MÉTODO K-MEANS RECORRENTE PARA EXTRAÇÃO DE TEXTO EM IMAGENS WEB

André P. N. Tahim, Rui Seara

LINSE - Laboratório de Circuitos e Processamento de Sinais
Depto. Engenharia Elétrica, Universidade Federal de Santa Catarina
88040-900, Florianópolis, Santa Catarina, Brasil

Emails: tahim@linse.ufsc.br, seara@linse.ufsc.br

Abstract— Content-based indexing and retrieval of images are emerging applications due to the increasing diffusion of image data on the internet and corporate intranets. Embedded texts within images have very useful semantic contents for keyword-based search and text-based indexing. In general, text extraction from images involves three stages: localization, extraction itself, and transformation for plain text through optical character recognition (OCR) systems. After locating text regions, a character extractor must provide a binary output to an OCR system, which has as goal to recognize the extracted characters. The extraction task is performed by an ad hoc algorithm, whose purpose is to segment the located region into two planes: text and background. The present research work aims to provide a new approach for the character extraction stage. The proposed algorithm clusters image colors by using a modification of the k-means method, which determines automatically the starting seeds, identifying (as incorrect) segmented regions including artifacts coming from the compression process. After identifying such regions, the corresponding seeds are discarded and an iterative version of the k-means method is applied to surviving seeds. Experimental results show that the proposed approach is robust for text character extraction in a context with reduced size fonts, low character density, complex background and non-uniform illumination.

Keywords— Color clustering, k-means, text extraction.

Resumo— Indexação e recuperação de imagens baseadas em seu conteúdo são aplicações emergentes originadas pela rápida proliferação de dados em formato de imagens na *internet* e em *intranets* corporativas. Textos embutidos em imagens possuem conteúdo semântico de grande utilidade para pesquisa considerando palavras-chave e indexação baseada em texto. Em geral, a extração textual em imagens envolve três etapas: localização, extração propriamente dita e transformação em texto plano através de sistemas OCR (*optical character recognition*). Após a localização das regiões textuais, um extrator de caracteres deve fornecer uma entrada binária para um sistema OCR, o qual tem como objetivo reconhecer os caracteres extraídos. A tarefa de extração é realizada por um algoritmo cujo propósito é segmentar a região localizada em dois planos: texto e plano de fundo. O presente trabalho visa contribuir com um novo método para a etapa de extração de caracteres. O algoritmo proposto clusteriza as cores da imagem utilizando uma modificação do método *k-means*, o qual determina automaticamente as sementes iniciais, identificando (quando incorretas) as regiões segmentadas que incluem artefatos provenientes do processo de compressão. Após a identificação de tais regiões, as correspondentes sementes são descartadas e uma versão iterativa do método *k-means* é aplicada às sementes sobreviventes. Resultados experimentais mostram que o algoritmo proposto é robusto na extração de regiões textuais contendo fontes de tamanho reduzido, caracteres de baixa densidade, plano de fundo complexo e iluminação não-uniforme.

Palavras-chave— Segmentação de cor, *k-means*, extração de texto.

1 Introdução

Atualmente, as informações veiculadas nos diferentes meios de comunicação se apresentam em variados formatos. Até recentemente, a difusão de tais informações era basicamente textual; porém, com o advento da *internet* e a crescente capacidade de transmissão de dados, disseminou-se uma grande quantidade de informações em forma de imagem e vídeo, culminando em problemas de indexação e recuperação de todo esse conteúdo.

Diversos estudos sobre a extração do conteúdo semântico em imagens vêm sendo utilizados sob a forma, por exemplo, de face, veículos e ação humana. Contudo, técnicas que avaliam o conteúdo semântico através de textos superpostos em imagens vêm suscitando o interesse de muitos pesquisadores pelos seguintes motivos: (*i*) textos possuem informações semânticas úteis para descrever o conteúdo de uma imagem; (*ii*) a extração é relativamente mais simples do que outras características semânticas; (*iii*) habilita aplicações, tais como pesquisa de imagens baseada em palavras-chave e indexação de imagens baseada em texto.

Os projetistas de sítios Web freqüentemente criam textos na forma de imagens (cabeçalhos, *banners*, títulos) para suprir as limitações estilísticas do HTML. Apesar dessas imagens possuírem um alto valor semântico, elas não são indexadas pelas atuais máquinas de busca da Web devido à ausência de uma tecnologia que permita a extração e o reconhecimento de texto em imagens (*Search Engine Watch*, 2006). Logo, todo o texto em forma de imagens na Web é ignorado.

Segundo estudo realizado por Antonacopoulos et al. (2001), 17% de toda informação textual contida na internet está sob a forma de imagem. Um dado ainda mais significante é que 76% dessas palavras, em forma de imagem, não aparecem codificadas em texto (ASCII ou UNICODE) em nenhum outro lugar. Assim, pode-se notar a necessidade de tornar as máquinas de busca aptas a

lidar com indexação de texto presente em imagens. No entanto, imagens da Web trazem uma nova série de desafios: (*i*) apresentam diversos artefatos devido à quantização de cor e perdas por compressão (Vergara Villegas et al., 2006); (*ii*) são de baixa resolução (72 dpi); (*iii*) as fontes presentes em *banners* e cabeçalhos são de tamanho bastante reduzido (por exemplo, 5 pt-7 pt); (*iv*) imagens coloridas podem apresentar dezenas de milhares de cores, aumentando a complexidade computacional dos algoritmos; (*v*) os sistemas OCR (*optical character recognition*) tradicionais não estão preparados para tratar imagens coloridas, com fontes reduzidas e resolução inferior a 300 dpi (Karatzas and Antonacopoulos, 2004).

Os sistemas de extração e reconhecimento de texto em imagens geralmente envolvem três etapas: localização, extração propriamente dita e OCR (Jung et al., 2004). Este artigo visa contribuir com a etapa de extração, cuja função é transformar as regiões previamente selecionadas como textuais em uma imagem binária (duas regiões): texto e plano de fundo. Essa transformação é necessária para adequar a região textual às necessidades dos OCRs tradicionais.

Existem duas abordagens principais para a etapa de extração utilizando clusterização de cor: *feature-space* e *image-domain* (Lucchese and Mitra, 2001). A abordagem *feature-space* utiliza como critério de similaridade unicamente as características de cor dos pixels, geralmente resultando em regiões fragmentadas. A sua grande vantagem é a baixa complexidade computacional. A abordagem *image-domain* gera regiões coesas por incluir, no critério de similaridade, a relação espacial entre os pixels; porém, a melhoria na segmentação é obtida às custas de uma alta complexidade computacional.

1.1 Trabalhos Relacionados

Para que o reconhecimento de texto seja bem sucedido, ele deve estar separado do plano de fundo em uma imagem binária. Algumas abordagens utilizam o *bit dropping*, técnica que coleta apenas os n bits mais significativos de cada plano (RGB) da imagem para reduzir a complexidade computacional no processamento de imagens coloridas. Jain and Yu (1998) utilizam o *bit dropping* (2 bits mais significativos de cada plano RGB) gerando uma imagem com no máximo 64 cores [Figura 1] e adotam como premissa a homogeneidade de cor no corpo dos caracteres; aplicam a clusterização de cor (*single-link*), separam cada região em uma imagem binária e identificam a região textual através de projeções de perfil. Observando a Figura 1(a), nota-se que a suposição de homogeneidade de cor no corpo dos caracteres é inapropriada por possuir uma grande quantidade de cores e artefatos incluídos durante o processo de compressão. Além disso, o *bit dropping* transforma dezenas de milhares de cores em apenas 64 cores igualmente espaçadas no espectro, incluindo cores inexistentes e perceptualmente muito diferentes no corpo dos caracteres [Figura 1(b)]. Como consequência, após a clusterização de cor, os caracteres de baixa densidade são fragmentados em várias regiões.

(a)

(b)

Figura 1: (a) Imagem original com 1679 cores. (b) Imagem após o *bit dropping* com 10 cores.

Song et al. (2005) utilizam o método *k-means* modificado para determinar o número de sementes automaticamente; porém, não discutem qualquer técnica de avaliação da clusterização final, visando equacionar e solucionar o problema dos artefatos de compressão nas bordas dos caracteres, que resultam na fragmentação destes em várias regiões.

O algoritmo de segmentação textual proposto possui a vantagem de utilizar todas as cores originais da imagem no processo de clusterização de cor. Tal processo emprega o método *k-means* e possui como inovações a determinação automática das sementes iniciais mediante uma avaliação do histograma de cor e a utilização da métrica *city-block* para determinar a similaridade entre cores, sendo esta última de menor complexidade computacional e mais próxima da percepção visual humana do que a métrica Euclidiana (Loo and Tan, 2004). Além disso, o maior avanço foi associar ao algoritmo proposto um método iterativo denominado aqui *k-means recorrente*, cuja função é aproveitar a baixa complexidade computacional da abordagem *feature-space* e verificar espacialmente a qualidade da segmentação apenas nas regiões críticas: as bordas dos caracteres. O método *k-means recorrente* é capaz de eliminar regiões segmentadas incorretamente nas bordas dos caracteres devido aos artefatos incluídos durante o processo de compressão, evitando assim a fragmentação dos caracteres em vários componentes conectados. O algoritmo gera regiões mais compactas (principal carência dos métodos *feature-space*) e mostra-se robusto na extração de texto em imagens contendo caracteres com fonte de tamanho reduzido, baixa densidade, plano de fundo complexo e com iluminação não-uniforme.

O presente trabalho é organizado como segue. A Seção 2 descreve o algoritmo proposto em detalhe. A Seção 3 apresenta os resultados experi-

mentais e a Seção 4 é destinada as conclusões e comentários finais do trabalho.

2 Algoritmo de Extração de Texto

O método aqui proposto para a etapa de extração de texto em imagens visa alcançar três objetivos:

- Baixa complexidade computacional.
- Geração do menor número de regiões possíveis durante a segmentação[1].
- Geração de regiões com coesão espacial, evitando que caracteres sejam fragmentados.

2.1 Espaço de Cor e Métrica de Similaridade

Para manter a complexidade computacional baixa, optou-se por trabalhar com o espaço de cor RGB. Embora ele não seja adequado à percepção visual humana, o conjunto de dados nesse espaço não exige qualquer transformação para inicializar o processamento.

Para reduzir a influência da não conformidade do espaço de cor utilizado com respeito à percepção visual humana, buscou-se uma métrica de similaridade em que cores distantes no espaço RGB sejam significativamente diferentes para o observador humano. Fundamentado em um experimento realizado por Loo and Tan (2004), escolheu-se a métrica *city-block* (Gonzalez and Woods, 2002). Tal experimento demonstrou que a referida métrica está mais próxima do sistema visual humano do que a métrica Euclidiana, com a vantagem de se obter uma importante redução de complexidade computacional.

2.2 Região de Borda

Aqui é discutido o conceito de região de borda como também é fundamentada a necessidade da eliminação desse tipo de região.

Define-se região de borda como toda região gerada durante a clusterização de cor que possui mais de 25% dos seus pixels sobre as bordas da imagem. Tais bordas são representadas por qualquer imagem binária obtida a partir de alguma técnica de detecção de borda, por exemplo, *Sobel*, *Canny* (Gonzalez and Woods, 2002).

Regiões de borda não são desejadas por três razões principais: (*i*) geralmente são regiões com muitas fragmentações; (*ii*) são pixels que sofreram variação de cor durante o processo de compressão, adquirindo cores muito distintas dos pixels de sua vizinhança, não representando qualquer objeto na imagem; (*iii*) possuem pixels que deveriam estar agregados à região do corpo do caractere, porém foram alocados em uma região diferente.

[1]Considera-se que duas regiões seja o número ótimo, na qual uma região corresponde ao texto e a outra, ao plano de fundo.

2.3 Método K-means Recorrente

As técnicas de clusterização de cor baseadas na abordagem *feature-space* possuem baixa complexidade computacional, porém geralmente fragmentam objetos caracterizados basicamente por bordas e de baixa densidade (características comuns em caracteres). O método proposto utiliza a abordagem *feature-space*, produzindo regiões compactas por associar uma malha de realimentação capaz de identificar e eliminar regiões segmentadas incorretamente nas bordas dos caracteres, como ilustrado no diagrama de blocos da Figura 2. Tal técnica foi denominada *k-means recorrente*, devido à recorrência feita sempre que ocorre a identificação e eliminação das sementes geradoras das regiões de borda. No presente trabalho, extraiu-se o texto da *imagem-exemplo* [Figura 1(a)] para descrever detalhadamente o algoritmo proposto.

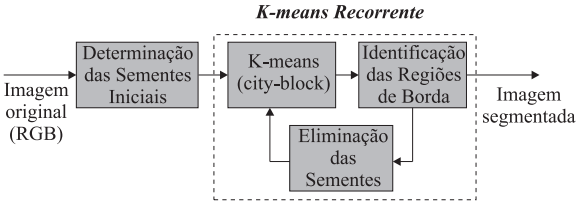


Figura 2: Diagrama de blocos do método de extração de caracteres *k-means recorrente*.

O método *k-means* tradicional não determina automaticamente nem o número nem quais as sementes iniciais que devem ser usadas para o processo de clusterização de cor. Para automatizar tal processo, o método proposto determina quais são as sementes iniciais através da avaliação do histograma de cores da imagem a partir dos seguintes passos (1º bloco, Figura 2):

1. Cálculo do histograma de cores da imagem.

2. Seleção das 5 cores de maior ocorrência no histograma como candidatas a sementes. Resultados experimentais mostraram que tal escolha produz resultados similares à avaliação de todas as cores como candidatas.

3. Dentre as cores candidatas, escolhe-se como semente aquela que possui o maior número de pixels com cores a uma distância *city-block* menor do que um limiar $\mathcal{T}$, em que $0 \leq \mathcal{T} \leq 765$ (ou seja, 3×255). Utilizou-se neste trabalho $\mathcal{T} = 120$, por tal valor estar em conformidade com o experimento de percepção realizado por Loo e Tan (2004).

4. Atribui-se à semente todas as cores que estão a uma distância (*city-block*) menor do que o limiar $\mathcal{T}$.

5. O processo se reinicializa do passo 2 com todas as cores não atribuídas a uma semente. A determinação do conjunto de sementes iniciais é concluída quando todas as cores da imagem estão atribuídas a uma semente.

Após a determinação automática das sementes, o método *k-means* é inicializado usando a métrica *city-block* para medir as distâncias entre as amostras e o centróide de cada *cluster* (bloco *k-means*, Figura 2). Ao final da clusterização, as cores da imagem são segmentadas em um número de *clusters* igual ao número de sementes iniciais. Para a *imagem-exemplo* [Figura 1(a)], são geradas 3 sementes, segmentando as cores da imagem em 3 *clusters*, como ilustrado na Figura 3. Cada *cluster* corresponde a uma região da imagem original, representada por uma imagem binária. Tais imagens (binárias) são obtidas através da substituição (na imagem original) dos pixels com cores pertencentes a um dado *cluster* pelo nível lógico '1' (branco), enquanto os pixels restantes, pelo nível lógico '0' (preto), gerando-se, assim, uma imagem binária para cada cluster, como ilustrado na Figura 4.

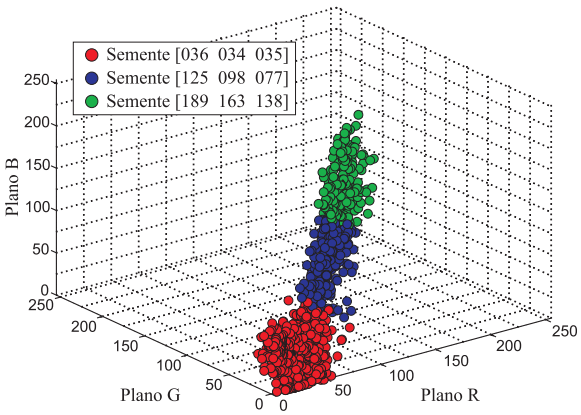


Figura 3: Clusterização de cores da *imagem-exemplo* através das 3 sementes iniciais.

Figura 4: (a) Imagem original. Região gerada pelo *cluster* referente à: (b) semente [036 034 035]; (c) semente [125 098 077]; (d) semente [189 163 138].

Observando as regiões geradas pelo método *k-means*, percebe-se que o *cluster* gerado pela semente [125 098 077] (representando os valores dos planos [R G B]), como ilustrado na Figura 4(c), é indesejável por originar uma imagem binária bastante fragmentada, não representando qualquer objeto da imagem e constituída preponderantemente das bordas dos caracteres. Os pixels das bordas da imagem possuem cores perceptualmente muito diferentes dos pixels da sua vizinhança, tornando comum agregar tais pixels, durante o processo de clusterização de cor, em uma região composta predominantemente pelas bordas da imagem [Figura 4(c)].

A geração automática de sementes pode eleger, erroneamente, uma semente indesejada, resultando em uma região de borda. Para corrigir esse tipo de erro na determinação do número de sementes, propôs-se o método *k-means recorrente*. Tal método verifica, ao final de cada clusterização, a existência de regiões de borda entre as regiões geradas (bloco de identificação, Figura 2). Caso alguma região seja caracterizada como região de borda, a semente geradora dessa região é descartada (bloco de eliminação, Figura 2) e o método *k-means* é reinicializado com as sementes sobreviventes até que não existam mais regiões de borda.

A etapa de identificação das regiões de borda (bloco de identificação, Figura 2) é feita por meio da operação binária '*AND*' entre as regiões referentes a cada *cluster* [Figuras 5(c), 5(e) e 5(g)] e uma *imagem-borda* [Figura 5(b)]. Esta última é obtida por qualquer método de detecção de bordas sobre a imagem original. No presente trabalho, utilizou-se o método *Sobel* (Gonzalez and Woods, 2002) para determinar a *imagem-borda* binária, como mostrado na Figura 5(b). Como resultado, obtém-se imagens de intersecção entre as regiões clusterizadas e a *imagem-borda*, como ilustrado nas Figuras 5(d), 5(f) e 5(h). Efetua-se então a contagem dos pixels de intersecção de cada imagem. Caso o número ultrapasse em 25% o número de pixels da respectiva região clusterizada, a região é considerada de borda.

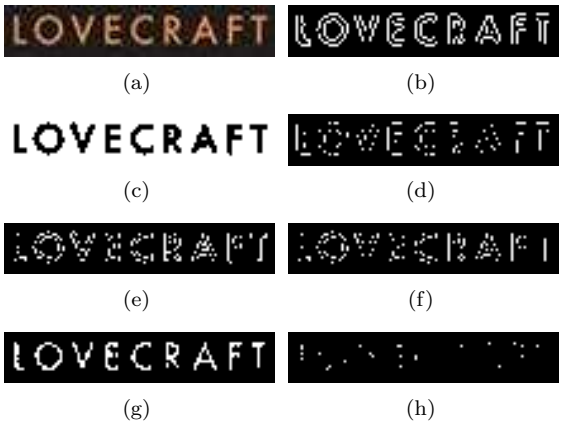


Figura 5: (a) Imagem original. (b) *Imagem-borda*. Imagens binárias representando o *cluster* referente à: (c) semente [036 034 035]; (e) semente [125 098 077]; (g) semente [189 163 138]. Imagens binárias de intersecção entre a *imagem-borda* e os *clusters* referentes à: (d) semente [036 034 035]; (f) semente [125 098 077]; (h) semente [189 163 138].

No exemplo em questão, pode-se observar que apenas a região representada pela Figura 5(e) produz uma imagem de intersecção [Figura 5(f)] possuindo mais do que 25% dos pixels da região que a originou [Figura 5(e)], caracterizando-a como região de borda. O bloco de eliminação da Figura 2

é responsável por descartar a semente geradora da região de borda e reinicializar o método *k-means* com as sementes sobreviventes.

Após cada iteração do método *k-means recorrente* usando as sementes sobreviventes, uma nova avaliação é feita sobre as imagens binárias que representam cada região até que não existam mais regiões de borda ou o número de *clusters* seja igual a 2. A Figura 6 apresenta o resultado da clusterização das cores referente a segunda iteração do método *k-means recorrente*, considerando as duas sementes sobreviventes. As imagens binárias representando cada região clusterizada são mostradas na Figura 7. Observa-se, ao final da clusterização utilizando o método *k-means recorrente* (Figura 7), que o número de regiões clusterizadas é reduzido e os pixels que antes representavam uma região de borda agora estão agregados ao corpo dos caracteres, melhorando significativamente a qualidade da segmentação.

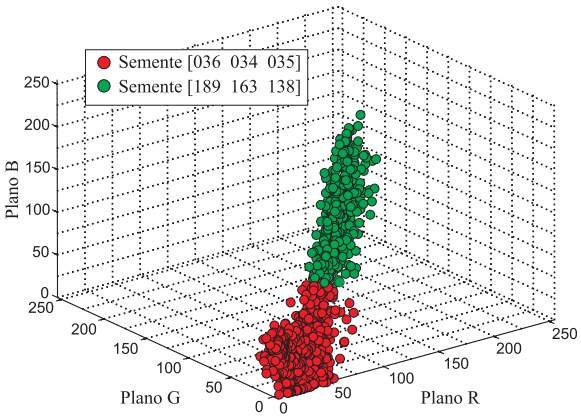


Figura 6: Clusterização das cores com as 2 sementes sobreviventes.

(a) (b)

Figura 7: Imagens binárias representando as regiões clusterizadas após a 1ª recorrência do método *k-means recorrente*. (a) Região gerada pela semente [036 034 035]. (b) Região gerada pela semente [189 163 138].

3 Resultados Experimentais

Para avaliar o desempenho do algoritmo proposto, 130 imagens contendo apenas regiões textuais foram coletadas aleatoriamente na Web. Tais imagens continham diferentes tipos de fonte, dimensões, planos de fundo e contrastes. A faixa de resolução das imagens variam entre 50×10 e 600×200 e o tamanho dos caracteres, entre 8 e 530 pixels de altura.

Muitos autores propõem, como método de avaliação de desempenho, a contagem da quantidade de caracteres reconhecidos corretamente do conjunto total de caracteres após a passagem por um sistema OCR. Tal técnica leva em consideração a qualidade do extrator em conjunto com o OCR, mascarando dessa forma o real desempenho do processo de extração. Por esse motivo, no presente trabalho, utilizou-se a avaliação visual (subjetiva) para comparar os resultados do método proposto com o método de Otsu (1979), visando avaliar a robustez referente à iluminação não-uniforme e, para julgar a qualidade dos caracteres extraídos e comparar o número de *clusters* gerados, foi utilizado o método de Jain e Yu (1998). Pode-se inferir, apesar da recorrência à clusterização *k-means*, que a complexidade computacional do método proposto é inferior aos algoritmos *image domain*, visto que a média de recorrências sobre as 130 imagens avaliadas foi de apenas 1.15 (muito baixo). Além do mais, a verificação espacial associada ocorre apenas nas regiões de borda por meio da operação lógica '*AND*'.

As Figuras 8 a 9 representam uma amostra dos resultados para diferentes tipos de fonte, planos de fundo e iluminação. É importante ressal-

(a) (b)

(c) (d)

Figura 8: Imagem com plano de fundo complexo e caracteres de baixa densidade. (a) Imagem original. (b) Método de Otsu. (c) Método de Jain & Yu (10 *clusters*). (d) Método proposto (5 *clusters*).

(a) (b)

(c) (d)

Figura 9: Imagem com iluminação não-uniforme. (a) Imagem original. (b) Método de Otsu. (c) Método de Jain & Yu (5 *clusters*). (d) Método proposto (2 *clusters*).

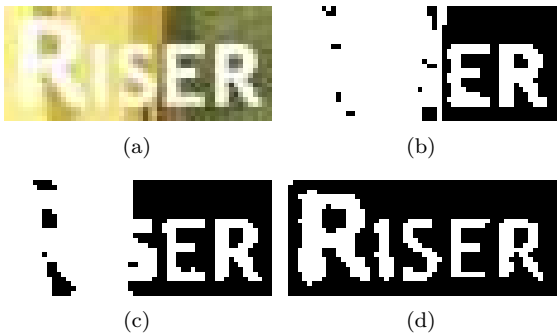
tar que apenas são ilustrados os *clusters* referentes à região textual; contudo, o número de *clusters* gerados é uma informação importante, visto que, após qualquer método de extração, deve-se utilizar uma ferramenta para identificar qual imagem binária representa a região textual (por exemplo, projeções de perfil). Um menor número de *clusters* indica menor complexidade computacional na identificação da imagem binária que representa os caracteres.

(a) (b) (c) (d)

Figura 10: Imagem com artefatos e caracteres de baixa densidade. (a) Imagem original. (b) Método de Otsu. (c) Método de Jain & Yu (6 *clusters*). (d) Método proposto (2 *clusters*).

(a) (b) (c) (d)

Figura 11: Imagem com caracteres de fontes reduzidas e baixa densidade. (a) Imagem original. (b) Método de Otsu. (c) Método de Jain & Yu (7 *clusters*). (d) Método proposto (2 *clusters*).



(a) (b) (c) (d)

Figura 12: Imagem com plano de fundo complexo. (a) Imagem original. (b) Método de Otsu. (c) Método de Jain & Yu (7 *clusters*). (d) Método proposto (4 *clusters*).

4 Conclusões

Neste trabalho, desenvolveu-se uma nova técnica de extração de texto que associa o bom desempenho das técnicas de clusterização de cor *feature-space* (utilizando todas as cores da imagem) com uma avaliação da segmentação em regiões críticas (bordas da imagem). O algoritmo proposto é capaz de identificar, iterativamente, regiões segmentadas desnecessariamente, corrigindo-as, de forma recorrente, pelo método *k-means* após a eliminação das sementes correspondentes às regiões de borda. O método mostrou-se robusto na extração de textos em imagens com artefatos, caracteres de baixa densidade, plano de fundo complexo, fontes de tamanho reduzido e iluminação não-uniforme, gerando um pequeno número de *clusters*.

Referências

Antonacopoulos, A., Karatzas, D., and Lopez, J. O. (2001). Accessing textual information embedded in internet images, *SPIE - Internet Imaging II*, San Jose, USA, pp. 198–205.

Gonzalez, R. and Woods, R. (2002). *Digital Image Processing, Second Edition*, Prentice Hall.

Jain, A. and Yu, B. (1998). Automatic text location in images and video frames, *Pattern Recognition* **31**(12): 2055–2076.

Jung, K., Kim, K. I., and Jain, A. K. (2004). Text information extraction in images and video: a survey, *Pattern Recognition* **37**(5): 977–997.

Karatzas, D. and Antonacopoulos, A. (2004). Text extraction from web images based on a split-and-merge segmentation method using color perception, *17th International Conference on Pattern Recognition (ICPR2004)*, IEEE-CS Press, Cambridge, UK, pp. 634–637.

Loo, P. K. and Tan, C. L. (2004). Adaptive region growing color segmentation for text using irregular pyramid, *Document Analysis Systems*, pp. 264–275.

Lucchese, L. and Mitra, S. (2001). Color image segmentation: a state-of-the-art survey, *Proc. of the Indian National Science Academy (INSA-A)*, New Delhi, India, pp. 207–221.

Otsu, N. (1979). A threshold selection method from gray-level histograms, *IEEE Transactions on Systems, Man and Cybernetics* **9**(1): 62–66.

Search Engine Watch (2006). *http://searchenginewatch.com

Song, Y. J., Kim, K. C., Choi, Y. W. Y. W., Byun, H. R., Kim, S. H., Chi, S. Y., Jang, D. K., and Chung, Y. K. (2005). Text region extraction and text segmentation on camera-captured document style images, *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, IEEE Computer Society, Washington, DC, USA, pp. 172–176.

Vergara Villegas, O., Elias, R., and Cruz Sanchez, V. (2006). Feature preserving image compression: A survey, *Electronics, Robotics and Automotive Mechanics Conference*, Cuernavaca, Mexico, pp. 35–40.