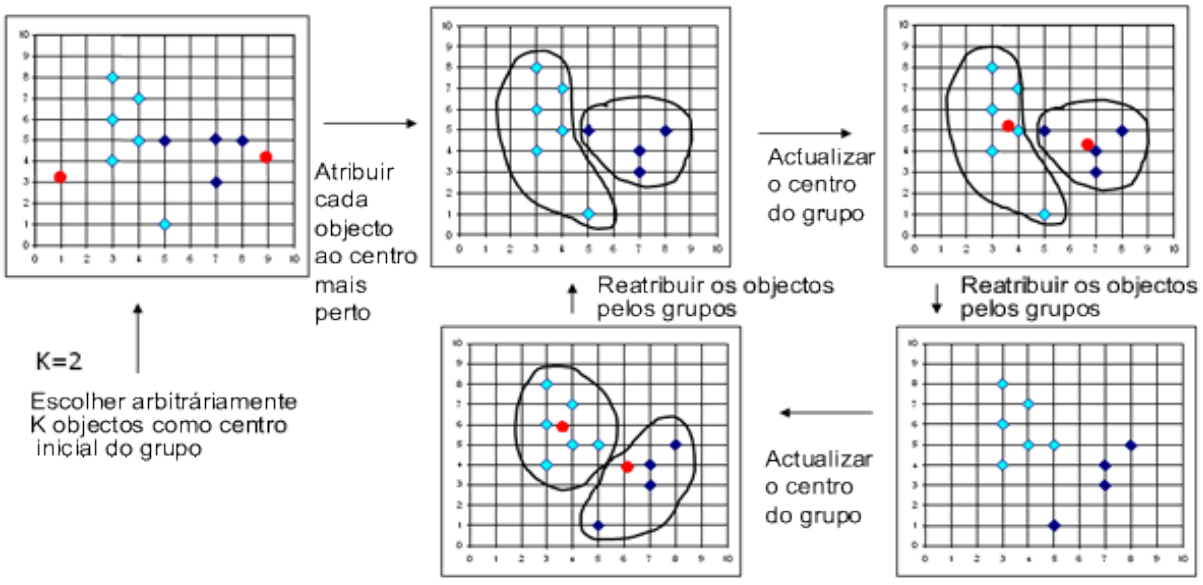


Figura 4. Passos de aplicação do algoritmo K-médias

O algoritmo de classificação K-médias é então aplicado para classificar o vector de características normalizado em dois grupos: os pertencentes a áreas sem texto e os candidatos a área de texto, logo k=2. A métrica de distância utilizada foi a distância euclidiana. Tal como em outros algoritmos de minimização numérica, os resultados do algoritmo K-médias dependem frequentemente dos pontos iniciais escolhidos. É possível que o K-médias encontre um mínimo local em que ao mover um ponto para outro grupo aumente a distância total entre os pontos desse grupo e o seu respectivo centróide, existindo contudo uma solução melhor. Para minimizar a problemática dos mínimos locais o algoritmo foi repetido duas vezes, pois desta forma novos pontos iniciais são atribuídos.

Uma vez que o algoritmo K-médias é um algoritmo não supervisionado é necessária uma classificação para determinar qual dos grupos corresponde as zonas candidatas a texto. É utilizada a média de cada *cluster* como base de classificação. Como os pixeis de texto apresentam valores de frequência mais elevados em comparação com os pixeis de áreas sem texto, a sua média será mais elevada. Os resultados obtidos podem ser visualizados na imagem binária da Figura 3(g) em que as zonas candidatas a texto aparecem a branco e as zonas sem texto a preto.

2.4 Operações morfológicas de abertura e dilatação

Em seguida, operações morfológicas [20] como a abertura e a dilatação são aplicadas sobre a imagem da Figura 3(g) para eliminar pequenas áreas detectadas e desta forma facilitar o processo de segmentação. A operação morfológica de abertura consiste numa erosão seguida de uma dilatação. A erosão e a dilatação são as operações básicas de processamento morfológico de imagens obtidas por varrimento da imagem por uma máscara ou elemento estruturante, obtendo-se à saída uma imagem de igual tamanho.

O procedimento de erosão consiste em centrar em cada pixel a branco da imagem binária, valor numérico 1, o centro do elemento estruturante. Para cada pixel a branco se algum dos oito vizinhos tiver pelo menos um valor numérico 0, equivalente ao preto, o pixel passa a ter o valor 0. Se pelo contrário todos os seus vizinhos têm o valor numérico 1 permanecerá com o mesmo valor. A operação de erosão, equivalente a função lógica *AND*, pode ser visualizada na Figura 5 em que A representa um qualquer objecto genérico e B o elemento estruturante considerado. Considera-se objecto como as áreas de uma imagem binária que surgem a branco, valor 1, inseridas num fundo a negro, valor 0.

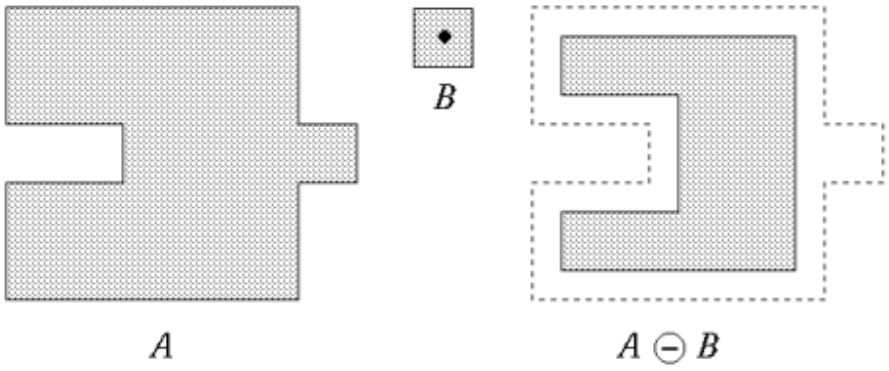


Figura 5. Operação morfológica de erosão

Este procedimento elimina objectos pequenos e objectos maiores têm a sua área reduzida.

A dilatação, por sua vez, realiza a operação inversa, em cada pixel da imagem binária é também mapeado o centro do elemento estruturante. Se o pixel apresenta o valor numérico 1 permanecerá com o mesmo valor, se apresenta valor numérico 0 e pelo menos um dos seus oito vizinhos tem valor numérico 1 então o valor é actualizado para 1. A operação de dilatação, equivalente à função lógica *OR*, encontra-se representada na Figura 6 em que A representa um qualquer objecto genérico e B o elemento estruturante considerado.

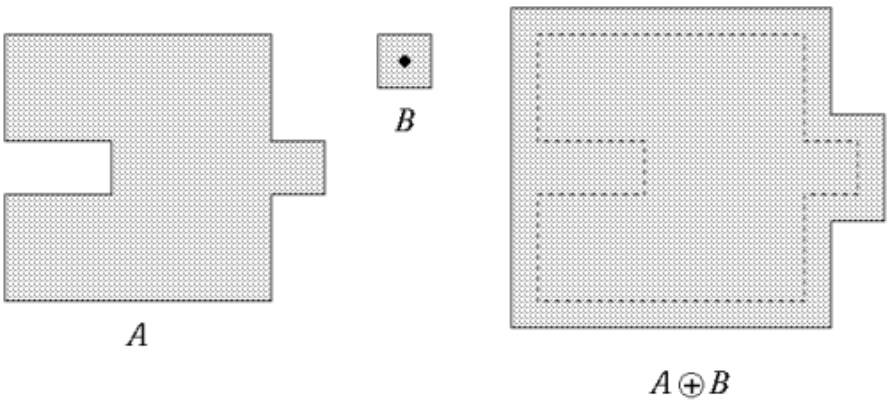


Figura 6. Operação morfológica de dilatação

Na dilatação, os objectos por sua vez têm a sua área aumentada.

Assim a operação de abertura vai permitir eliminar pequenos objectos a branco pela erosão e recuperar a forma dos restantes objectos, pela dilatação, sem restaurar os primeiros. O resultado da operação de abertura pode ser visualizado na Figura 3(h) onde um pequeno objecto no canto superior da Figura 3(g) surge agora eliminado. O resultado da operação de dilatação pode ser visualizado na Figura 3(i) onde os objectos são aumentados e buracos a negro surgem depois com o seu tamanho diminuído. Esta operação é vantajosa pois ao "engordar" as áreas a branco permitirá em casos em que apenas uma parte de um dado caractere ou de uma expressão

seja detectada minimizar a possibilidade de a detecção final aparecer incompleta. Para ambas as operações foi utilizado um elemento estruturante quadrado de dimensões 5x5.

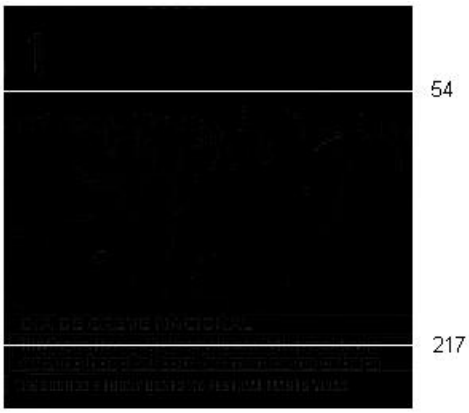
2.5 Mapeamento sobre a imagem média

Em seguida, a imagem resultante das operações morfológicas, na Figura 3(i), vai ser projectada sobre a imagem média, Figura 3(f), para se obterem os candidatos a texto correspondentes na banda média. A imagem média é determinada de acordo com a seguinte equação:

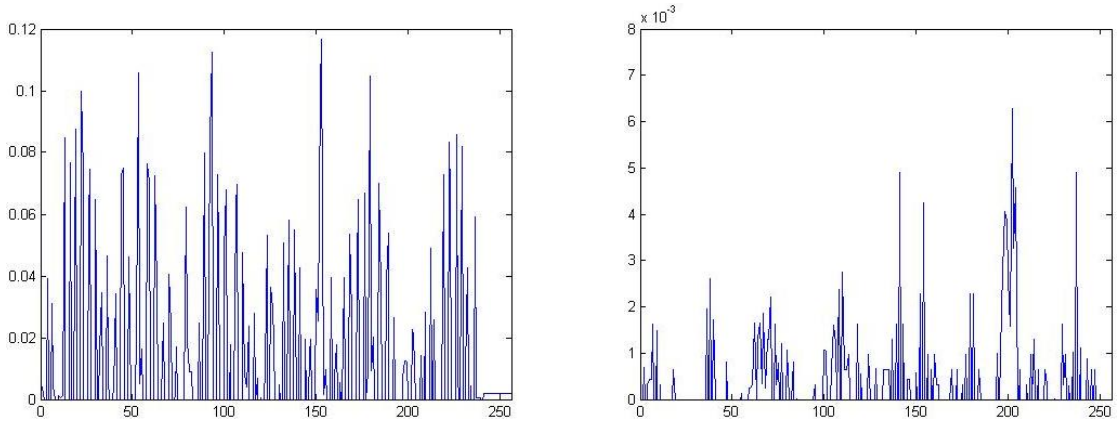
$$Avg(x,y) = \frac{1}{3}\sum_{k=1}^{3} W_k(x,y) \qquad \textbf{(9)}$$

Em que Avg(x,y) corresponde à imagem média e W_k(x,y) corresponde a cada uma das sub-bandas LH, HL e HH da Figura 3(c)-(e).

A motivação para o uso da Transformada Wavelet de Haar para a detecção de texto consiste na maior expressividade dos seus coeficientes em zonas de texto do que em zonas sem texto. Podem visualizar-se os coeficientes da Wavelet na imagem média, Figura 7(a), para uma linha de texto (217 na Figura 7(b)) e para uma linha sem texto (54 na Figura 7(c)).



(a)



(b) **(c)**

Figura 7. (a) Coeficientes da Wavelet de Haar na imagem média na Figura 3(f), **(b)** Perfil na linha 217 de (a), linha de texto e **(c)** Perfil na linha 54 de (a), linha sem texto

Posteriormente é obtida a imagem mapeada, MapIm(x,y) na Figura 3(j), esta imagem contém os pixeis de texto inerentes da detecção e designa-se por mapeada pois resulta da projecção da imagem resultante das operações morfológicas, Morf(x,y), na Figura 3(i), sobre a imagem média das 3 sub-bandas, Avg(x,y) na Figura 3(f). A imagem MapIm(x,y) é obtida de acordo com a equação seguinte:

$$Map\,\mathrm{Im}(x,y) = \begin{cases} 1, & if\ ((Avg(x,y) > T)\ \&\ (Morf(x,y) = 1)) \\ 0, & else \end{cases} \qquad \textbf{(10)}$$

A variável T, consiste num limiar (*threshold*) o qual é calculado automaticamente sobre a imagem média, Avg(x,y), obtida de acordo com a equação (9). A base para a sua determinação [13] consiste em extrair apenas os pixeis que apresentam coeficientes com maior frequência que os pixeis que correspondem a áreas sem texto, já que, conforme se verificou na Figura 7, em zonas de texto os coeficientes apresentam maior valor. Considere-se então FC como o vector que contém os coeficientes de frequência na imagem média, maiores ou iguais que 0,05. A escolha deste valores resulta da projecção dos valores dos coeficientes para inúmeras linhas de texto e sem texto, tal como exemplificado na Figura 7(a)-(c). A média de FC (*Frequency coefficients*), MFC (*Mean frequency coefficient*), é calculada de acordo com a equação seguinte

$$MFC = \frac{1}{m}\sum_{i=1}^{m} FC_i \qquad \textbf{(11)}$$

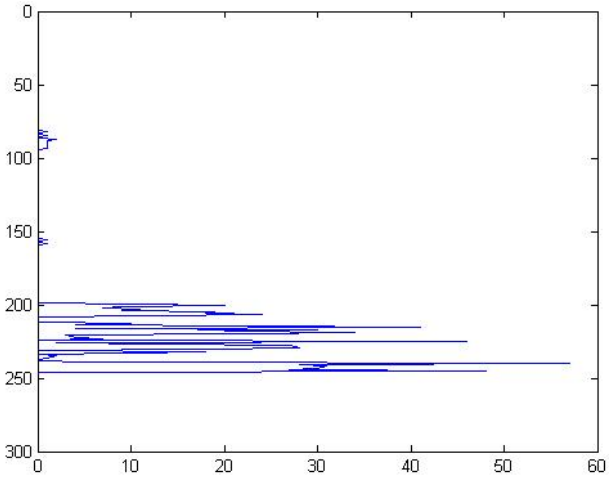
Com m o número de elementos no vector FC. Considere-se agora HFC (*High frequency coefficients*) como o vector que contém os valores de Avg(x,y)>=MFC. Para o cálculo de T, contabilizar também a variável SFC (*Sum frequency coefficients*) a qual corresponde a soma dos valores no vector FC e NHFC *(Number of high frequency coefficients*) que corresponde ao número de elementos em HFC. O limiar T é então definido como:

$$T = \frac{SFC}{(m + NHFC)} \qquad \textbf{(12)}$$

2.6 Segmentação

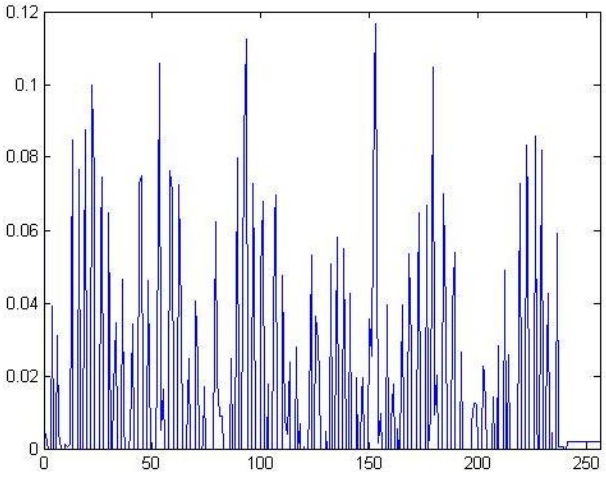
A segmentação consiste em definir as caixas que envolvem as zonas candidatas a texto. Esta tarefa foi das mais exaustivas e das que exigiram mais tempo, pois nesta fase e tal como nas restantes procurou determinar-se sempre um método que se aplique à generalidade das frames de vídeo.

Para a imagem mapeada, Figura 3(j) são determinados os rectângulos mais pequenos que contém cada componente conectado (*bounding boxes*) da imagem, Figura 8(b), com as *bounding boxes* a vermelho. Um componente conectado identifica um conjunto de pixeis no qual cada um está conectado a um ou mais vizinhos que se encontrem em uma das 8 direcções possíveis. Em seguida são determinadas a projecções horizontais e verticais das *bounding boxes*, [21], na Figura 8(a) e (c), respectivamente. Pode visualizar-se na projecção horizontal picos elevados nas zonas de texto seguidos de vales indicativos das zonas a segmentar. As projecções foram efectuadas ao nível das *bounding boxes* e não ao nível de cada píxel a branco por forma a diminuir o esforço computacional.



(a)

(b)



(c)

Figura 8. (a) Projecção horizontal das *bounding boxes* de (b), **(b)** Imagem mapeada com representação dos rectângulos para cada componente conectado e **(c)** Projecção vertical das *bounding boxes* de (b)

Para a projecção horizontal considerem-se as *bounding boxes* definidas por $BB = \{b1, b2, b3, ...b_N\}$ onde b_i inclui a região:

$$R_{b_i} = \{(x, y) \mid x_{b_{i,\min}} \le x \le x_{b_{i,\max}} \;\&\; y_{b_{i,\min}} \le y \le y_{b_{i,\max}}\} \qquad \textbf{(13)}$$

Para i=1,...,N.

A projecção horizontal da *bounding box* b_i consiste em associar b_i com a função escalar H_{b_i} a qual apresenta-se definida na equação seguinte:

$$H_{b_i} = \begin{cases} 1, & if\ (y_{b_{i,\min}} \le y \le y_{b_{i,\min}}) \\ 0, & else \end{cases} \qquad \textbf{(14)}$$

Com y=1..256.

De igual forma, a projecção vertical da *bounding box* b_i consiste em associar b_i com a função escalar V_{b_i} a qual apresenta-se definida na equação seguinte:

$$V_{b_i} = \begin{cases} 1, & if\ (x_{b_{i,\min}} \le x \le x_{b_{i,\min}}) \\ 0, & else \end{cases} \qquad \textbf{(15)}$$

Com x=1..256.

A projecção horizontal do conjunto de todas as *bounding boxes*, BB, é uma função H_b definida pela soma das projecções horizontais individuais de cada *bounding box*.

$$H_b = \sum_{i=1}^{N} H_{b_i} \qquad \textbf{(16)}$$

De igual modo, a projecção vertical horizontal do conjunto de todas as *bounding boxes*, BB, é uma função V_b definida pela soma das projecções verticais individuais de cada *bounding box*.

$$V_b = \sum_{i=1}^{N} V_{b_i} \qquad \textbf{(17)}$$

A figura seguinte ilustra como é calculada a projecção horizontal. A projecção vertical é calculada de modo similar. Conforme se pode visualizar na Figura 9, uma projecção é uma função cujos valores representam o número de *bounding boxes* que intersectam cada linha de projecção.

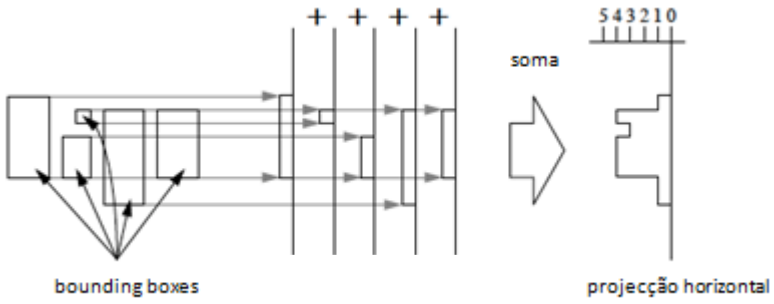


Figure 9. Esquema representativo da projecção horizontal

Em seguida, o algoritmo conhecido por algoritmo de corte recursivo X-Y (*recursive X-Y cut*) é implementado decompondo a imagem recursivamente em blocos rectangulares. O algoritmo consiste em determinar a cada passo as projecções horizontal e vertical definindo as linhas nos vales mais proeminentes de ambas as projecções e o processo continua até que não existam mais vales suficientemente proeminentes em ambas as projecções, estando nesta altura definidas as caixas finais que envolvem o texto, Figura 3(k). Horizontalmente foi definido efectuar a segmentação em vales nulos pois definir um limiar de corte tem a desvantagem de eliminar por exemplo acentos ou ainda parte de um caractere que se prolongue mais que o restantes contidos na mesma *bounding box*. Verticalmente foi definida a segmentação em vales com largura maior do que 13 pixeis por forma a evitar a segmentação a cada palavra de uma dada expressão.

2.7 Eliminação de falsos positivos

O processo de detecção de texto conduz a alguns falsos positivos, isto é, áreas detectadas que efectivamente não correspondem a texto. Na figura 3(k) visualizam-se duas caixas que não correspondem a zonas de texto da frame. Diversos métodos são propostos na literatura no sentido de eliminar estes falsos positivos e assim aumentar o desempenho dos algoritmos propostos. No entanto, depois de extensa pesquisa e testes levados a cabo de várias abordagens propostas, chegou-se à conclusão que a maioria são baseadas em regras heurísticas que dependem das imagens presentes na base de dados para efeitos de teste. Estas regras heurísticas eliminam caixas muito altas, com base na ideia que o texto que surge nas frames é praticamente todo horizontal, impedindo dessa forma a sua extracção, e caixas muito pequenas eliminando desta forma as que contém texto com fonte muito pequena. Outras abordagens baseiam-se no diferente comportamento dos contornos em blocos de texto e sem texto. No entanto, como na secção 6, é apresentada uma sugestão de melhoria dos algoritmos propostos baseada no exposto, optou-se por implementar um outro método [22].

A ideia deste algoritmo consiste em considerar a existência de dois tipos de contornos (*edges*) os que vêm de uma área vizinha exterior a uma caixa designados por extrínsecos e os que estão contidos numa caixa designados como intrínsecos. Desta forma, se a soma dos contornos que permanecem na caixa, uma vez apagados os que são extrínsecos e intrínsecos, for muito pequena é indicador de que essa caixa não contém texto e portanto deve ser eliminada. Uma vez que o texto contém contornos curtos e de pequenas dimensões e as imagens contidas no fundo contém contornos longos é facilmente perceptível que numa caixa não contendo texto, ao serem eliminados os que são extrínsecos, poucos ou nenhuns contornos permanecerão no interior dessa caixa, conduzindo à sua eliminação.

Passemos agora a explicar como foi implementada a ideia apresentada. Inicialmente, por forma a determinar quais os contornos extrínsecos e intrínsecos projectaram-se as caixas resultantes da segmentação sobre G(x,y), a frame em tons de cinza da Figura 3(b). Em seguida, e para cada caixa, considere-se uma caixa B uma expansão da caixa original A e uma caixa C uma contracção de A, de acordo com as equações seguintes:

$$Tamanho_B = Tamanho_A \times (1+\alpha) \qquad \textbf{(18)}$$

$$Tamanho_C = Tamanho_A \times (1-\alpha) \qquad \textbf{(19)}$$

Com $\alpha = 0.2$, o factor de escala.

Posteriormente procede-se à detecção de Canny para cada caixa B. Considerando-se agora, e ainda para cada caixa, o conjunto E de contornos que pertencem à área incluída pela caixa expandida B. Se um *edge* de E intersectar a fronteira de B e de A então significa que é um contorno extrínseco e deverá portanto ser apagado. Se um *edge* de E estiver totalmente incluído em C então deverá ser considerado intrínseco e também apagado. Um *edge* que intersecte A mas que não intersecte B deverá ser mantido. Finalmente, e uma vez apagados os contornos intrínsecos e extrínsecos, determinam-se as caixas a eliminar de acordo com a seguinte equação:

$$\frac{\text{Número de contornos que restam em A}}{\text{Número total de contornos em A}} < T \qquad \textbf{(20)}$$

Com T=0,15.

A decisão no factor de expansão, α, foi baseada em testes efectuados tendo essencialmente em consideração que muitas linhas de texto aparecem segmentadas com partes dos caracteres fora da caixa, uma vez que aquando o mapeamento esses pixeis não são considerados. Como tal pretende-se que a caixa maior, a caixa B inclua essas partes para que esses contornos não sejam considerados extrínsecos e consequentemente eliminados. Nos casos em que a segmentação não permitiu a definição de uma caixa, com as dimensões o mais próximas do mínimo necessário para envolver toda a informação, pode acontecer que a caixa C inclua toda a informação e consequentemente haja a eliminação desses contornos, por serem intrínsecos. O valor de T foi também escolhido com base em testes, denote-se no entanto que implica que restem apenas 15% dos contornos totais existentes, para cada caixa. Na Figura 10(d)-(e) pode visualizar-se a aplicação do método implementado para uma caixa sem texto e uma secção de uma caixa sem texto. Denote-se que apenas foi utilizada uma secção pois a caixa inteira não permitia uma clara visualização das caixas e dos contornos. Contudo, a secção escolhida exemplifica a eliminação de contornos intrínsecos e extrínsecos.

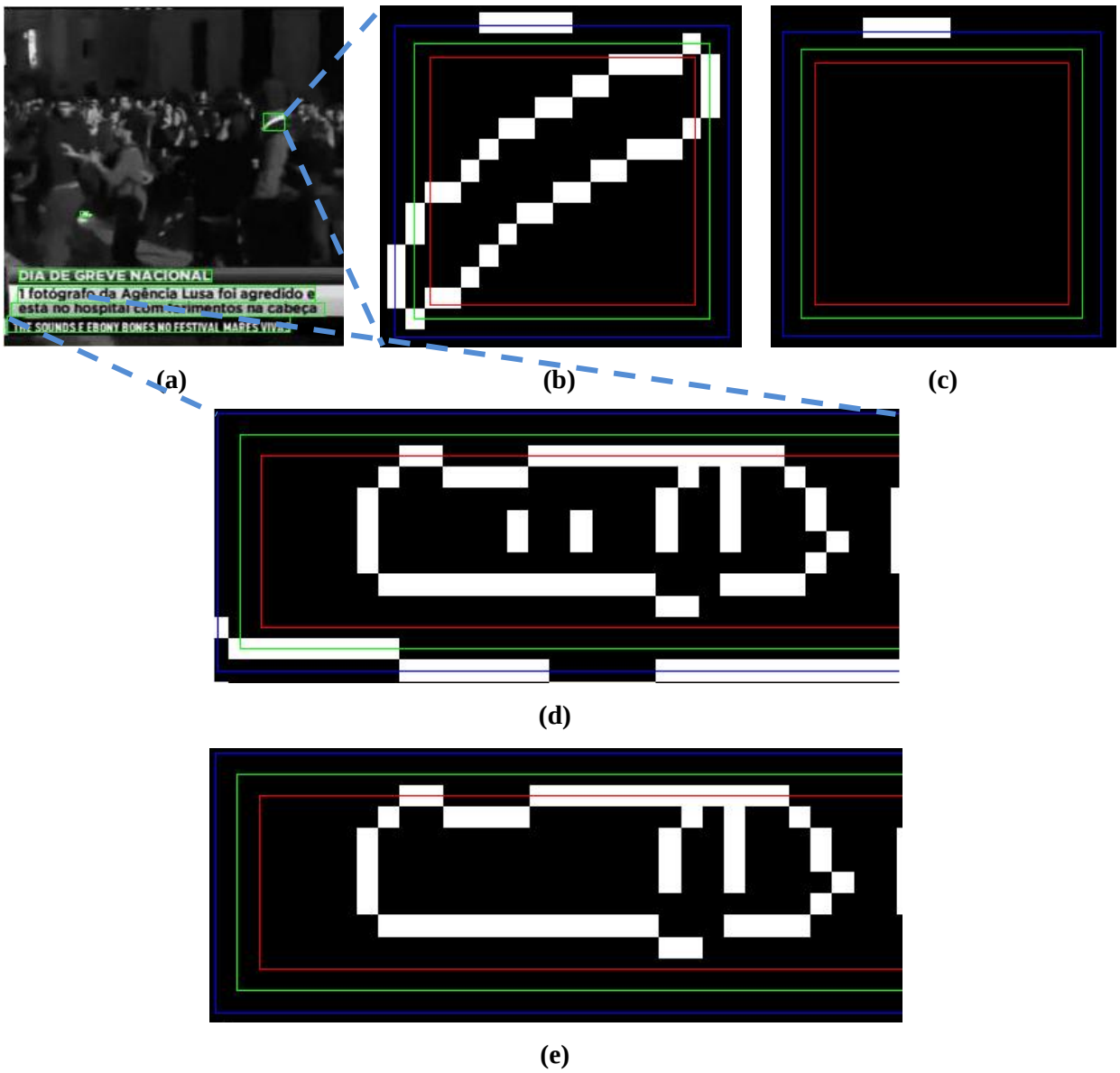


Figura 10. (a) G(x,y) com caixas resultantes do processo de segmentação, **(b)** Projecção de uma caixa sem texto, **(c)** Eliminação dos contornos extrínsecos e intrínsecos de (b), **(d)** Projecção de parte de uma caixa de texto e **(e)** Eliminação dos contornos intrínsecos e extrínsecos de (d)

Uma vez eliminados os falsos positivos o resultado final pode ser visualizado na Figura 3(l), na qual se pode verificar que o método de segmentação utilizado permitiu isolar individualmente cada uma das linhas de texto da frame original, Figura 3(a).

3. Detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB

Uma vez desenvolvido todo o código relativo à Transformada de Haar, anteriormente descrito, considerou-se necessário implementar novas abordagens para efeitos de comparação quanto ao desempenho que apresentavam na detecção de texto. Desta forma a detecção é agora

efectuada com base na Transformada de Fourier no espaço de cores RGB mantendo os restantes fases iguais às do primeiro algoritmo, permitindo a comparação. O algoritmo de reconhecimento de texto com base na Transformada de Fourier pode ser visualizado no fluxograma da figura seguinte.

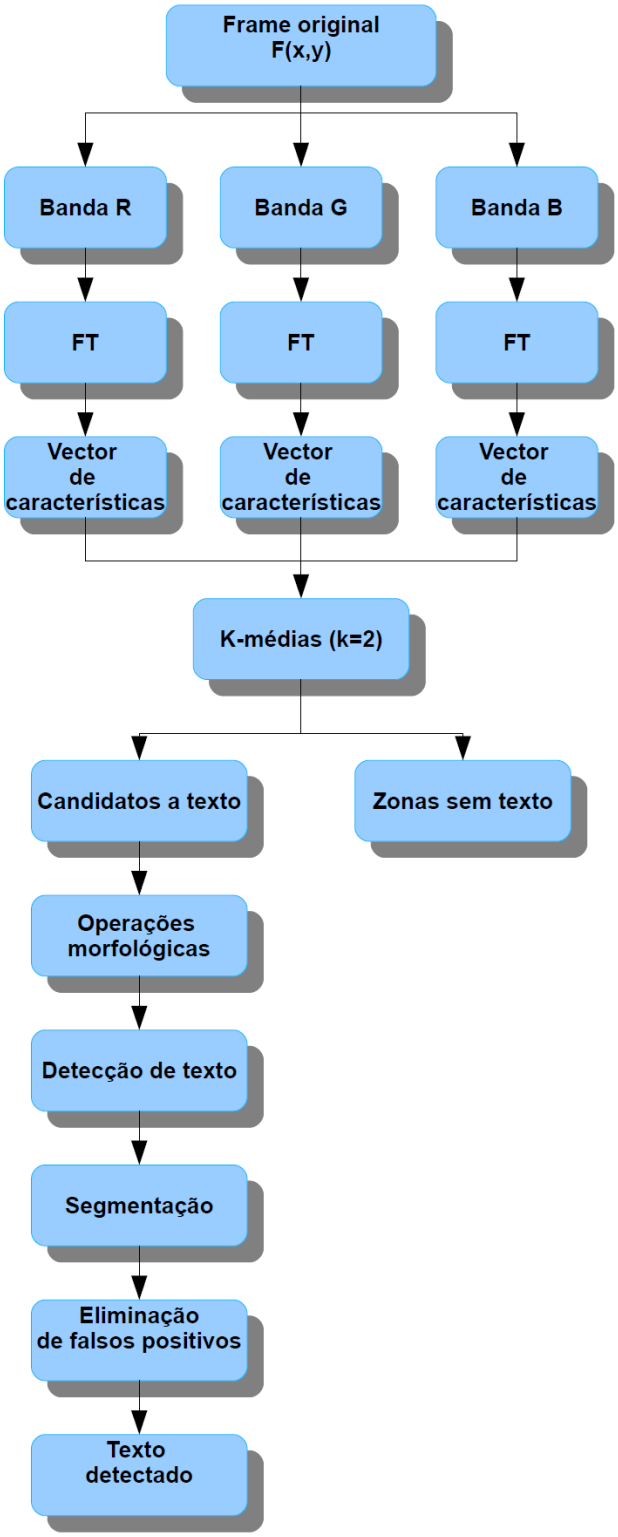


Figura 11. Fluxograma do algoritmo para detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB

A Transformada de Fourier bidimensional, 2D FT (*2D Fourier Transform*), é aplicada sobre as sub-bandas R, G e B. Experiências baseadas na FT demonstram que apresenta resultados positivos na diferenciação entre pixeis de texto e não texto em frames de vídeo, [17].

Na Figura 12(c) pode visualizar-se o módulo da Transformada de Fourier sobre a sub-banda R onde se podem distinguir os eixos vertical e horizontal os quais representam a frequência para cada direcção, respectivamente. Resultados similares foram obtidos para as bandas G e B, daí que se tenham omitido. Cada sub-banda é filtrada por forma a remover as frequências mais baixas conforme se pode visualizar na Figura 12(d).

Em seguida, cada sub-banda filtrada é reconstruída pela Transformada Inversa de Fourier, IFT (*Inverse Fourier Transform*), Figura 12(e). Na Figura 12(f) pode visualizar-se a representação dos valores absolutos de IFT, AIFT (*Absolute Inverse Fourier Transform*). Pixeis de texto apresentam valores negativos elevados em IFT, daí a recurso aos seus valores absolutos. A Figura 12(g) demonstra os resultados normalizados de AIFT, NAIFT (*Normalized Absolute Inverse Fourier Transform*) por divisão dos seus valores absolutos pelo seu máximo . Mais especificamente, os passos descritos são apresentados nas equações seguintes:

$$FT^k(x,y) = FT(F^k(x,y)) \quad \textbf{(21)}$$

$$IFT^k(x,y) = IFT(FT^k(x,y)) \quad \textbf{(22)}$$

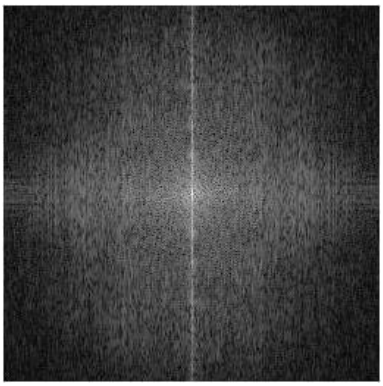
$$AIFT^k(x,y) = |\, IFT^k(x,y)\,| \quad \textbf{(23)}$$

$$NAIFT^k(x,y) = \frac{AIFT^k(x,y)}{\max(AIFT^k(x,y))} \quad \textbf{(24)}$$

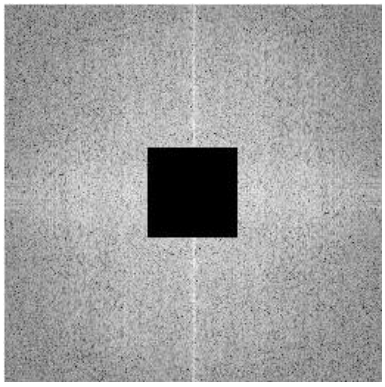
Com k=R,G,B.

(a) Imagem original

(b) Sub-banda R



(c) Módulo de FT para R

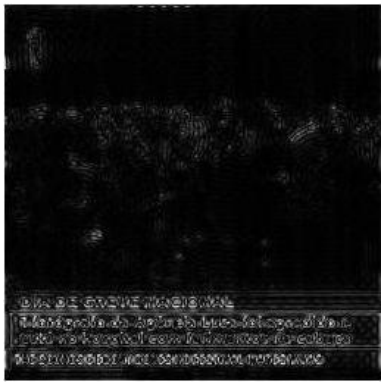


(d) FT Filtrada para R

(e) IFT para R

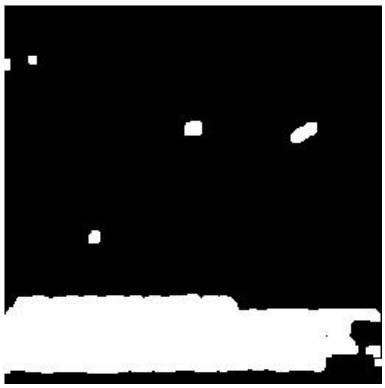
(f) AIFT para R

(g) NAIFT para R



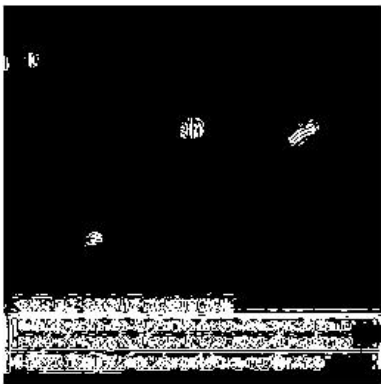
(h) Imagem média

(i) Áreas candidatas a texto

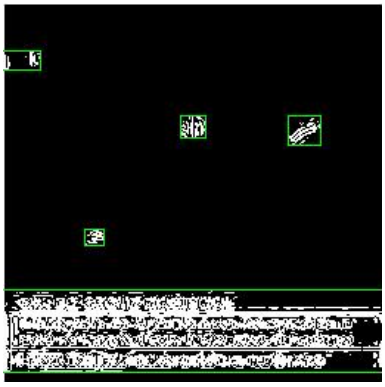


(j) Abertura morfológica

(k) Dilatação morfológica



(l) Imagem mapeada



(m) Segmentação

(n) Texto extraído

Figura 12. Passos intermediários no processo de detecção de texto pela Transformada de Fourier sobre o espaço de cores RGB

Os restantes passos do algoritmo foram calculados da mesma forma que os do algoritmo de detecção da secção anterior. O vector de características foi agora calculado sobre NAIFT das sub-bandas R,G e B da imagem original F(x,y) e a imagem média, na Figura 12(h), é portanto a média de NAIFT para as três sub-bandas.

A motivação para o uso da Transformada de Fourier sobre RGB consiste nos diferentes valores dos coeficientes da transformada para pixeis de texto e de não texto em cada sub-banda, conforme se pode visualizar na Figura 13(a)-(c). A figura 13(d) é a imagem média das três sub-bandas. Podem ainda visualizar-se os coeficientes da FT na imagem média (Figura 14(a)) para uma linha de texto (217 na Figura 14(b)) e para uma linha sem texto (54 na Figura 14(c)), os quais claramente apresentam maior expressividade para a linha de texto. Tal como na secção anterior o cálculo do limiar T, para a obtenção da imagem mapeada, Figura 12(l), foi efectuado para valores de frequência, na imagem média da Figura 12(h), superiores ou iguais a 0,05.

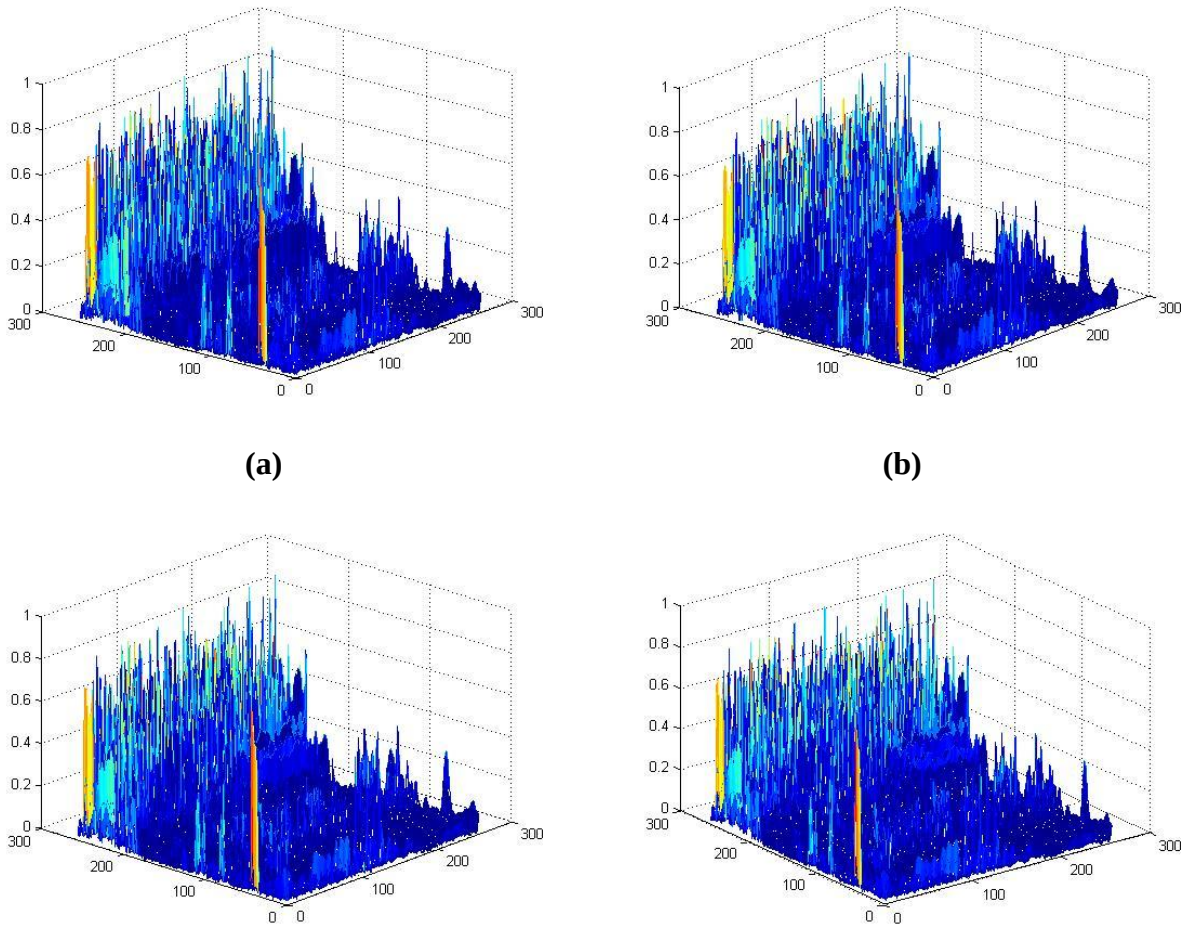
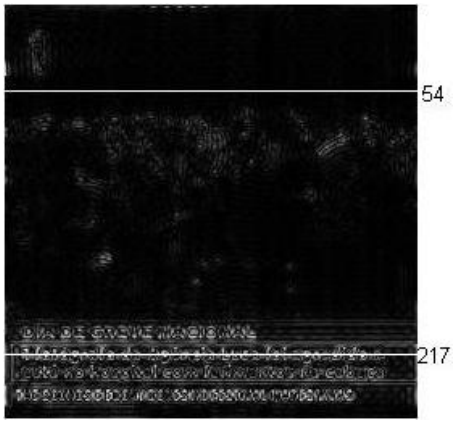
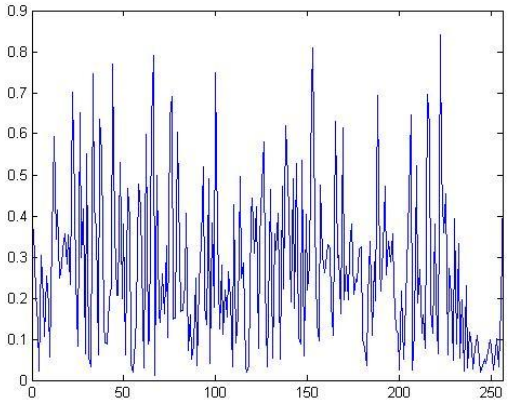
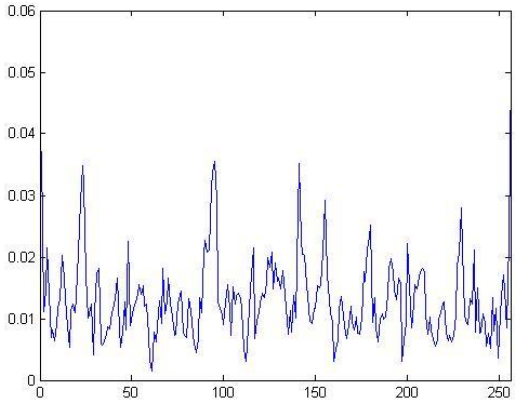


Figura 13. Representação 3D dos coeficientes de NAIFT **(a)** da sub-banda R, **(b)** da sub-banda G, **(c)** da sub-banda B e **(d)** da banda média Avg



(a)





(b) **(c)**

Figura 14. (a) Coeficientes de NAIFT na imagem média na Figura 12(h), **(b)** Perfil na linha 217 de (a), linha de texto e **(c)** Perfil na linha 54 de (a), linha sem texto

4. Detecção de texto baseada no comportamento dos contornos de Sobel sobre o espaço de cores RGB

Uma outra abordagem foi ainda desenvolvida para a detecção de texto, baseada no detector de contornos de Sobel sobre o espaço de cores RGB. Estudos literários [13] demonstram a efectividade na detecção de texto gráfico e de cena com base neste método. As restantes fases são iguais às dos algoritmos anteriores, permitindo assim comparar os três algoritmos, conforme se pode visualizar no fluxograma da figura seguinte:

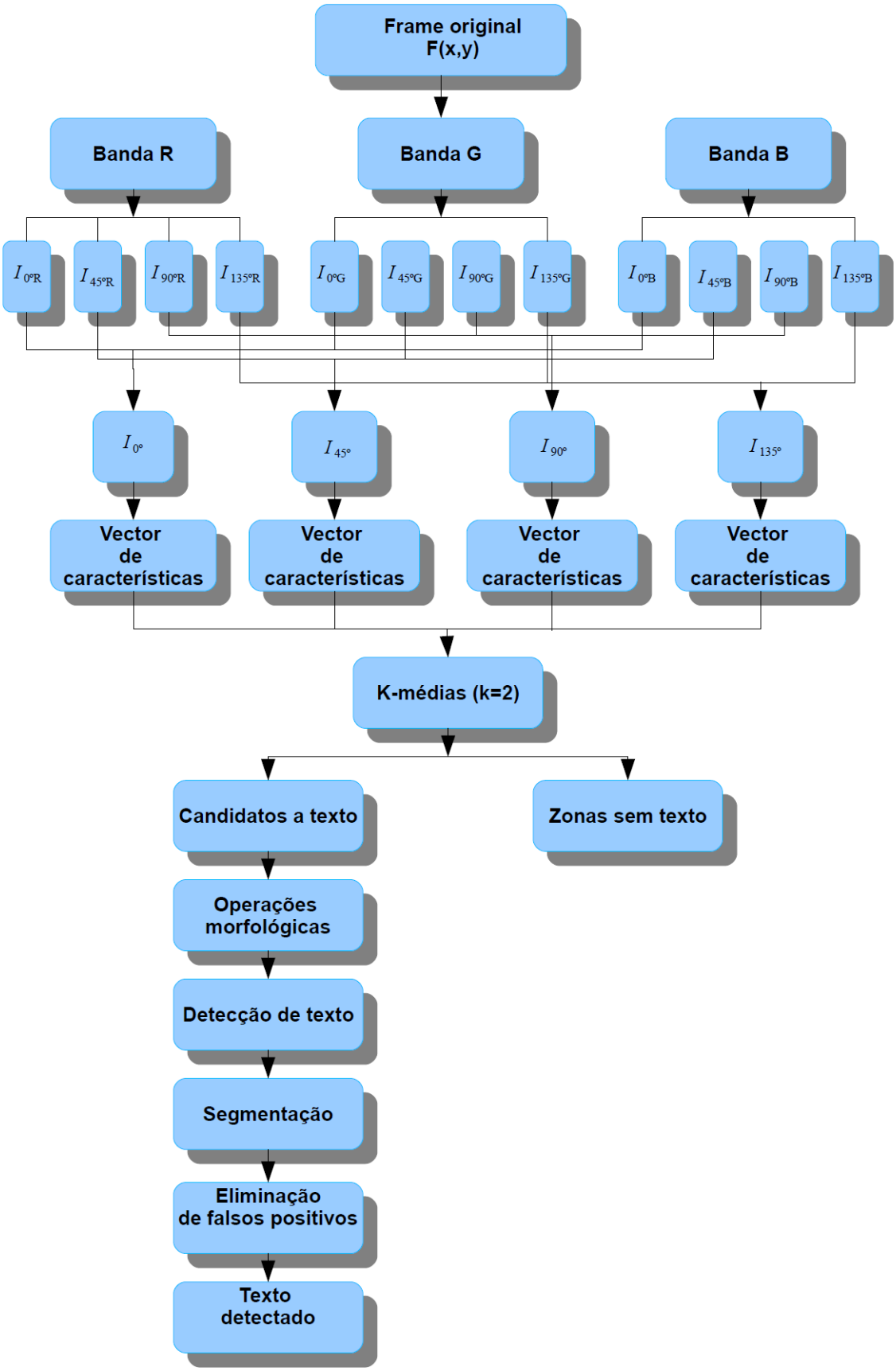


Figura 15. Fluxograma do algoritmo para detecção de texto baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB

A ideia consiste em detectar os contornos de Sobel em quatro direcções distintas, 0^o, 90^o, 45^o e 135^o, sobre cada sub-banda R, G e B. As direcções escolhidas consistem no facto de o texto ser essencialmente constituído por traços dispostos na horizontal, vertical, diagonal direita pra cima e diagonal esquerda para cima.

Considere-se então i=R,G,B como cada sub-banda do espaço de cores e $\theta = 0^0, 45^0, 90^0, 135^0$, para cada uma das direcções especificadas. Então a média para cada para cada uma das direcções consideradas, I_θ, Figura 16 (b)-(e), é definida por:

$$I_\theta = \sqrt{\frac{1}{3}\sum_{i=R,G,B} I_{\theta i}^2} \qquad \textbf{(25)}$$

Onde, $I_{\theta i}$ consiste no mapa de contornos da cor i na direcção θ.

(a) Imagem original

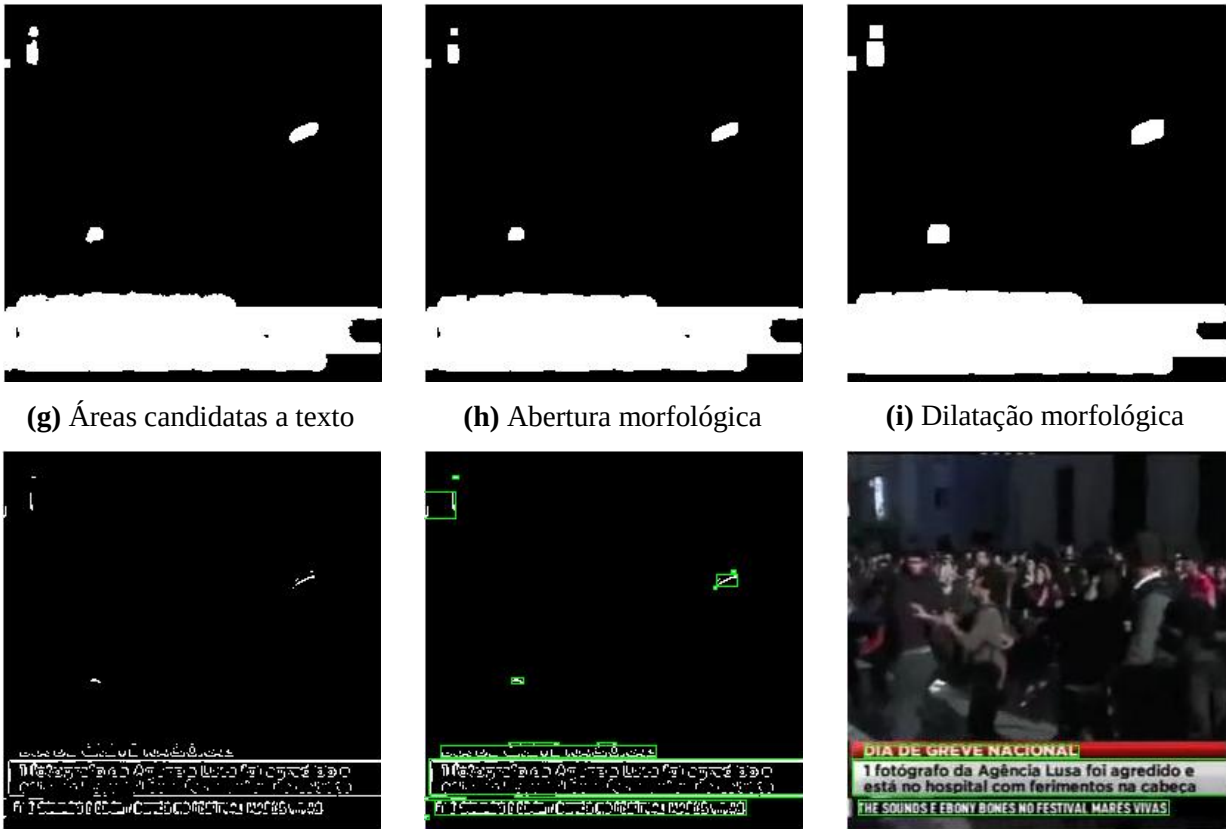
(b) I_{0^0}

(c) I_{45^0}

(d) I_{90^0}

(e) I_{135^0}

(f) Imagem média



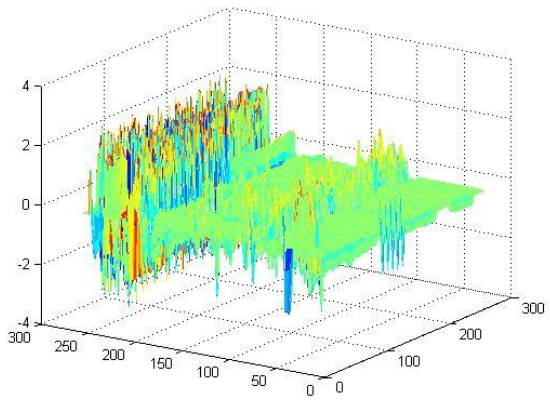
(g) Áreas candidatas a texto **(h)** Abertura morfológica **(i)** Dilatação morfológica

(j) Imagem mapeada **(k)** Segmentação **(l)** Texto extraído

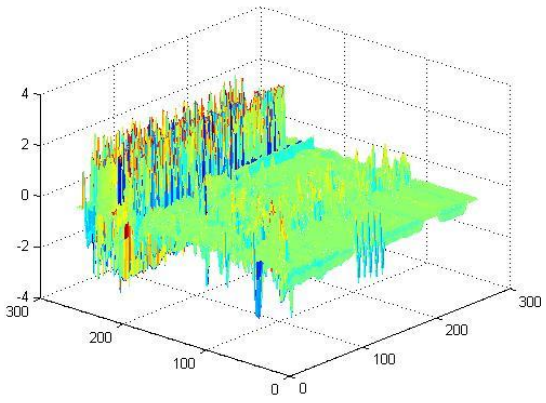
Figura 16. Passos intermediários no processo de detecção de texto baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB

Os restantes passos do algoritmo foram calculados da mesma forma que os dos algoritmos de detecção das secções anteriores. O vector de características foi agora calculado sobre I_θ para as 4 direcções consideradas e a imagem média, Figura 16(f), é portanto a média de I_θ para $\theta = 0^0, 45^0, 90^0, 135^0$.

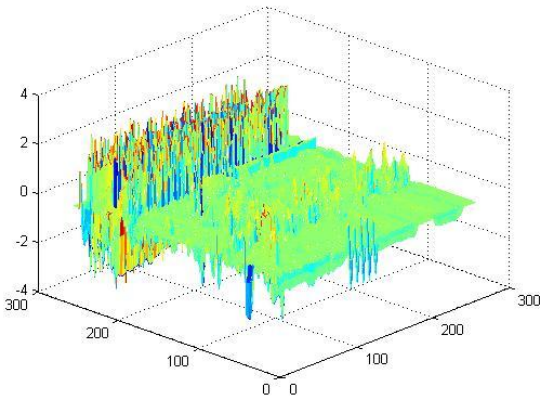
A motivação para o uso do detector de Sobel sobre o espaço de cores RGB consiste nos diferentes valores que apresenta para pixeis de texto e de não texto em cada sub-banda, conforme se pode visualizar na Figura 17(a)-(c), para $\theta = 135^0$. Podem ainda visualizar-se os valores de frequência na imagem média (Figura 18(a)) para uma linha de texto (217 na Figura 18(b)) e para uma linha sem texto (54 na Figura 18(c)), os quais apresentam maior expressividade para a linha de texto. O limiar T, para a obtenção da imagem mapeada, Figura 16(j), foi calculado para valores de gradiente, na imagem média da Figura 16(f), maiores ou iguais que 1.



(a) $I_{135^0,R}$



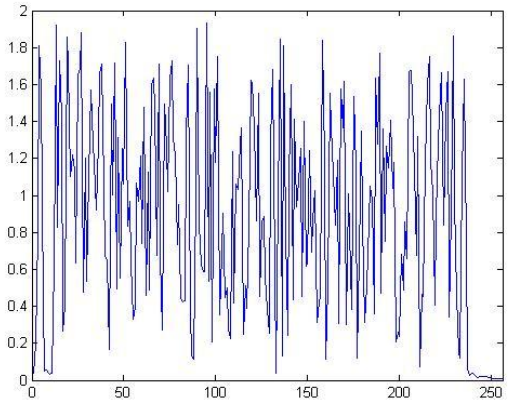
(b) $I_{135^0,G}$

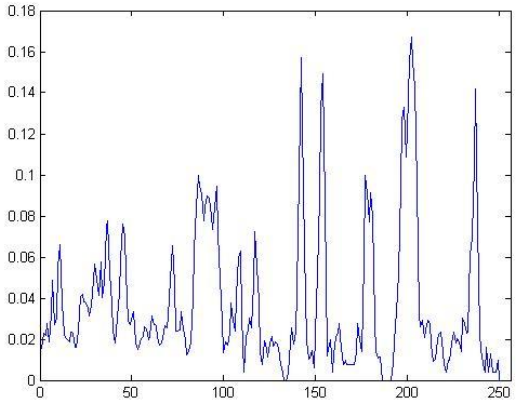


(c) $I_{135^0,B}$

Figura 17. Representação 3D para $\theta = 135^o$ **(a)** na sub-banda R, **(b)** na sub-banda G e **(c)** na sub-banda B

(a)





(b) **(c)**

Figura 18. (a) Valores do gradiente na imagem média da Figura 16(f), **(b)** Perfil na linha 217 de (a), linha de texto e **(c)** Perfil na linha 54 de (a), linha sem texto.

5. Pré-Processamento

Uma das falhas que apresentam os algoritmos de detecção de texto apresentados, assim como outros presentes na literatura, consiste no facto de não serem testados para frames que não contém texto. Desta forma os algoritmos desenvolvidos foram deliberadamente testados para frames sem texto conduzindo à detecção de falsos positivos. É então apresentada uma abordagem [23] que permite classificar as frames antes de uma posterior extracção de texto, minimizando a detecção de falsos positivos, daí a designação atribuída de pré-processamento. A ideia consiste em dividir uma dada frame em blocos e efectuar uma classificação designada por método Máx-Min. Este método permite a classificação de cada bloco enquanto um bloco de texto ou um bloco sem texto. Em seguida, cada bloco classificado como de texto é novamente avaliado de acordo com um conjunto de regras, designadas por R1, R2 e R3. O pré-processamento encontra-se representado no fluxograma da Figura 19.

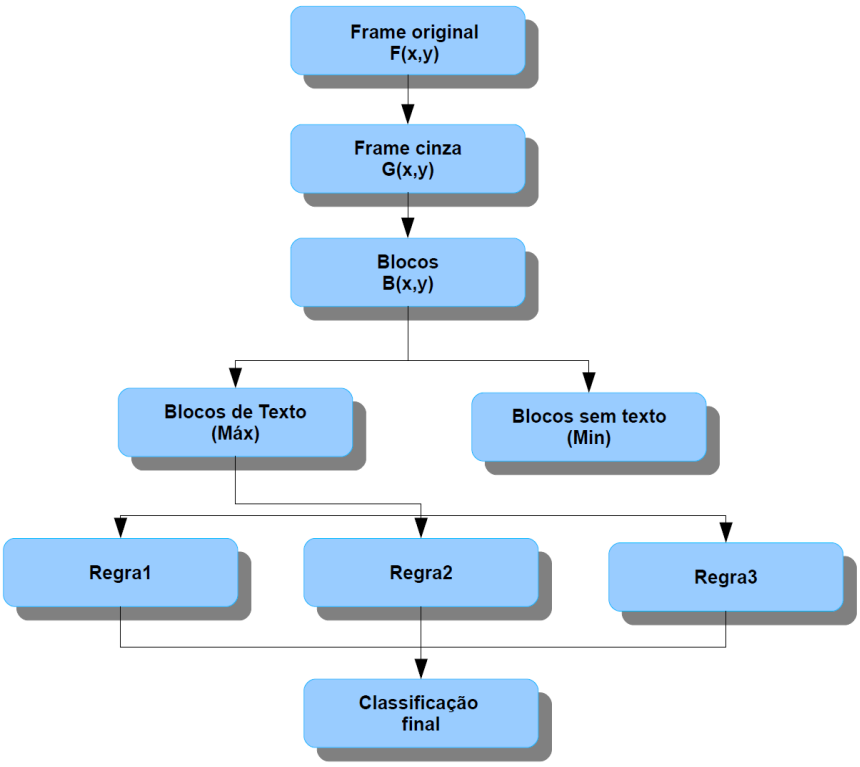


Figura 19. Fluxograma do Pré-Processamento

5.1. Método Máx-Min

Numa frame o texto é um fenómeno local, isto é, encontra-se normalmente concentrado em uma ou mais zonas e não difundido por toda a frame com caracteres de fonte maior. Com base nesta premissa, a classificação de uma frame baseia-se em pequenas características locais. Desta forma, a frame em escala de cinzas de dimensões 256x256, Figura 20(a) é dividida em 16 blocos de igual dimensão (64x64), Figura 20(b). A divisão em 16 blocos permite determinar características que permitam identificar texto mais rapidamente do que processando toda a frame. A escolha da dimensão referida para os blocos baseia-se na necessidade de cada bloco conter um número de caracteres suficiente que permitam identificar as características. Na figura 20(b) os blocos 13,14,15 e 16 contém texto e os restantes blocos não.

(a) Frame **(b) Blocos**

Figura 20. Divisão em blocos

Considere-se agora o bloco 15 como um exemplo de um bloco com texto e o bloco 5 como um exemplo de um bloco sem texto. Com efeito, visualizando os contornos (*edges*) de Sobel para ambos os blocos nas Figuras 21(e) e (b), respectivamente, denota-se que no bloco de texto os contornos são mais rectos e densos ao passo que no bloco 5 são mais cursivos. Com base nesta observação filtrando os contornos cursivos dos blocos de Sobel e verificando a quantidade dos contornos que permanecem, quantidade de contornos rectos, poderá classificar-se cada bloco quanto à presença de texto.

Desta forma, para classificar os contornos em cursivos ou rectos considere-se, $\overline{X} = \{x_1, x_2, ..., x_n\}$ e $\overline{Y} = \{y_1, y_2, ..., y_n\}$ o conjunto das coordenadas x e y, respectivamente, dos pixeis dos contornos de Sobel, para cada bloco. O centróide de cada contorno, (C_x,C_y), é definido por:

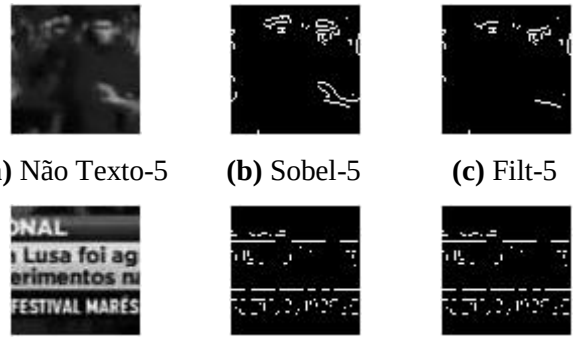
$$C_x = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \textbf{(26)}$$

$$C_y = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad \textbf{(27)}$$

Onde n é o número de pixeis no contorno. Assim, com base na definição do centróide, um contorno é considerado recto, Rec_Edge, se:

$$\mathrm{Re}c\_Edge = \begin{cases} 1, & (C_x \in X) \cap (C_y \in Y) \\ 0, & else \end{cases} \qquad \textbf{(28)}$$

Com base nesta classificação os contornos cursivos são filtrados, Figuras 21(c) e (f). Pode observar-se que os contornos rectos no bloco de texto 15 aparecem em maior quantidade do que no bloco sem texto, bloco 6, o que valida o bloco 15 como um bloco de texto.



(a) Não Texto-5 **(b)** Sobel-5 **(c)** Filt-5

(d) Texto-15 **(e)** Sobel-15 **(f)** Filt-15

Figura 21. Blocos 5 e 15

Considere-se ainda N_Rect como o vector que contém o número de contornos rectos em cada bloco. Por forma a facilitar a comparação, N_Rect é normalizado entre 0 e 1, $N = \{n_1, n_{2,}..., n_{16}\}$. É calculada a média, Med, do valor máximo e mínimo em N por:

$$Med = \frac{\max(N) + \min(N)}{2} \qquad \textbf{(29)}$$

E um bloco é considerado de texto se:

$$Bloco\_Texto = \begin{cases} 1, & if\ (n(i) \geq Med) \\ 0, & else \end{cases} \qquad \textbf{(30)}$$

A Figura 22 demonstra que os blocos 13,14,15 e 16 são classificados enquanto blocos de texto e os restantes blocos enquanto blocos sem texto.

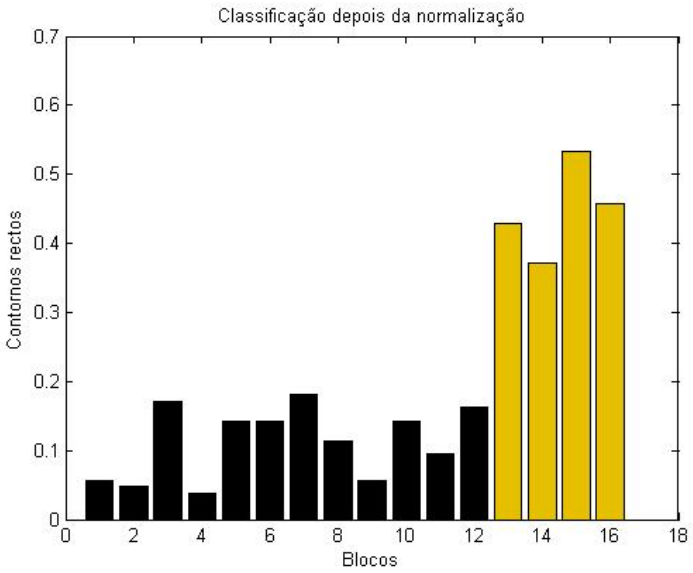
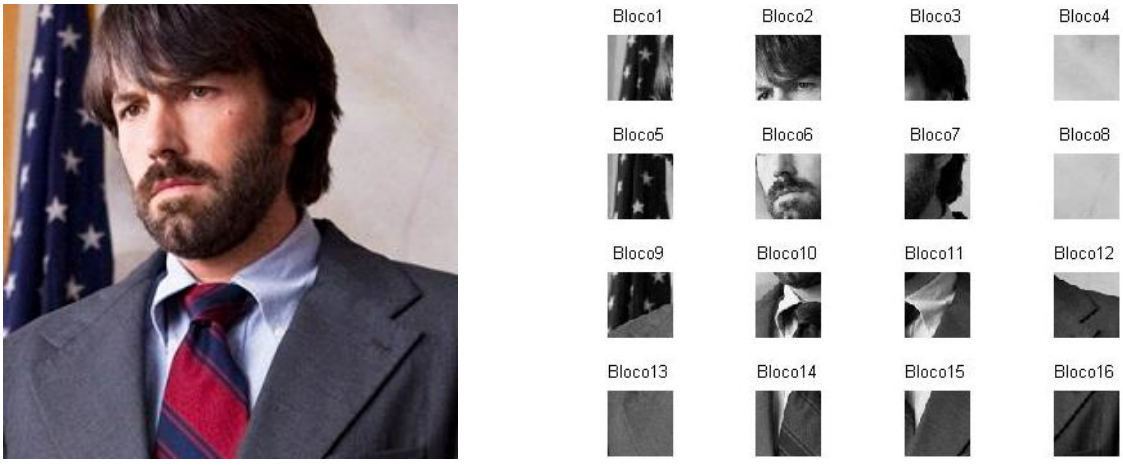
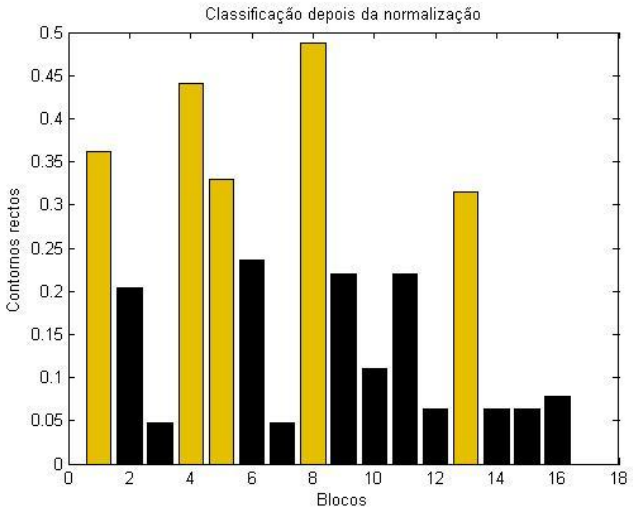


Figura 22. Classificação Máx-Min

O exemplo apresentado não sugere a necessidade de uma avaliação posterior pelas três regras já que a classificação Máx-Min determina correctamente quais os blocos de texto. No entanto, para uma frame sem texto, como a classificação Máx-Min é efectuada com base na média do número de contornos rectos em cada bloco haverá sempre blocos identificados como potenciais candidatos a blocos de texto, conforme se pode visualizar na Figura 23(a)-(c). Além do mais, mesmo para outras frames de texto, como a da Figura 24(a), nem sempre a classificação identifica correctamente quais são os blocos com texto e sem texto, como é demonstrado na Figura 24(a)-(c).

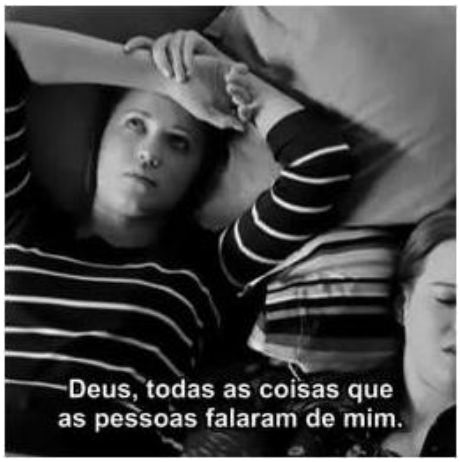


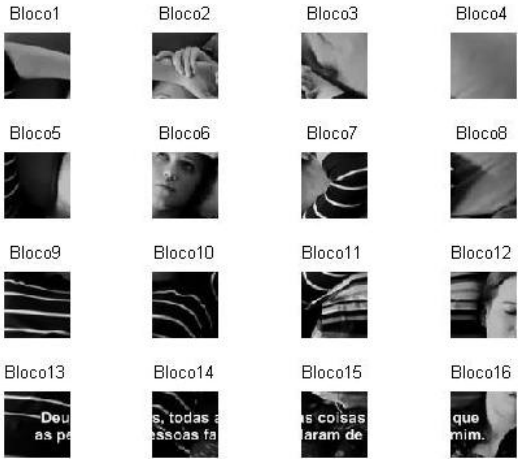
(a) **(b)**



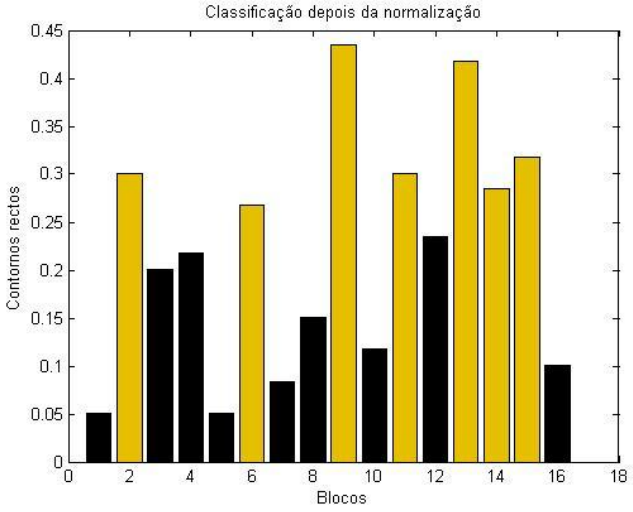
(c)

Figura 23. (a) Frame sem texto, **(b)** Blocos e **(c)** Classificação Máx-Min





(a) **(b)**



(c)

Figura 24. (a) Frame com texto, **(b)** Blocos e **(c)** Classificação Máx-Min

Os blocos classificados pelo Máx-Min são agora classificados por três regras as quais se designam por R1,R2 e R3 e cuja respectiva implementação é apresentada em seguida. Estas regras são formuladas por observação das propriedades locais dos contornos.

5.2. R1

A nitidez é uma característica inerente ao texto na maioria das frames uma vez que o mesmo contém informação dirigida aos telespectadores. Desta forma com base na premissa de que a nitidez é maior para os contornos de texto do que os contornos de não texto é formulada R1.

Inicialmente é aplicado o filtro aritmético, AF(Arithmetic Filter), o qual produz as imagens com aspecto embaçado da Figura 26(a) e (f). Em seguida os blocos são filtrados por um filtro mediano, MF (Median Filter) por forma a reduzir o ruído, Figuras 26(b) e (g). Posteriormente é aplicado o detector de Sobel na imagem filtrada por AF, obtendo-se $Sobel_{AF}$, representado nas Figuras 26(d) e (i). A subtracção entre MF e AF dá origem a Diff, nas Figuras 26(c) e (h). Por fim o detector de Canny é então aplicado em Diff originando as imagens das Figuras 26(e) e (j), para um bloco sem texto e um bloco com texto, respectivamente. Os passos descritos podem resumir-se de acordo com o fluxograma da Figura 25:

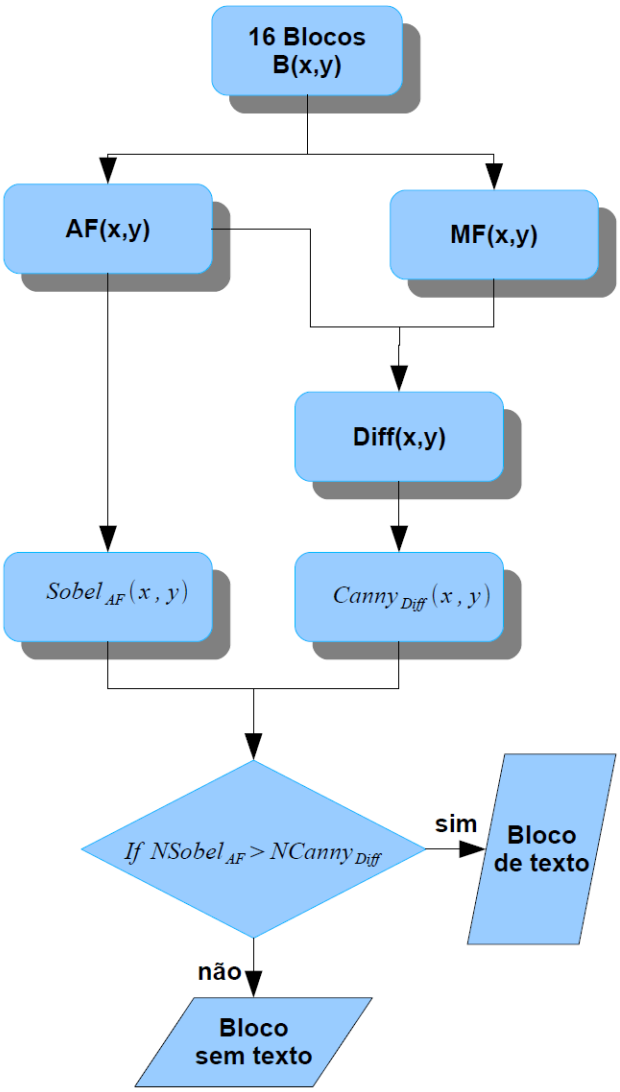


Figura 25. Fluxograma representativo de R1

Observando as Figuras 26(f)-(j) denota-se que com o operador de Sobel há mais contornos quando comparado com o operador de Canny, embora por inspecção visual a sensação seja distinta já que os contornos deste último são mais longos e como tal ocupam mais a imagem. Para o bloco 5, sem texto, e observando as Figuras 26(a)-(e), com os mesmos filtros o resultado é o oposto.

(a) AF-5 **(b)** MF-5 **(c)** Diff-5 **(d)** $Sobel_{AF}$-5 **(e)** $Canny_{Diff}$-5

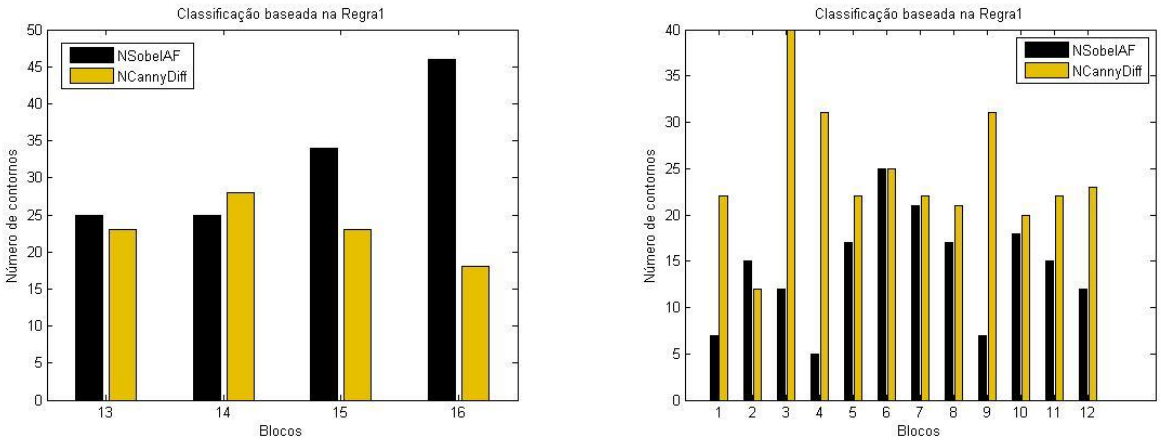
(f) AF-15 **(g)** MF-15 **(h)** Diff-15 **(i)** $Sobel_{AF}$-15 **(j)** $Canny_{Diff}$-15

Figura 26. Passos da classificação com base em R1

Considere-se $NSobel_{AF}$ e $NCanny_{Diff}$ o número de contornos em $Sobel_{AF}(x,y)$ e $Canny_{Diff}(x,y)$, respectivamente então:

$$R1 = \begin{cases} \text{Bloco de Texto}, & \text{if}(Sobel_{AF} > Canny_{Diff}) \\ \text{Bloco sem texto}, & \text{else} \end{cases} \quad \textbf{(31)}$$

Os resultados para os blocos classificados pelo Máx-Min são apresentados no gráfico da Figura 27(a) onde se pode verificar que os blocos 13, 15 e 16 são correctamente classificados por R1 e o bloco 14 incorrectamente classificado. Para o bloco 14 $NSobel_{AF}$ é menor do que $NCanny_{Diff}$ ao contrário do que acontece para os blocos 13, 15 e 16 conforme se pode visualizar na Figura 28(a)-(f). Na Figura 27(b) são apresentados os resultados para os restantes blocos para efeitos de verificação da efectividade desta regra. Denota-se que o bloco 2 é erradamente classificado como texto e os restantes blocos correctamente classificados como sem texto.



(a) **(b)**

Figura 27. Classificação com base em R1 **(a)** para os blocos classificados pelo Máx-Min **(b)** para os restantes blocos

(a) Sobel$_{AF}$-13 **(b)**Canny$_{Diff}$-13 **(c)**Sobel$_{AF}$-14 **(d)**Canny$_{Diff}$-14 **(e)**Sobel$_{AF}$-16 **(f)**Canny$_{Diff}$-16

Figura 28. NSobel$_{AF}$ e NCanny$_{Diff}$ para os blocos 13, 14 e 16

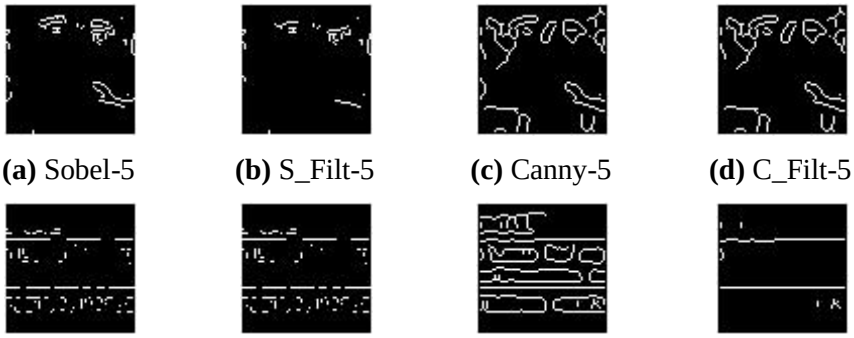
Em seguida procede-se a classificação, dos blocos classificados pelo método Máx-Min, de acordo com R2.

5.3. R2

Tanto o detector de Sobel como o de Canny definem inúmeros contornos num bloco de texto, no entanto esses contornos exibem uma aparência distinta para ambos os detectores. Os contornos de Sobel, num bloco de texto, tendem a ser de pequena dimensão e maioritariamente apresentam-se rectos dispostos na horizontal ou verticalmente. Por outro lado, os contornos de Canny, para um bloco de texto, tendem a ser alongados e cursivos nas suas terminações. Em blocos sem texto estas observações não se verificam com a mesma persistência. Com base nestas premissas é possível formular R2. A ideia consiste em filtrar cada bloco de Sobel e de Canny por forma a remover os contornos cursivos e em seguida somar os comprimentos, o número total de pixeis, dos contornos rectos. Uma grande soma num bloco será indicativa da presença de texto no mesmo. Na secção 6.1 demonstrou-se o procedimento, com base no cálculo do centróide de cada contorno, que permite eliminar os contornos cursivos.

Considerem-se então novamente os blocos 5 como um exemplo de um bloco sem texto e o bloco 15 como um exemplo para um bloco de texto. Na Figura 29(a)-(d), para o bloco 5, pode visualizar-se o bloco de Sobel e o de Canny e os respectivos blocos filtrados de contornos cursivos. O mesmo se pode visualizar para o bloco 15 na Figura 29(e)-(h). A soma dos comprimentos dos contornos do bloco de Sobel filtrado designa-se por S_S e a mesma soma para o bloco de Canny filtrado de contornos cursivos designa-se por S_C. Um bloco é então classificado como um bloco de texto por R2 se:

$$\mathrm{R2} = \begin{cases} \text{Bloco de Texto}, & \text{if } (\mathrm{S\_S} > S\_C) \\ \text{Bloco sem texto}, & \text{else} \end{cases} \qquad \textbf{(32)}$$

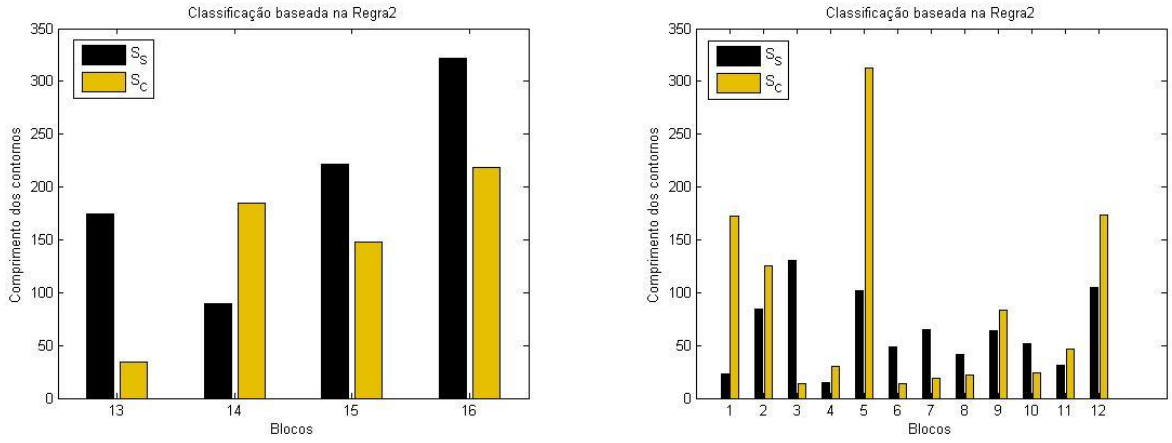


(a) Sobel-5 **(b)** S_Filt-5 **(c)** Canny-5 **(d)** C_Filt-5

(e) Sobel-15 **(f)** S_Filt-15 **(g)** Canny-15 **(h)** C_Filt-15

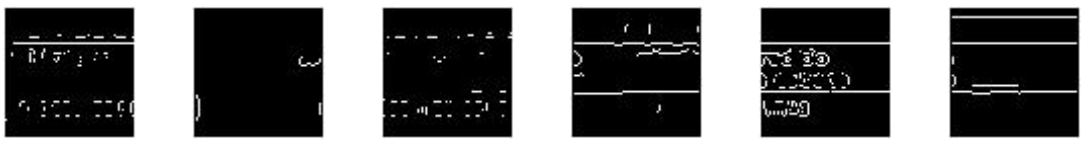
Figura 29. (a)-(d) Bloco 5 sem texto e **(e)-(h)** Bloco 15 de texto

Os resultados para os blocos classificados pelo Máx-Min são apresentados no gráfico da Figura 30(a) onde se pode verificar que os blocos 13, 15 e 16 são correctamente classificados por R2 e o bloco 14 incorrectamente classificado. Para o bloco 14 S_S é menor do que S_C ao contrário do que acontece para os blocos 13, 15 e 16 conforme se pode visualizar na Figura 31(a)-(f). Na Figura 30(b) são apresentados os resultados para os restantes blocos onde os blocos 3, 6, 7, 8 e 10 são erradamente classificados como blocos de texto e os restantes blocos correctamente classificados como sem texto.



(a) **(b)**

Figura 30. Classificação com base em R2 **(a)** para os blocos classificados pelo Máx-Min **(b)** para os restantes blocos



(a)S_Filt-13 **(b)**C_Filt-13 **(c)**S_Filt-14 **(d)**C_Filt-14 **(e)**S_Filt-16 **(f)**C_Filt-16

Figura 31. S_S e S_C para os blocos 13, 14 e 16

Procede-se seguidamente à classificação dos mesmos blocos de acordo com R3. Denote-se que no caso de serem classificados caracteres Árabes ou Persas esta regra não será a mais

indicada devido à sua natureza cursiva. No entanto, não foram consideradas frames com esses caracteres e para esses casos R1 e R3 poderão na mesma ser regras válidas e podem derivar-se novas regras caso seja necessário.

5.4. R3

Além do facto de os contornos num bloco de texto apresentarem-se mais rectos, pode também visualizar-se que os mesmos apresentam uma proximidade consistente e regular devido ao agrupamento dos caracteres num bloco de texto, Figura 21(e). O mesmo não se verifica num bloco sem texto, Figura 21(b). Considere-se então o centróide de cada contorno no bloco de Sobel filtrado, tal como descrito na secção 6.1. A distância euclidiana entre os centroídes dos contornos no bloco é estimada por:

$$Dist(i,j) = \sqrt{(Cx_j - Cx_i)^2 + (Cy_j - Cy_i)^2} \quad \textbf{(33)}$$

Com Cx e Cy as coordenadas dos centróides dos contornos e i e j a variar de 1..n com n o número de contornos no bloco filtrado. Desta forma obtém-se uma matriz de proximidade para todos os contornos no bloco de Sobel filtrado. Esta matriz é normalizada:

$$NDist(i,j) = \frac{Dist(i,j)}{\max(Dist(i,j))} \quad \textbf{(34)}$$

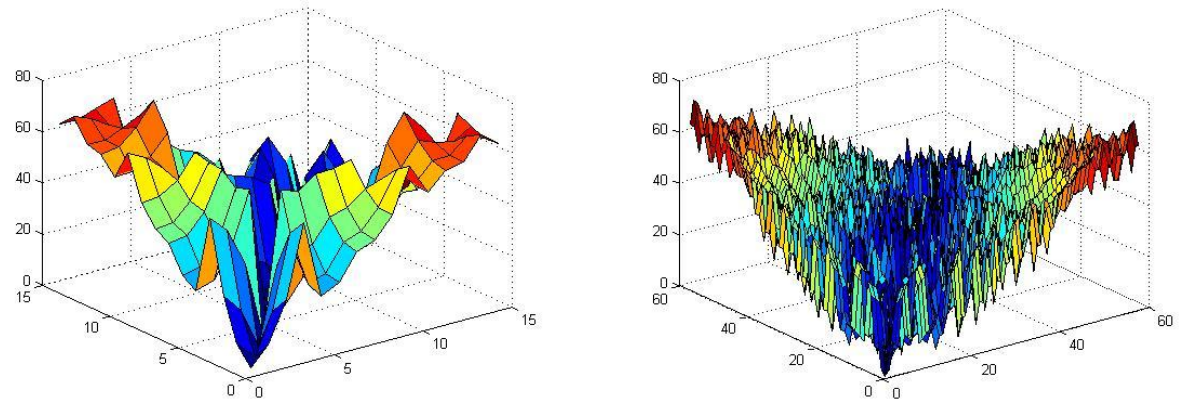
Posteriormente para medir a distância entre os centróides é determinado o desvio padrão de NDist(i,j) o qual se designa por Std (*Standard Desviation*). Então, um bloco de texto com o seus contornos regularmente próximos apresenta um valor menor de Std em comparação com o mesmo valor para um bloco sem texto. As matrizes de distâncias para o bloco 5 sem texto e o bloco 15 de texto podem ser visualizadas nas Figura 32 (a) e (b), respectivamente, onde se denota a proximidade regular e consistente para o bloco de texto.

A regra 3 fica então definida por:

$$R3 = \begin{cases} \text{Bloco de Texto,} & \text{if } (Std < 0.09) \\ \text{Bloco sem texto,} & \text{else} \end{cases} \quad \textbf{(35)}$$

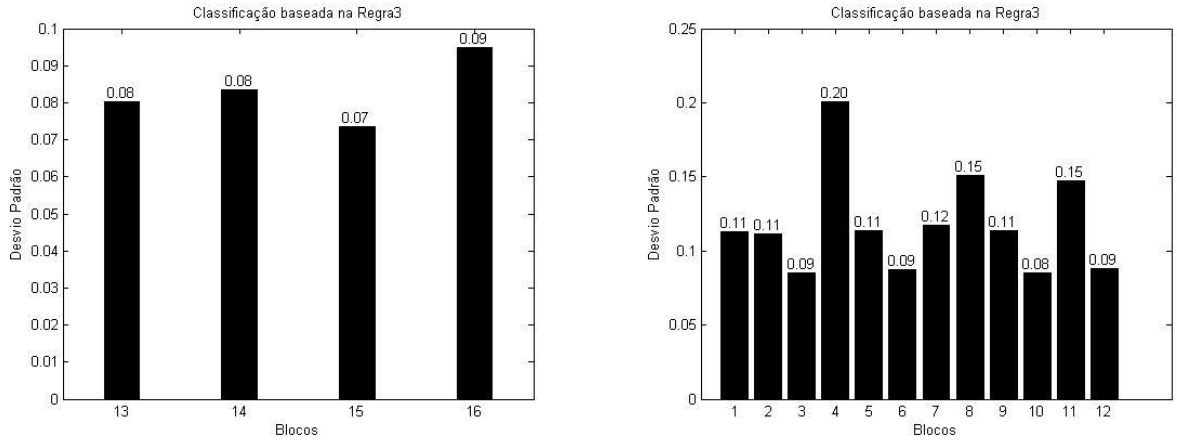
Os resultados para os blocos classificados pelo Máx-Min são apresentados no gráfico da Figura 33(a) onde se pode verificar que os blocos 13, 14 e 15 são correctamente classificados por R3 e o bloco 16 incorrectamente classificado. Para o bloco 16 Std>=0.09 ao contrário do que acontece para os blocos

13, 14 e 15 conforme se pode visualizar na Figura 34(a)-(c). Na Figura 33(b) são apresentados os resultados para os restantes blocos onde o bloco 10 é incorrectamente classificado como bloco de texto e os restantes blocos correctamente classificados como sem texto.



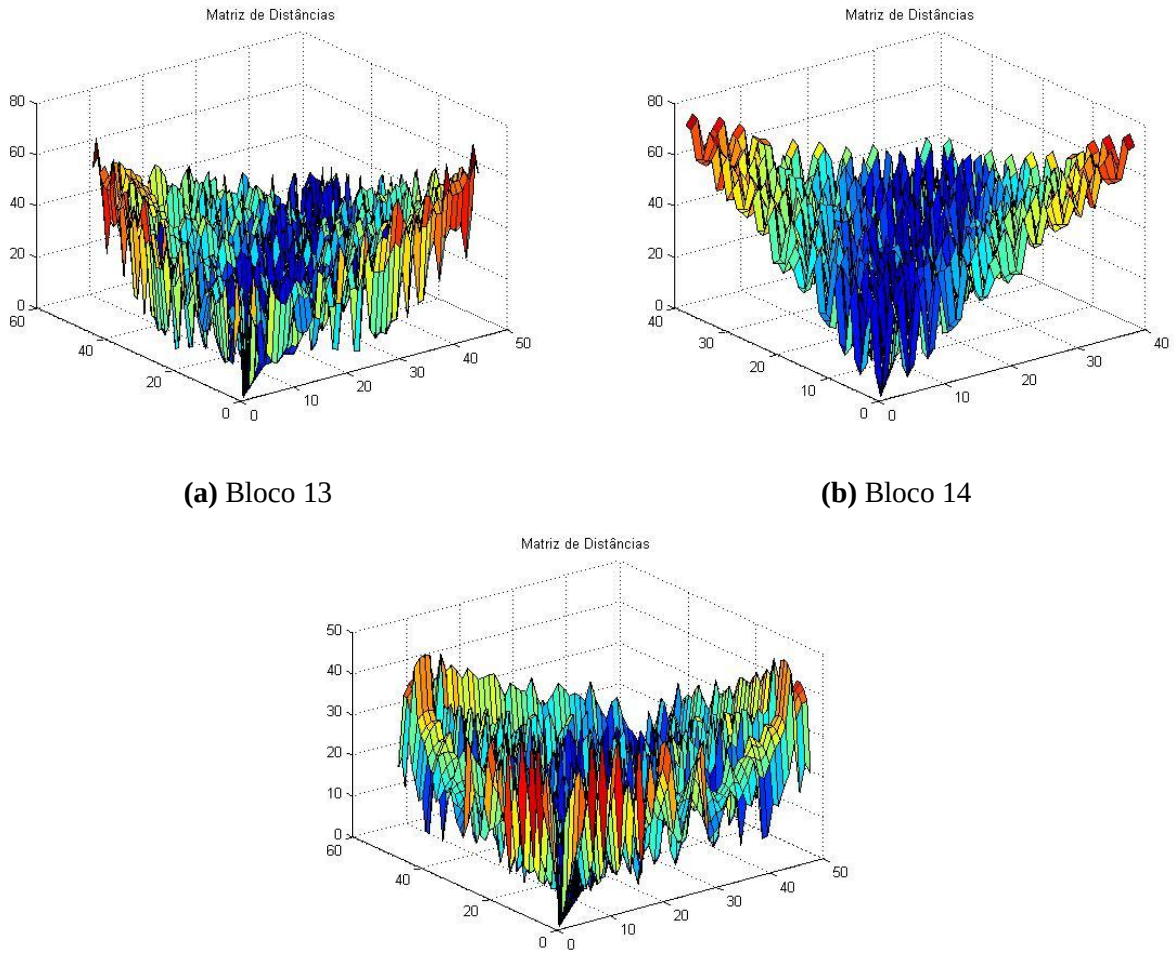
(a) Bloco 5-Sem texto **(b)** Bloco 15-Com texto

Figura 32. Proximidade entre os centróides dos contornos para os blocos 5 e 15



(a) **(b)**

Figura 33. Classificação de acordo com R3 **(a)** para os blocos classificados pelo Máx-Min e **(b)** para os restantes blocos



(a) Bloco 13

(b) Bloco 14

(c) Bloco 16

Figura 34. Proximidade entre os centróides dos contornos para os blocos 13, 14 e 16

5.5. Classificação final

O último objectivo consiste em classificar a frame enquanto frame de texto ou frame sem texto. Para tal é estipulado que uma frame é considerada de texto se tiver pelo menos um bloco, entre os 16, que seja um bloco de texto. Os resultados de R1, R2 e R3 demonstram que a classificação para cada bloco não é a mesma em todas as regras, e que uma só regra não permite uma classificação fidedigna dos blocos. Assim, um bloco só é classificado como de texto se for simultaneamente classificado pelas três regras, equação (36). Desta forma esta classificação prévia aos algoritmos das secções anteriores, diminui a possibilidade de falsos positivos para frames sem texto. É difícil que um bloco sem texto seja incorrectamente classificado pelas três regras, ou seja, a possibilidade de falsos positivos é reduzida e o desempenho dos algoritmos melhorado.

$$\text{Bloco de texto} = \begin{cases} 1, & \text{if } ((\text{R1} = 1)\&(\text{R2} = 1)\&(R3 = 1)) \\ 0, & \text{else} \end{cases} \quad \textbf{(36)}$$

O gráfico da Figura 35 demonstra o resultado do pré-processamento para a frame da Figura 20(a). Verifica-se que os blocos 13 e 15 são correctamente classificados pelas três regras e portanto a frame é classificada como uma frame de texto.

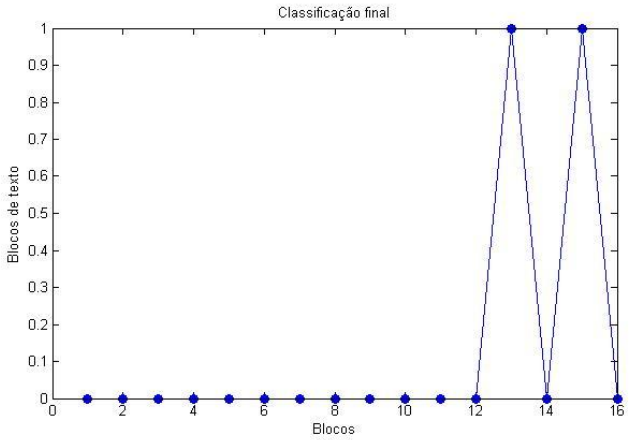
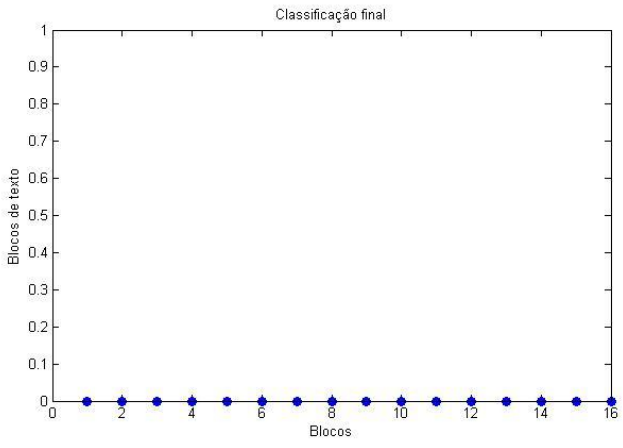


Figura 35. Classificação final da frame.

Na figura seguinte, Figura 36(b), pode visualizar-se o resultado do pré-processamento para uma frame sem texto, para demonstrar a sua efectividade. Denota-se que todos os blocos são classificados como blocos sem texto, o que significa que não foram classificados positivamente pelas três regras em simultâneo.



(a) **(b)**

Figura 36. (a) Frame sem texto e **(b)** Classificação final da frame (a).

6. Resultados e Conclusões

Para fins experimentais foi criada uma base de dados própria, uma vez que não foi encontrada nenhuma disponível na literatura. Nesta base de dados foram incluídas uma vasta variedade de frames de vídeo retiradas de filmes, notícias e eventos desportivos. Não foram avaliadas frames da mesma sequência uma vez que nas mesmas o texto aparece disposto nos mesmos sítios e com as mesmas características. O texto que aparece nestas imagens inclui múltiplas linguagens tal como o Português, o Inglês, o Coreano e o Chinês. Nestas frames há tanto texto gráfico como texto de cena. O texto gráfico consiste no texto que é adicionado, automaticamente ou sinteticamente às frames, 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. O texto de cena consiste no 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. O método foi implementado no software MATLAB num PC com Pentium Intel Core i7 2,00Ghz de processador.

Foram computados os resultados para os três algoritmos propostos e também para o Pré-Processamento

6.1 Resultados do Pré-Processamento

Para o cálculo dos resultados do Pré-Processamento foram avaliadas 100 frames de texto e 100 frames sem texto. Na Tabela 1 são apresentados os resultados para as frames avaliadas apenas pelo Método Máx-Min, ou seja, uma frame de texto é considerada correctamente classificada se este método encontrar pelo menos um bloco que seja de texto. Pelo contrário uma frame sem texto é considerada correctamente classificada se nenhum bloco for identificado como bloco de texto. Na Tabela 2 podem visualizar-se os resultados para as frames classificadas por R1, R2 e R3, ou seja, uma frame só é considerada como uma frame de texto se pelo menos um dos 16 blocos for classificado como de texto simultaneamente pelas três regras. Em oposição uma frame sem texto é correctamente classificada se as regras não identificarem em comum nenhum dos blocos como um bloco de texto. Por fim, a Tabela 3 contém os resultados de todo o Pré-Processamento ou seja o Máx-Min e a avaliação dos blocos resultantes do Máx-Min pelas três regras.

Tabela 1. Resultados da classificação pelo Máx-Min

Frames	Número	Identificadas	Não identificadas	Rendimento(%)	Tempo(seg)
Texto	100	100	0	100	40
Sem Texto	100	0	100	0	34

Analisando agora os resultados na Tabela 1 denota-se que as 100 frames de texto são todas identificadas e que nenhuma das frames sem texto é identificada. Este resultado era esperado uma vez que este método se baseia em classificar os blocos de acordo com a média dos seus contornos rectos, conforme descrito na secção 5.1. Desta forma haverá sempre blocos em que o seu número de contornos supera a média e portanto sempre blocos de texto.

Tabela 2. Resultados da classificação por R1,R2 e R3

Frames	Número	Identificadas	Não identificadas	Rendimento(%)	Tempo(seg)
Texto	100	92	8	92	197
Sem Texto	100	98	2	98	163

Na tabela 2 denota-se um elevado rendimento no processo de classificação tanto das frames de texto como das frames sem texto. Estes resultados também eram os esperados na medida em que a condição de classificação da equação (36) apenas permite a classificação se os resultados para as três regras forem consistentes entre si para cada bloco, o que significa que mesmo que uma regra classifique erradamente um bloco basta que alguma das outras duas o classifique correctamente para que no final a classificação seja a devida.

Tabela 3. Resultados da classificação integrada (Máx-Min e R1,R2 e R3)

Frames	Número	Identificadas	Não identificadas	Rendimento(%)	Tempo(seg)
Texto	100	88	12	92	125
Sem Texto	100	99	1	99	110

Na tabela 3 os resultados do rendimento para ambos os tipos de frames são bastante satisfatórios e similares aos obtidos apenas pelas três regras. No entanto, a classificação

integrada permite avaliar as frames em menos tempo do que a classificação apenas pelas regras. Uma vez que o Máx-Min classifica inicialmente os potenciais blocos de texto e que apenas estes são posteriormente avaliados por R1, R2 e R3 o tempo de computação é menor, conforme se pode verificar comparando os tempos nas Tabelas 2 e 3.

6.2 Resultados dos algoritmos propostos

As métricas de avaliação utilizadas foram as seguintes: Um bloco é considerado verdadeiramente detectado, TD (*Truly detected*), se a caixa resultante da segmentação incluir todos os caracteres ou praticamente todos. Se a caixa apresentar alguns caracteres excluídos, mais de cerca de 20%, é contabilizada como uma detecção incompleta, MD (*Miss detection*). Se a caixa incluir outros elementos que não texto então é considerada uma falsa detecção, FD (*False detection*). A determinação destas métricas é efectuada por inspecção visual de cada bloco detectado para cada uma das 100 frames de texto. Os resultados para os três algoritmos implementados encontram-se na Tabela 4. Em seguida foram definidas três métricas para avaliar o desempenho dos algoritmos. Assim, DR (*Detection Rate*) é definida como a taxa de TD/BD, em que BD (*Block Detected*) consiste no número total de blocos detectados e que foi também contabilizado por inspecção visual. MDR (*Miss Detection Rate*) consiste na taxa de MD/BD e FDR (*False detection Rate*) na percentagem de FD/BD. Os resultados para estas percentagens podem ser visualizados na Tabela 5. Os algoritmos avaliados demoram em média 1,3 minutos por frame.

Tabela 4. Resultados para as 100 frames de texto para os algoritmos

Algoritmos	**BD**	**TD**	**MD**	**FD**
HWT	580	559	12	9
FT	517	493	13	11
Sobel	528	485	23	20

Tabela 5. Desempenho para as 100 frames de texto para todos os algoritmos

Algoritmos	DR (%)	MDR (%)	FDR (%)
HWT	96,4	2,1	1,5
FT	95,3	2,5	2,2
Sobel	91,9	4,4	3,7

Os resultados na Tabela 5 para a percentagem de detecção de texto, em função do número de blocos detectados, foram bastante bons para os três algoritmos. Em anexo encontram-se os resultados para cada uma das implementações para o mesmo conjunto de frames permitindo a comparação.

No geral era esperado que os piores resultados fossem para a detecção de Sobel sobre o espaço de cores RGB pois este detector é de facto bastante sensível ao ruído. No entanto os resultados consultados na literatura, [13], não contemplavam o mesmo conjunto de características, nem o cálculo automático do treshold e nem o mesmo método de eliminação de falsos positivos. A modificação do algoritmo claramente melhorou o desempenho do mesmo. Resultados relativos a este algoritmo podem ser consultados no anexo III.

No caso do Fourier sobre o espaço RGB a consulta literária, [17], sugeria bons resultados o que praticamente se verificou para a maioria das imagens no que diz respeito à detecção de texto. No entanto no processo de segmentação as caixas obtidas quase nunca permitiam isolar cada linha de texto da que estivesse imediatamente abaixo ou acima o que efectivamente também era de esperar uma vez que a FT é bastante global para dificilmente se adaptar a singularidades locais. Por este motivo entre as linhas de texto ficava sempre algum ruído impedindo a sua separação. Pelo mesmo motivo também muitas vezes as caixas que isolavam o texto não ficavam exactamente ajustadas ao texto correndo o risco de serem consideradas falsos positivos com a metodologia utilizada o que foi efectivamente tido em consideração na definição do factor de expansão e contracção. Resultados relativos a este algoritmo podem ser visualizados no anexo II.

A detecção de Haar foi a que melhores resultados teve superando a FT por permitir uma análise local, adaptando-se desta forma melhor a singularidades. Com esta transformada não só a detecção foi bem sucedida como as caixas ao redor dos blocos de texto ficaram ajustadas permitindo o melhor resultado final, conforme se pode verificar no anexo I.

7. Limitações dos algoritmos propostos

Quando foi projectada a segmentação com base nas projecções horizontais e verticais e no algoritmo de corte recursivo foi assumido que o texto nas frames estaria sempre disposto em formato horizontal ou vertical. No entanto, em vários casos, em especial para o texto de cena, o texto aparece disposto em múltiplas direcções. Um desses casos pode ser visualizado na figura seguinte, Figura 37. Os algoritmos propostos procedem à detecção do texto independentemente da direcção em que este aparece, o que significa que o que poderia melhorar as abordagens efectuadas seria a possibilidade de uma segmentação que permitisse linhas de texto multi-orientadas. Outro ponto que poderia melhorar seria fazer uso da informação temporal para detectar os falsos positivos com maior precisão.

Figura 37. Exemplo de frame com texto multi-orientado

Referências Bibliográficas

[1] J. Zang and R. Kasturi. "Extraction of Text Objects in Video Documents: Recent Progress". *DAS*, 2008, pp 5-17.

[2] K. Jung, K.I. Kim and A.K. Jain. "Text information extraction in images and video: a survey". *Pattern Recognition*, 37, 2004, pp. 977-997.

[3] H. Li, D. Doermann and O. Kia. "Automatic Text Detection and Tracking in Digital Video". *IEEE Transactions on Image Processing*, Vol. 9, No. 1, January 2000, pp 147-156.

[4] Q. Ye, Q. Huang, W. Gao and D. Zhao. "Fast and robust text detection in images and video frames". *Image and Vision Computing* 23, 2005, pp. 565-576.

[5] A.K. Jain and B. Yu. "Automatic Text Location in Images and Video Frames". *Pattern Recognition*, Vol. 31(12), 1998, pp. 2055-2076.

[6] Y. Zhong, H. Zhang and A.K. Jain. "Automatic Caption Localization in Compressed Video". *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 4, 2000, pp. 385-392.

[7] K. L Kim, K. Jung and J. H. Kim. "Texture-Based Approach for Text Detection in Images using Support Vector Machines and Continuous Adaptive Mean Shift Algorithm". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 12, December 2003, pp 1631-1639.

[8] W. Mao, F. Chung, K. K. M. Lam and W. Siu, "Hybrid Chinese/English Text Detection in Images and Video Frames", *ICPR* 2002, pp 1015-1018..

[9] D. Chen, J. M. Odobez and J. P. Thiran, "A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning", *Signal Processing: Image Communication* 19, 2004, pp 205-217.

[10] C. Liu, C. Wang and R. Dai. "Text Detection in Images Based on Unsupervised Classification of Edge-based Features". *ICDAR 2005*, pp. 610-614.

[11] M. Anthimopoulos, B. Gatos and I Pratikakis. "A Hybrid System for Text Detection in Video Frames", *DAS*, 2008, pp 286-293.

[12] M. R. Lyu, J. Song and M. Cai, "A Comprehensive Method for Multilingual Video Text Detection, Localization, and Extraction", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 15, No. 2, February 2005, pp 243-255.

[13] P. Shivakumara, W. Huang and C. L. Tan. "An Efficient Edge based Technique for Text Detection in Video Frames". *DAS*, 2008, pp 307-314.

[14] V. Y. Marinano and R. Kasturi, "Locating Uniform-Colored Text in Video Frames", *ICPR*, 2000, pp 539-542.

[15] M. Cai, J. Song and M. R. Lyu, "A New Approach for Video Text Detection", *ICIP*, 2002, pp 117-120.

[16] S. P. Chowdhury, S. Dhar, A. K Das, B. Chanda and K. McMenemy, "Robust Extraction of Text from Camera Images", *ICDAR*, pp 2009, 1280-1284.

[17] K. Jung, "Neural network-based text location in color images", *Pattern Recognition Letters* 22, 2001, pp 1503-1515.

[18] W. Mao, F. Chung, K. K. M. Lam and W. Siu, "Hybrid Chinese/English Text Detection in Images and Video Frames", *ICPR* 2002.

[19] "Impact of Normalization Distributed K-Means Clustering", International Journal of Soft Computing, Vol.4, Issue 4, 2009, pp 168-172.

[20] R.M.Haralick, S.R.Sternberg and X. Zhuang, "Image Analysis Using Mathematical Morphology", IEEE Trans. Patter Analysis and Machine Intelligence, Vol.9, Issue 4, Julho 1987, pp 532-550

[21] Ha Jaekyu, R.M.Haralick, "Document Page Decomposition by the Bounding-Box Projection Technique ", IEEE, 1995.

[22] J. Zhang, D. Goldgof, and R. Kasturi, "A New Edge-Based Text Verification Approach for Video", IEEE International Conference on Pattern Recognition (ICPR) Florida, USA, 2008.

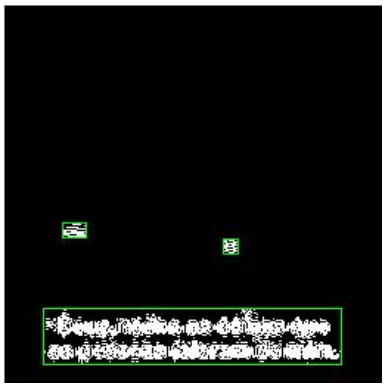
[23] Monteiro E Silva, A. B., Portugal, M. S., Cechin, A. L. (2001), "Redes Neurais Artificiais e Análise de Sensibilidade: Uma Aplicação à Demanda de Importações Brasileira", Revecap vol. 5 n. 4.

ANEXOS

ANEXO I

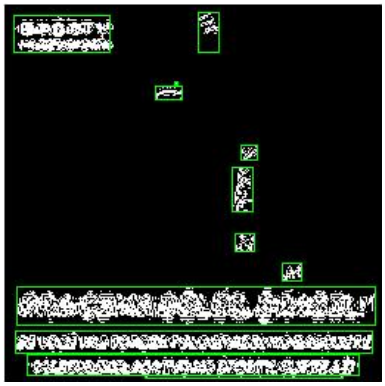
Figura A-1. Resultados do Algoritmo de Detecção pela Transformada Wavelet de Haar

ANEXO II



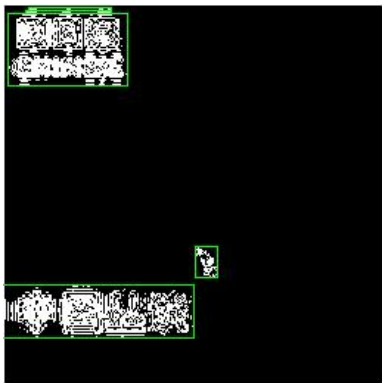

Deus, todas as coisas que
as pessoas falaram de mim.


Deus, todas as coisas que
as pessoas falaram de mim.



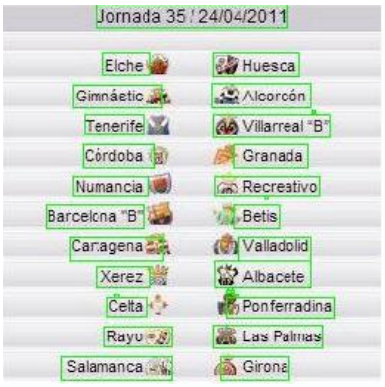

SPORTV
FB.COM/SPORTV
PRA CIMA DELES, BRASIL!
SE VOCÊ VAI TORCER PELO BRASIL NO BASQUETE
MASCULINO, AO VIVO NO SPORTV, CURTE AÍ!


SPORTV
FB.COM/SPORTV
PRA CIMA DELES, BRASIL!
SE VOCÊ VAI TORCER PELO BRASIL NO BASQUETE
MASCULINO, AO VIVO NO SPORTV, CURTE AÍ!




BBC
CHINESE
中国丛谈


BBC
CHINESE
中国丛谈


Jornada 35 ! 24/04/2011
Elche
Huesca
Gimnáetic
Alcorcón
Tenerife
Villarreal "B"
Córdoba
Granada
Numancia
Recreativo
Barcelona "B"
Betis
Cartagena
Valladolid
Xerez
Albacete
Celta
Ponferradina
Rayo
Las Palmas
Salamanca
Girona

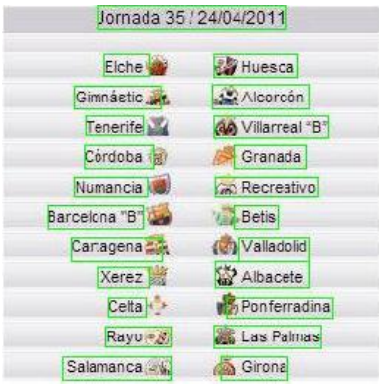

Jornada 35 ! 24/04/2011
Elche
Huesca
Gimnáetic
Alcorcón
Tenerife
Villarreal "B"
Córdoba
Granada
Numancia
Recreativo
Barcelona "B"
Betis
Cartagena
Valladolid
Xerez
Albacete
Celta
Ponferradina
Rayo
Las Palmas
Salamanca
Girona

Figura A-2. Resultados do Algoritmo de Detecção pela Transformada de Fourier sobre o espaço de cores RGB

ANEXO III


Deus, todas as coisas que as pessoas falaram de mim.
Deus, todas as coisas que as pessoas falaram de mim.
Deus, todas as coisas que as pessoas falaram de mim.
PRA CIMA DELES, BRASIL!
SPORTV
FB.COM/SPORTV
PRA CIMA DELES, BRASIL!
SE VOCÊ VAI TORCER PELO BRASIL NO BASQUETE MASCULINO, AO VIVO NO SPORTV. CURTE AÍ!
SPORTV
FB.COM/SPORTV
PRA CIMA DELES, BRASIL!
SE VOCÊ VAI TORCER PELO BRASIL NO BASQUETE MASCULINO, AO VIVO NO SPORTV. CURTE AÍ!
BBC CHINESE
中国从谈
BBC CHINESE
中国从谈
BBC CHINESE
中国从谈
Jornada 35 ! 24/04/2011
Elche Huesca
Gimnàetic Alcorcón
Tenerife Villarreal "B"
Córdoba Granada
Numancia Recreativo
Barcelona "B" Betis
Cartagena Valladolid
Xerez Albacete
Celta Ponferradina
Rayo Las Palmas
Salamanca Girona
Jornada 35 ! 24/04/2011
Elche Huesca
Gimnàetic Alcorcón
Tenerife Villarreal "B"
Córdoba Granada
Numancia Recreativo
Barcelona "B" Betis
Cartagena Valladolid
Xerez Albacete
Celta Ponferradina
Rayo Las Palmas
Salamanca Girona

Figura A-3. Resultados do Algoritmo de Detecção de texto baseado no comportamento dos contornos de Sobel sobre o espaço de cores RGB