MARCELO SOUZA RAMOS

LOCALIZAÇÃO DE CÓDIGOS DE IDENTIFICAÇÃO EM VAGÕES DE TREM

CURITIBA

2012

MARCELO SOUZA RAMOS

LOCALIZAÇÃO DE CÓDIGOS DE IDENTIFICAÇÃO EM VAGÕES DE TREM

Projeto apresentado ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná como requisito para obtenção do título de Mestre em Informática.

Área de Concentração: *Visão Computacional, Reconhecimento de Padrões e Computação Forense*

Orientador: Prof. Dr. Alceu de Souza Britto Jr.,
Co-Orientador: Prof. Dr. Jacques Facon

CURITIBA
2012

Agradecimentos

A elaboração deste trabalho não teria sido possível sem a colaboração, estímulo e empenho de diversas pessoas. Gostaria de expressar toda a minha mais sincera gratidão e estima a todos aqueles que, direta ou indiretamente, contribuíram para que esta tarefa árdua se tornasse uma realidade. A todos quero manifestar os meus sinceros agradecimentos.

Agradeço a empresa ALL-América Latina Logística por ter possibilitado a coleta e uso dos vídeos nos experimentos.

Agradeço especialmente aos meus pais por terem apoiado em todas as minhas escolhas. A minha esposa Juliana pela compressão que foi com certeza fundamental para o sucesso desse trabalho.

Agradeço aos meus orientadores Prof. Dr. Alceu de Souza Britto Jr. e Prof. Dr. Jacques Facon pela atenção, apoio e paciência durante o desenvolvimento do projeto, mas principalmente por acreditarem em mim na execução deste trabalho.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

H	Altura.
A	Área.
EA	Área de borda.
B	Bloco de texto.
f	Função.
W	Largura.
SM	Mapa binário de borda de Sobel.
$\in$	Pertence.
HP	Perfil de projeção horizontal.
VP	Perfil de projeção vertical.
AR	Relação de aspecto.

$\sum$	Somatório.

Lista de Abreviaturas

HP	*Horizontal Profile.*
MFI	*Multiple Frame Integration.*
MGD	*Maximum Gradient Difference.*
OCR	*Optical Characters Recognition.*
VP	*Vertical Profile.*
TBG	*Text Block Group.*
LBP	*Local Binary Pattern.*
DLBP	*Dominant Local Binary Pattern.*

Resumo

Inúmeras pesquisas têm sido feitas na área de Visão Computacional com objetivo de propor uma solução para o problema de se localizar texto em vídeos. A motivação está no número potencial de aplicações. Neste contexto, o presente trabalho tem como objetivo propor um método para localização de códigos de identificação em vagões de trem a partir de vídeos. Tal método será útil no escopo de projeto maior cujo objetivo é fazer a leitura automática de códigos em vagões a partir de vídeos.

O método proposto está dividido em quatro etapas: Pré-processamento, Segmentação, Detecção do Bloco de Texto e Pós-processamento. Na etapa de Pré-processamento o vídeo é segmentado em múltiplos quadros e transformado para tons de cinza. Na etapa de Segmentação são propostas duas abordagens: na primeira abordagem o quadro (imagem) é submetido a um único processo de limiarização, já na segunda, a imagem é submetida a três técnicas de limiarização cujos resultados são combinados. Na etapa de Detecção de Blocos de Texto utiliza-se um filtro tipo passa-alta (Laplaciano) normalizando-se os resultados para o intervalo [0,1]. A detecção do texto tem como base a diferença do máximo gradiente. Por fim, na etapa de Pós-processamento é empregado um filtro baseado no fator de compacidade dos componentes conexos conectados.

Experimentos sobre uma base de vídeos contendo 2582 quadros onde aparecem 116 vagões com diferentes formatos (graneleiros, tanques e plataformas), posição, fonte e cor dos códigos, além dos mais variados problemas gerados pela falta de manutenção dos vagões, demonstram que o método é promissor. O método proposto localizou o código em 84,48% dos casos.

Palavras-Chave: Visão Computacional, Textos em Imagens

Abstract

Many efforts have been done to develop methods based on Computer Vision techniques for text location in video scenes. The motivation is the large number of applications. In this context, the subject of the present work is to develop a method for text code location in train wagons from vídeo images. The proposed method is part of a major project where the main objective is the automatic reading of text codes used to identify the train wagons.

The proposed method is divided into four steps: Pre-processing, Segmentation, Text Block Detection and Post-processing. In the Pre-processing step, the video is segmented into multiple frames and converted to grayscale. In the Segmentation step, two approaches are proposed: in the first one the frame (image) is submitted to a single thresholding process, while in the second one, three thresholding techniques are used and their results are combined. In Text Block Detection, firstly, a high-pass filter type (Laplacian) is used and the results are normalized for the the interval [0,1]. Afterwards, the text detection is done based on the difference of the maximum gradient. Finally, in the Post-processing step, a filter based on factor compactness is applied on the connected components. Theobjective is to keep just the components that have their shape similar to that of blocks of texts.

Preliminary experiments on a videos data-base containing 2582 frames in which appears 116 wagons with different formats (bulk carriers, tanks and platforms), positions, fonts and colors of text codes, and the most varied problems created by lack of maintenance of the wagons, show that the method is very promising. The method has correctly located the code in 84.48% of the images.

Keywords: Text location in images, Computer Vision

Capítulo 1

Introdução

A tecnologia digital avança rapidamente e a quantidade de informações disponíveis em multimídia (vídeos, som e etc.) continua crescendo. Com esse crescimento, existe uma necessidade emergente em navegar e resgatar informações contidas nessas mídias. Um dos recursos de multimídia mais utilizado é o vídeo. Neste contexto, a interpretação de uma cena em vídeo muitas vezes exige a localização e/ou a leitura de textos que podem ser de dois tipos: texto gráfico e texto na cena. Texto gráfico é aquele adicionado artificialmente no vídeo durante o processo de edição e o texto na cena ocorre naturalmente nas imagens capturadas pelas câmeras, como por exemplo, os textos contidos em objetos contidos na cena.

Embora muitos métodos tenham sido propostos na literatura, a detecção de textos em vídeo ainda é um problema desafiador. O grande desafio a ser enfrentado é localizar textos escritos em objetos contidos em cenas de vídeo. Existem várias técnicas que estudam mecanismos para localizar textos em vídeo, no entanto esse estudo tem se mostrado extremamente complexo pelo fato de existir um razoável número de variáveis que devem ser levadas em consideração no processo. Os problemas mais comuns encontrados são: os vídeos freqüentemente têm uma baixa resolução; o fundo da imagem complexo; textos com diferentes tamanhos, estilos e alinhamentos diferenciados dentre outros. No caso de texto na cena, existe mais um problema relacionado às condições de iluminação que nem sempre são as ideais e também às distorções de perspectivas que dificultam ainda mais o trabalho.

1.1. Definição do problema

Há uma demanda emergente por soluções que permitam resgatar informações textuais contidas em vídeos digitais. Dentre estas, destaca-se o problema de localização de códigos de identificação de vagões filmados por questões de segurança ou controle logístico.

Os principais fatores de complexidade desta aplicação são: a grande variabilidade no formato e pintura dos vagões; e a ausência de um padrão que defina fonte e cor dos códigos a serem localizados. Além da precária manutenção dos vagões.

A Figura 1.1 apresenta um exemplo da composição dos códigos de vagões (3 letras e 7 dígitos). Embora exista um padrão para a composição, não existe uma definição precisa quanto ao local onde este deve aparecer no vagão. Na Figura 1.2 percebe-se que o código foi pintado no meio do vagão tanque. Esse modelo de vagão possui uma superfície cilíndrica e em algumas situações o código do vagão pode estar inclinado, dificultando ainda mais a localização do código de identificação do vagão.

Figura 1.1 – Código do vagão no canto superior esquerdo.

Figura 1.2 – Código do vagão quase no meio.

As intempéries têm grande influência ocasionando o desgate de pintura, ou ainda, a presença de sujeiras no vagão, conforme apresentado na Figura 1.3.

Figura 1.3 – Código do vagão sujo.

O vandalismo, apresentado na Figura 1.4, é outro fator que deve ser levado em consideração no processo de localização dos códigos de identificação. Freqüentemente são encontrados vagões pichados.

Figura 1.4 – Código do vagão pichado.

O desgaste do vagão, em alguns casos, impossibilita que haja a localização do código conforme apresentado na Figura 1.5. Além disso, o espaço do vagão pode ser cedido para que empresas parceiras pintem a sua marca no vagão.

Figura 1.5 – Código do vagão ilegível.

1.2. Objetivos

O objetivo principal dessa pesquisa é desenvolver um método para a localização de códigos de identificação de vagões de diferentes modelos presentes em cenas de vídeo digital. Para isso serão realizados os seguintes passos:

a) Elaborar um levantamento bibliográfico sobre técnicas de localização de texto em imagens e vídeos
b) Construir uma base de vídeos contendo cenas de composições de vagões de diferentes modelos
c) Desenvolver um *framework* para avaliar o método proposto

1.3. Desafios

Separar fundos complexos dos vídeos coletados, melhorar e ajustar o contraste, avaliar as condições de iluminação e as possíveis distorções de perspectivas a fim de realizar a localização do código de identificação do vagão.

1.4. Motivação

A principal motivação para a realização desse trabalho é a possibilidade de disponibilizar uma ferramenta futura para localização e leitura automática de códigos em vagões.

Atualmente os trabalhos nessa área são dirigidos a localizar textos em vídeos e grande parte deles são textos adicionados artificialmente. A proposta é localizar textos impressos em objetos (vagões) que fazem parte da cena.

Além da motivação tecnólogica, pode-se destacar como motivação científica a possibilidade de avaliar diferentes técnicas de visão computacional e processamento de imagens em aplicação, prática e complexa, voltada à localização de textos em imagens de vídeo.

1.5. Contribuições

A principal contribuição desse trabalho é apresentar uma nova abordagem de localização de texto em vídeo, em particular códigos de identificação de vagões. Também

espera-se contribuir com novos experiementos que agreguem valor à área de visão computacional.

1.6. Organização

Essa dissertação de mestrado está organizada em 5 (cinco) capítulos. Após uma breve introdução, o Capitulo 2 apresenta o estado da arte com alguns dos principais trabalhos diretamente relacionados ao tema de pesquisa desta dissertação. O Capitulo 3 apresenta todos passos do método proposto, enquanto, o Capitulo 4 descreve os experimentos realizados com o método proposto. Finalmente, o Capitulo 5 apresenta as considerações finais e os trabalhos futuros a serem realizados.

Capítulo 2

Estado da Arte

Este capítulo apresenta alguns conceitos referentes à detecção, localização e extração de textos em imagens de vídeo e também alguns trabalhos relacionados ao método proposto nesta dissertação.

Nos últimos anos, vários algoritmos de localização e extração de textos em imagens de vídeo vêm sendo propostos, apesar de existirem vários estudos sobre esse tema, ainda não é fácil projetar um sistema de propósito geral para a localização e extração de informações de textos em imagens de vídeo. A localização de texto é ainda um problema desafiador pelo fato dos vídeos freqüentemente terem uma baixa resolução, fundo complexo e textos com diferentes tamanhos, estilos e alinhamentos. Além do texto poder ser afetado pelas condições de iluminação e distorções de perspectivas.

O reconhecimento de texto em vídeo é geralmente dividido em quatro etapas, a saber: detecção, localização, extração e reconhecimento conforme ilustrado no fluxograma da Figura 2.1. A etapa de detecção de texto classifica as regiões de texto e, não texto, baseando-se em características particulares de cada uma delas. A etapa de localização de texto determina os limites precisos do texto detectado. Já a etapa de extração é responsável por extrair características para posterior reconhecimento.

As etapas de detecção, localização e extração acima citadas, dizem respeito, basicamente, à preparação dos dados (imagens) para serem utilizados na etapa de reconhecimento que nada mais é que realizar o reconhecimento do texto na imagem. Normalmente para o reconhecimento do texto os três passos acima se encarregam de gerar uma imagem binária (do texto). O reconhecimento pode ser feito por uma ferramenta comercial de OCR (*Optical Character Recognition*). Existem inúmeras pesquisas que

estudam apenas a fase de reconhecimento, nessa dissertação não será abordada essa etapa em maiores detalhes.

Figura 2.1 – Fluxograma das etapas do reconhecimento de texto em imagens de vídeo.

2.1. Detecção de Textos em Imagens de Vídeo

A etapa de detecção de textos de vídeos é utilizada para classificar as regiões de texto e não texto baseando-se em características particulares de cada uma delas. De modo geral as principais características são:

- **Contraste**: no texto sobreposto é utilizada a diferença de brilho entre as áreas claras e escuras de uma imagem a fim de fornecer informações específicas. Parte do princípio que deve existir contraste suficiente com o fundo para a extração das informações.
- **Cor**: as linhas de textos geralmente têm uma cor uniforme. Parte do princípio que essa uniformidade nas cores pode viabilizar a extração das informações.
- **Tamanho da fonte**: deve ter um tamanho adequado, geralmente existe uma variação que não é exagerada.
- **Formato da fonte**: devido à densidade da linha do texto, é possível manipular as informações de bordas para detectar o texto em vídeo.
- **Orientação**: utiliza orientação vertical ou horizontal pré-definida para realizar a detecção do texto.

Considerando as características acima pode-se dizer que a detecção de textos é apresenta três classes. A primeira classe trata o texto como sendo um tipo de textura, a segunda classe assume que os textos têm cores uniformes e a terceira e última classe utiliza informações das bordas do texto.

2.1.1 Detecção baseado em textura

Na detecção baseada em textura é assumido que os textos de vídeos têm uma freqüência de textura similar e uma única orientação. Por esse motivo, tipos de texto de vídeo podem ser tratados como um tipo especial de textura. Essa técnica geralmente consiste em dividir toda a imagem em blocos menores.

Em um primeiro momento essa técnica consiste em avaliar as características da textura do bloco. Para isso podem ser utilizadas as abordagens que utilizam filtro de Gabor, variância espacial ou transformada de *wavelet*. Em um segundo momento é utilizado um classificador de padrões, para identificar os blocos de textos.

Li [Li, 2000] propôs como extração de características a utilização da transformada *wavelet* em três sub-bandas (*LH Horizontal High Frequency, HL Vertical High Frequency, HH Horizontal e Vertical High Frequency*) por entender que dessa forma é possível obter aproximações sucessivas da imagem detectando as bordas em uma filtragem passa-alta (*highpass filtering).* Após as características serem extraídas e selecionadas, uma rede neural é treinada para eliminar as falsas regiões de texto. O método permite tratar imagens com fundo simples e complexo, podendo realmente detectar texto de vídeos em imagens borradas (o contraste do texto de vídeo não é alto suficiente em relação ao fundo). Contudo, o inconveniente dessa técnica está no custo do treinamento necessário que muitas vezes inviabiliza sua aplicação prática.

2.1.2. Detecção baseada em cor

A detecção baseada em cor pressupõe que o texto de vídeo contém cores uniformes. Por esse motivo essa técnica é normalmente utilizada em vídeos com fundo simples, ou seja, de cor uniforme. A utilização dessa técnica é basicamente dividida em dois momentos. No primeiro momento é realizada uma redução de cores seguida de uma segmentação utilizando os canais de cor do espaço de cor escolhido. No segundo momento é executada uma análise de componentes conectados (*connected-component*) para detectar as regiões de textos.

Jain, em [Jain, 1998] propôs um método para tratar imagens coloridas. O objetivo é avaliar a similaridade de diferentes valores de cor. No espaço de cor RGB (*Red, Green e Blue*) os valores variam de 0 a 255 para cada canal, resultando em 256^3 diferentes valores de cor. Por esse motivo uma redução de cor faz-se essencial para aumentar a velocidade de avaliação de similaridade. Jain, em [Jain, 1998] adotou uma redução de *bits* e um método de quantização para executar a redução de cores. Em seu trabalho foram necessários 8 *bits* para representar os canais R, G e B. Ele separou os 6 menores *bits* de modo que os 2 *bits* maiores permanecessem. Com apenas 6 bits à esquerda para medida de similaridade foi possível reduzir os valores das cores notavelmente, de 2^{24} para 2^6. Após isso foi aplicado um método de histograma para quantizar a cor, em outras palavras, foi realizada uma mescla nas regiões de cores similares para descobrir a região de texto.

Com essa técnica, detecta-se primeiramente uma região da imagem contendo apenas um caractere. No entanto, em seguida, conecta-se a outros caracteres próximos para formar uma cadeia de caracteres significativos. Esse processo é realizado por uma análise de componentes conectados (*connected-component*) que é dividida em três passos:

- **Primeiro passo**: consiste em formar uma região pelos *pixels* de textos detectados e seus vizinhos (*pixels* com cores similares) e encontrar o menor retângulo para essa região a qual é chamada de componente de texto.
- **Segundo passo**: consiste em determinar, de acordo com o tamanho do texto, um limiar para filtrar as regiões de texto falsas (o retângulo das regiões formadas no primeiro passo).
- **Terceiro passo**: consiste em conectar o texto por uma mesma linha horizontal. Partindo do pressuposto que nos casos mais comuns os textos estão orientados na horizontal e uma seqüência de texto significativo é formada por retângulos de texto.

2.1.3. Detecção baseada em bordas

Na detecção baseada em bordas são utilizadas a densidade do traço e características de contraste. Com isso é possível utilizar essa técnica a fim de detectar uma região de texto na imagem de vídeo ou em um quadro [Zhong, 1999].

Nessa técnica um mapa de bordas é primeiramente gerado e um método de mescla é utilizado para conectar os caracteres de texto formando uma cadeia significativa. Essa técnica

geralmente inclui uma análise dos componentes conectados (*connected-component*) e operações morfológicas. Também permite detectar textos de vídeo sobrepostos com fundo simples e complexo. Nas cenas de vídeo, esta técnica é indicada quando houver texto de vídeo com alto contraste e fundo simples. Contudo, pode-se detectar objetos pequenos como sendo caracteres de texto caso estes possuam densidade no traço e características de contraste similares.

A detecção de texto baseada em bordas pode ser classificada em duas categorias: a de domínio da compressão (*compressed domain*) onde o vídeo está em um formato comprimido e domínio do *pixel* (*pixel domain*). No domínio da compressão é utilizado o coeficiente de DCT (*Discrete Cosine Transform*) para medir a variação da intensidade horizontal e vertical em um bloco DCT de um quadro. Já no domínio do *pixel* são empregados filtros como de Canny, Sobel ou Gaussiano (*Gausian)* para executar a detecção da borda [Zhong, 1999].

2.2. Localização de Textos em Imagens de Vídeo

Grande parte das pesquisas de localização foca geralmente em textos com um posicionamento fixo, no entanto, nem sempre as regiões de texto estão na mesma posição. Neste caso, o texto pode ser dividido em três classes: texto estático (na mesma posição), movimento simples linear (por exemplo, rolagem de crédito de filme quando o filme acaba) e movimento complexo não linear (por exemplo, aumento e diminuição de *zoom*, rotação e movimentos livres na cena).

Normalmente existem três abordagens para localização do texto em imagens de vídeo: a primeira abordagem assume que o texto é estático [Wolf, 2002]; a segunda abordagem é baseada no valor do *pixel* do texto que pode os textos estáticos como textos com movimento simples linear [Li, 2000][Zhang, 2003]; e na terceira abordagem aplicam-se os perfis de projeção horizontal e vertical para localizar o texto [Lienhart, 2002] [Gao, 1983] [Wernicke, 2000] [Cai, 2002].

2.2.1. Localização em texto estático

Segundo Christian Wolf [Wolf, 2002] uma vez que a região do texto é detectada, é assumido que o texto é estático. Pode ser localizada a região de texto no mesmo lugar em sucessivos quadros a fim de reduzir os cálculos da detecção do texto. O processo de detecção

da região de texto é baseado no número máximo de quadros sucessivos no qual um único texto pode estar para evitar a falsa localização.

2.2.2. Localização por valor do *pixel*

Li, em [Li, 2000] propôs um método robusto para localizar textos estáticos e texto em movimento (movimento simples linear e movimento complexo não linear). O processo de detecção ainda é determinado pelo número máximo de quadros sucessivos no qual um único texto pode estar. Para reduzir o cálculo de detecção de texto, Li em [Li, 2000] propôs dois métodos. O primeiro método é o SSD (*Sum of Square Difference*) ou a soma das diferenças quadráticas baseado em correspondência de valores do *pixel* que pode tratar textos estáticos ou textos com movimentos lineares simples. O segundo método é baseado na estabilização do texto pelo contorno que trata texto com movimento complexo não linear.

2.2.3. Localização por perfil de projeção

Lienhart, em [Lienhart, 2002] propôs um método de perfil de projeção para localizar blocos de texto em imagens. O perfil de projeção de uma região da imagem é uma representação compacta da distribuição espacial dos *pixels* e tem sido empregado com sucesso na segmentação de documentos de texto. Enquanto o histograma apenas captura a freqüência de distribuição de algumas características das imagens assim como a intensidade do *pixel* (toda a informação espacial é perdida), o perfil de projeção preserva a distribuição espacial bruta mantendo uma maior agregação aos valores de conteúdo dos *pixels*.

Baseado no método de perfil de projeção, pode-se localizar o texto estático ou um movimento simples no bloco de texto. Quanto ao movimento complexo não-linear, ele só localiza partes de todo o bloco de texto. Para evitar esse tipo de problema Lienhart, em [Lienhart, 2002] propôs detectar a imagem inteira a cada quatro quadros.

2.3. Extração de Texto em Imagens de Vídeo

Após os processos de detecção e localização serem realizados, ainda é importante que seja feita a extração do texto para poder realizar o reconhecimento propriamente dito. Para isso é preciso realizar um processo de binarização da imagem quando os caracteres do texto são segmentados a partir do fundo.

O método de extração do texto pode ser dividido em dois grupos, um grupo inclui métodos baseados em cor [Antani, 2000] [Lyu 2005], e o outro grupo inclui métodos baseado no traço (*stroke-based*). O primeiro grupo parte do princípio que os *pixels* de texto têm cor diferente em relação aos *pixels* de fundo, assim sendo eles podem ser segmentados por um elo limiar. Por outro lado, os métodos baseados no traço empregam filtros buscando selecionar os *pixels* que pertencem ao tal como o filtro assimétrico [Chen, 2001], filtro de extração de caractere de quatro direções [Sato, 1998] e a máscara e características topográficas aplicadas em [Chun, 1999].

Devido à pequena resolução e ao ruído normalmente presente nas imagens de vídeo em diferentes aplicações, é comum buscar o aumento do contraste do texto em relação ao fundo. Existem vários métodos de melhoramento de imagens entre eles a de integração de múltiplos quadros (*Multiple Frame Integration*) proposto em[Hua, 2002].

2.3.1. Integração de Múltiplos Quadros

A integração de múltiplos quadros parte do princípio que as cadeias de caracteres de uma imagem são estáticas (o texto permanece na mesma posição durante vários quadros sucessivos). Ocorre que durante a sucessão dos quadros o valor do *pixel* de um fundo complexo pode variar, entretanto, o valor do *pixel* no texto é estático.

Sato, em [Sato, 1998] propôs um método para encontrar o valor mínimo do *pixel* de texto na mesma posição de um bloco de texto durante sucessivos quadros, onde em N quadros sucessivos é encontrado o menor valor de *pixel* em uma mesma posição de um bloco de texto. Para isto, o valor do *pixel* para uma cadeia de caractere deve permanecer próximo a 255 (cor branca), mas o fundo sofre um "desgaste" devido à variação do valor do *pixel* do fundo. Após isso, o contraste entre a cadeia de caracteres e o fundo é reforçado a fim de facilitar a extração do texto.

Kwak, em [Kwak, 2000] propôs um método similar ao de Sato [Sato, 1998], onde utiliza-se uma cadeia de caracteres estática e a cor branca, entretanto, não é executado o processo de classificação para encontrar o menor valor do *pixel*, ao invés disso, é executada uma operação lógica (AND). A grande vantagem desse método é o baixo custo computacional pela razão que uma operação binária é muito mais rápida do que uma operação de comparação. A representação binária da cor branca é 11111111 (255 em decimal), caso uma operação AND seja executada em N quadros sucessivos a cadeia de caractere ainda pode

permanecer próximo de 11111111, entretanto, o fundo sofre uma degradação pela razão que a operação binária de 1 AND 1 permanece 1. Esse método também melhora o contraste e o fundo deve ser reforçado a fim de facilitar a extração do texto.

2.3.2. Interpolação

Após o processo de localização de textos é muito comum encontrar imagens com baixa resolução o que faz necessário que a resolução da imagem seja melhorada. Para resolver esse problema, uma técnica adotada é a de interpolação da imagem original.

Wolf, em [Wolf, 2002] propôs uma interpolação bilinear e bicúbicas (*bi-cubic*) para aumentar a resolução. No algoritmo bilinear, o valor de cinza de cada *pixel* $F'_i(p', q')$ é uma combinação linear dos valores de cinza de seus quatro vizinhos. (Equação 1),

$(F_i(p,q), F_i(p+1, q), F_i(p,q+1), F_i(p+1, q+1))$ sendo:

$$p = \left\lfloor \frac{p'}{u} \right\rfloor \qquad q = \left\lfloor \frac{q'}{u} \right\rfloor \tag{1}$$

sendo u o fator de interpolação.

Os pesos de $w_{m.n}$ para cada vizinho $F_i(p + m, q + n)$ $(m,n \in [0.1])$ dependem da distância entre o *pixel* e o respectivo vizinho (calculado através das distâncias horizontal e vertical a e b respectivamente, entre o *pixel* e o vizinho de referência $F_i(p,q)$) (Equação 2):

$$\begin{aligned} a &= \frac{p'}{u} - \left\lfloor \frac{p'}{u} \right\rfloor \\ b &= \frac{q'}{u} - \left\lfloor \frac{q'}{u} \right\rfloor \end{aligned} \tag{2}$$

Nesta distância os pesos são calculados conforme a Equação 3:

$$\begin{aligned} w_{0,0} &= (1-a)*(1-b) \\ w_{1,0} &= a*(1-b) \\ w_{0,1} &= (1-a)*b \\ w_{1,1} &= a*b \end{aligned} \tag{3}$$

O *pixel* interpolado $F'_i(p',q')$ de um dado quadro $F_{i:}$ é obtido pela Equação 4.

$$F_i^{'} = (p^{'}, q^{'}) = \frac{\sum_{m=0}^{1}\sum_{n=0}^{1} w_{m,n} F_i(p+m, q+n)}{\sum_{m=0}^{1}\sum_{n=0}^{1} w_{m,n}} \tag{4}$$

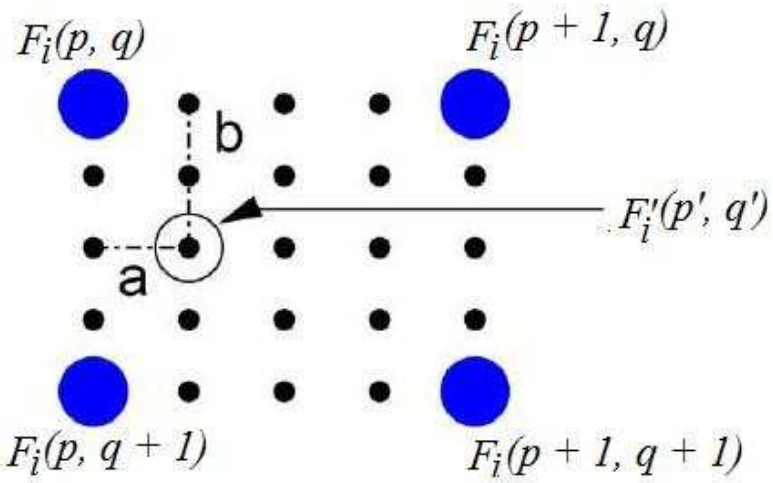


Figura 2.2 – Interpolação bilinear (fator 4X), as variáveis *a* e *b* significam as distâncias entre o *pixel* e seus vizinhos [Wolf, 2002].

No método de interpolação bicúbica é passado um polinomial cúbico através dos pontos vizinhos ao invés de uma função linear. Com isso, são considerados dezesseis pontos ao invés de quatro pontos de vizinhança no cálculo de uma nova imagem $F'_i(p',q')$. Assim como no esquema da interpolação bilinear, os pesos para os diferentes vizinhos $F_i(p + m, q + n)$ dependem das distâncias do *pixel* interpolado e calculado usando as distâncias *a* e *b* para o ponto de referência $F_i(p,q)$. No entanto, ao invés de ajustar os pesos proporcionais nas distâncias, o peso $w_{m,n}$ para cada vizinho é calculado como na Equação 5:

$$w_{m,n} = R_c(m-a)R_c(-(n-b)) \tag{5}$$

Sendo R_c um polinomial cúbico (Equação 6 e Equação 7)

$$R_c = \frac{1}{6}(x^3 - 3x^2 - 12x + 9) \tag{6}$$

e

$$(x)^m = \begin{cases} x^m \, se \; x > 0 \\ 0 \;\; se \; x \leq 0 \end{cases} \tag{7}$$

o *pixel* interpolado $F_i^{'}(p^{'}, q^{'})$ pode portanto ser calculado pela Equação 8:

$$F_i^{'}(p^{'}, q^{'}) = \frac{\sum_{m=-1}^{2} \sum_{n=-1}^{2} w_{m,n} F_i(p+m, q+n)}{\sum_{m=-1}^{2} \sum_{n=-1}^{2} w_{m,n}} \tag{8}$$

A Figura 2.3 ilustra a interpolação bicúbica de fator quatro. Na Figura 2.4 obersva-se um exemplo da interpolação bilinear e bicúbica e suas binarizações correspondentes.

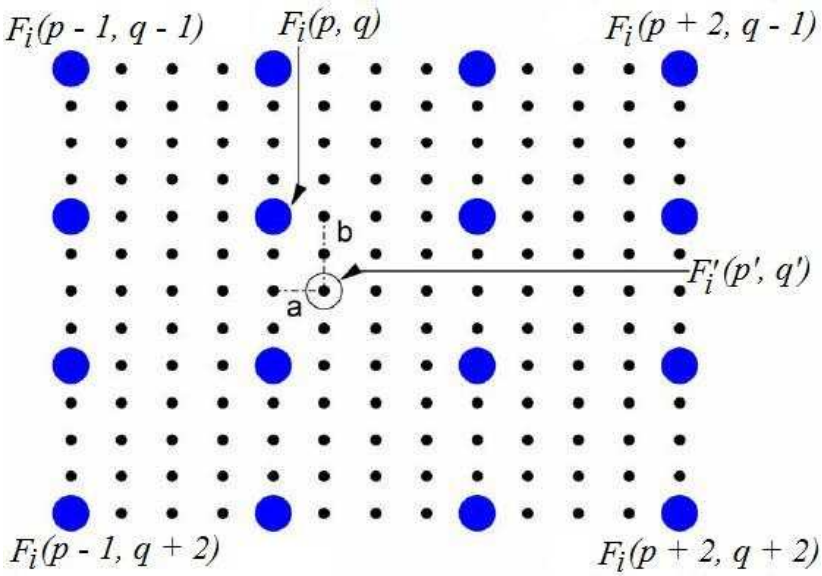


Figura 2.3 – Interpolação bicúbica (fator 4X), as variáveis *a* e *b* significam as distâncias entre o *pixel* e seus vizinhos [Wolf, 2002].

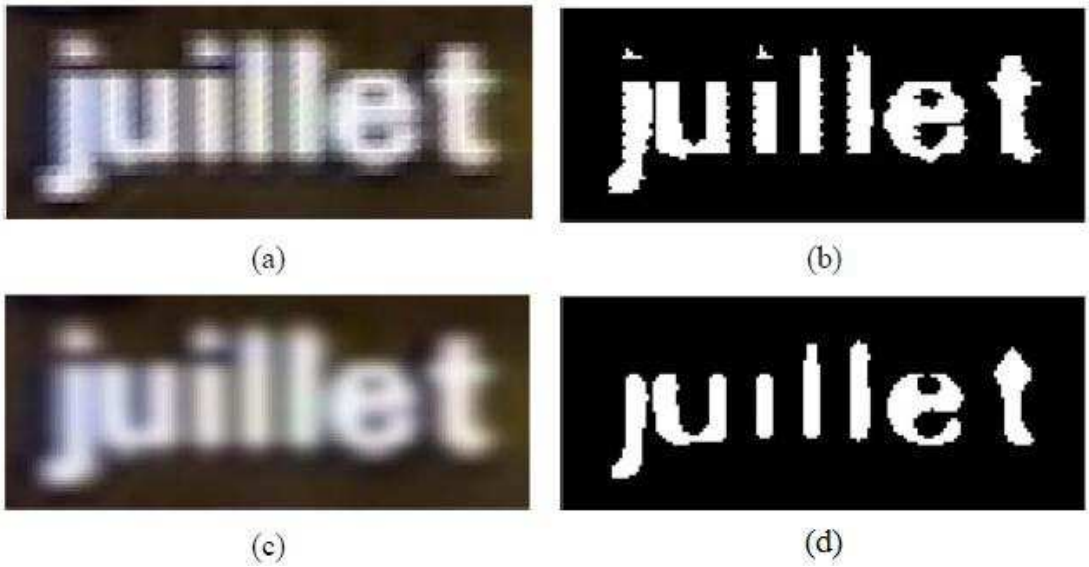


Figura 2.4 – Imagens interpoladas: (a) interpolação bilinear, (b) resultado da binarização da imagem interpolada bilinear, (c) imagem bicúbica, (d) resultado da binarização da imagem interpolada bicúbica [Wolf, 2002].

2.4. Trabalhos Diretamente Relacionados

Esse tópico apresenta alguns trabalhos diretamente relacionados ao tema desta dissertação. Abordando a localização de textos em imagens de vídeo, às técnicas adotadas pelos autores e os resultados obtidos.

Trung, em [Trung, 2009] utiliza o operador Laplaciano a fim de detectar os textos em imagens de vídeo em três passos: 1) um operador Laplaciano é usado para detectar as regiões de textos candidatas; 2) realiza-se um refinamento de fronteiras quando uma análise do perfil de projeção determina a fronteira exata para cada bloco de texto; 3) realiza-se uma filtragem des falsos positivos baseada em propriedades geométricas. Segundo Trung [Trung, 2009] os métodos de detecção de textos podem ser classificados em três abordagens: a primeira abordagem é baseada em componente conectado (*connected component-based*) que não funciona muito bem para todas as imagens de vídeo devido assumir que os *pixels* de textos estão em uma mesma região e têm cores similares ou uma mesma intensidade de cinza. A segunda abordagem é baseada em bordas (*edge based*) a qual exige que o texto possua um alto contraste e um fundo simples para detecção das bordas. Tanto na primeira como na segunda abordagem observa-se um grande problema relacionado ao tratamento de fundos complexos o qual produz muitos falsos positivos. A terceira abordagem é baseada em textura (*texture-based*) considerando o texto um tipo especial de textura, e emprega as transformadas

de Fourier, DCT (*Discrete Cosine Transform*), decomposição por *wavelet* e filtros de Gabor na extração de características. Um possível problema desta abordagem é o alto custo computacional quando for indispensável trabalhar com grandes quantidades de imagens.

Para eliminar as descontinuidades, Trung [Trung, 2009] converteu a imagem de entrada para escala de cinza e filtrou-a utilizando a máscara do Laplaciano de 3 x 3 para detectar as descontinuidades nas direções horizontal, vertical, para cima à esquerda e para baixo à direita. A máscara do Laplaciano 3x3 é mostrada na Figura 2.5.

1	1	1
1	-8	1
1	1	1

Figura 2.5 – Máscara do Laplaciano 3 x 3.

A máscara produz dois valores para cada borda. A imagem filtrada por Laplaciano contém valores positivos e negativos, sendo que as transições entre os valores (cruzamento entre os zeros) correspondem às transições entre o texto e o fundo. A fim de capturar o relacionamento entre os valores positivos e negativos é utilizada a diferença do máximo gradiente (MGD - *Maximum Gradient Difference*) entre os valores máximos e mínimos dentro de uma região de 1 x *N* janelas [Wong, 2003]. O valor de MGD no *pixel* (i,j) é calculado na imagem filtrada por Laplaciano f . A Equação 9 mostra a fórmula do MGD:

$$MGD(i, j) = \max(f(i, j-t)) - \min(f(i, j-t)) \qquad (9)$$

sendo $t \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$. O mapa MGD é obtido pelo movimento da janela sobre a imagem. Ainda na fase de detecção de texto Trung [Trung, 2009] normalizou o mapa MGD para o intervalo [0,1] e usou o algoritmo de *K*-médias (*K-means*) [Wong, 2003] para classificar os *pixels* em dois grupos textos e não textos. A Figura 2.6 mostra a imagem original e os resultados obtidos no processo de detecção.

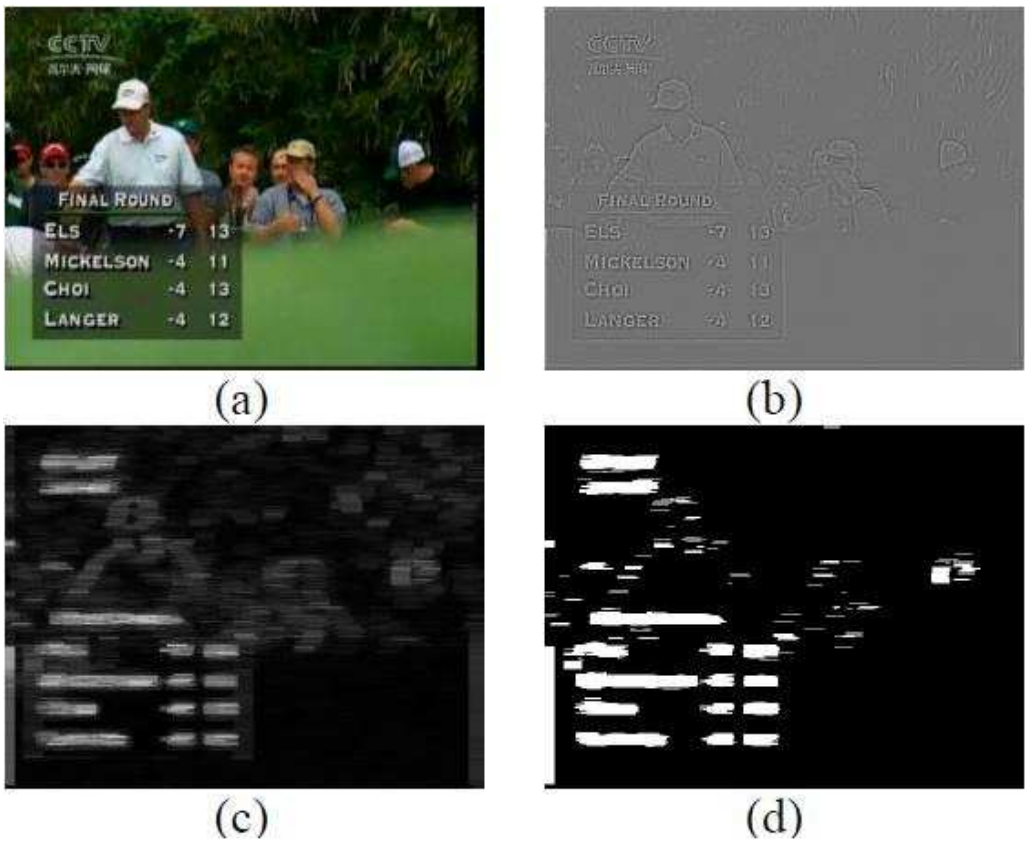


Figura 2.6 – Passo de detecção de texto, (a) imagem original, (b) imagem filtrada por Laplaciano, (c) mapa da diferença do máximo gradiente e (d) grupos de textos.

Embora no passo de detecção seja possível encontrar grande parte dos grupos de texto, ainda sim é muito difícil detectar onde estão as fronteiras dos blocos de texto. Para resolver esse problema Trung [Trung, 2009] executa o passo de refinamento de fronteiras, onde é calculado o mapa de borda binário de Sobel *SM* na imagem de origem (somente para as regiões de texto). O perfil de projeção é definido nas Equações 10 e 11.

$$HP(i) = \sum_{j} SM(i, j) \tag{10}$$

Se *HP(i)* é maior que um certo limiar, a linha *i* é parte da linha de texto, caso contrário ela é parte da lacuna entre diferentes linhas de textos. A partir dessa regra pode-se determinar a linha de cima i_1 e a linha de baixo i_2 para cada linha de texto.

$$VP(j) = \sum_{i=i_1}^{i_2} SM(i, j) \tag{11}$$

O perfil de projeção vertical é definido pela Equação 11 e é similar ao perfil horizontal, se *VP(j)* é maior que certo limiar, a coluna *j* é parte da linha do texto, caso contrário é parte da lacuna entre diferentes palavras. Finalmente, diferentes palavras na mesma linha de textos são mescladas, se estão próximas umas das outras. Esse processo é aplicado recursivamente a fim de determinar a precisão da fronteira para cada bloco de texto. A Figura 2.7 mostra os resultados obtidos utilizando o passo de refinamento de fronteiras.

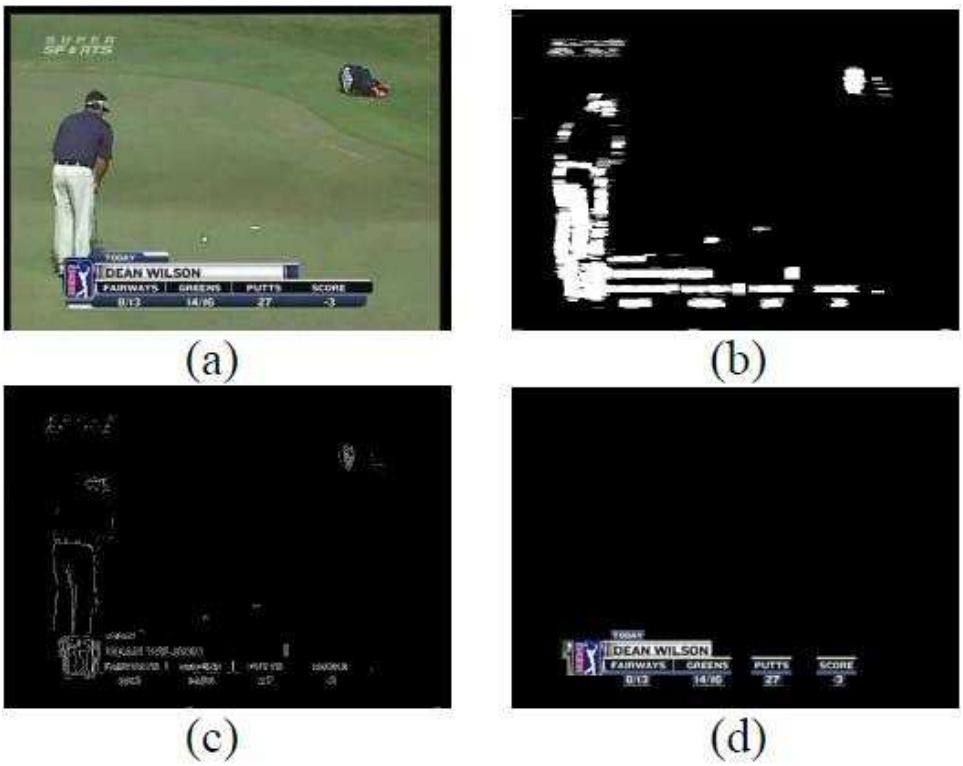


Figura 2.7 – Passo de refinamento de fronteira, (a) imagem original, (b) grupo de textos, (c) mapa de borda Sobel e (d) blocos de textos.

O último passo proposto por Trung [Trung, 2009] consiste na eliminação de falsos positivos sendo baseado em propriedades geométricas. Com a largura *(W)*, altura *(H)*, relação largura/altura (ou relação de aspecto, *AR)*, área *(A)* e área de borda *(EA)* de um bloco de texto *(B)* são formuladas as seguintes equações: a Equação 12 determina a relação de aspecto, na Equação 13 determina a área e na Equação 14 é determinada a área de borda.

$$AR = W \div H \tag{12}$$

$$A = W \times H \tag{13}$$

$$EA = \sum_{(i,j)\in B} SM(i,j) \tag{14}$$

se $AR < T_1$ ou $EA / A < T_2$ o bloco candidato é considerado como um falso positivo, caso contrário é aceitado com um bloco de texto.

A fim de medir a eficiência do método, Trung [Trung, 2009] definiu as seguintes categorias para cada bloco de texto detectado pelo método proposto:

- ✓ Bloco verdadeiramente detectado (TDB): um bloco detectado que contém uma linha de texto parcialmente ou totalmente detectada.
- ✓ Bloco falso detectado (FDB): um bloco detectado que não contém texto.
- ✓ Bloco de texto com dados faltando (MDB): um bloco de texto detectado que perdeu alguns caracteres da linha de texto.

Para cada imagem no banco de dados, Trung [Trung, 2009] contou manualmente os blocos de textos (ATB) totalizando 491. Bloco de textos verdadeiramente detectados (TDB) totalizou 458. Bloco falso detectado (FDB) totalizou 39. Bloco de texto com dados faltando (MDB) totalizou 55.

As medidas das taxas de desempenho adotadas por Trung [Trung, 2009] foram as seguintes:

- ✓ Taxa de detecção (DR) = TDB / ATB.
- ✓ Taxa de falso positivo (FPR) = FDB / (TDB + FDB).
- ✓ Taxa de detecção perdida (MDR) = MDB / TDB;

Aplicando as fórmulas para calcular as taxas, o método proposto obteve uma taxa de detecção (DR) de 93,3%, uma taxa de falso positivo (FPR) de 7,9% e uma taxa de detecção perdida (MDR) de 12,0%.

Jian, em [Jian, 2009] propôs uma nova abordagem utilizando a integração por múltiplos quadros (MFI – *Multiple Frame Integration*]. Segundo Jian [Jian, 2009] os métodos convencionais de localização de textos de vídeos utilizando MFI [Hua, 2002] geralmente focam apenas nas duas fases chaves do método: identificação do bloco de texto (*Text-Block Group Identification)* e na integração (*Text-Block Group Integration).*

Nessa nova abordagem adotada por Jian [Jian, 2009] o método de MFI passa a ter três fases não mais duas como nas abordagens anteriores. A Figura 2.8 mostra o fluxograma da abordagem proposta por Jian [Jian, 2009] o qual é composto por quatro fases: detecção de

texto, MFI, extração de texto e OCR, lembrando sempre que o foco dessa abordagem de Jian [Jian, 2009] é a fase de MFI.

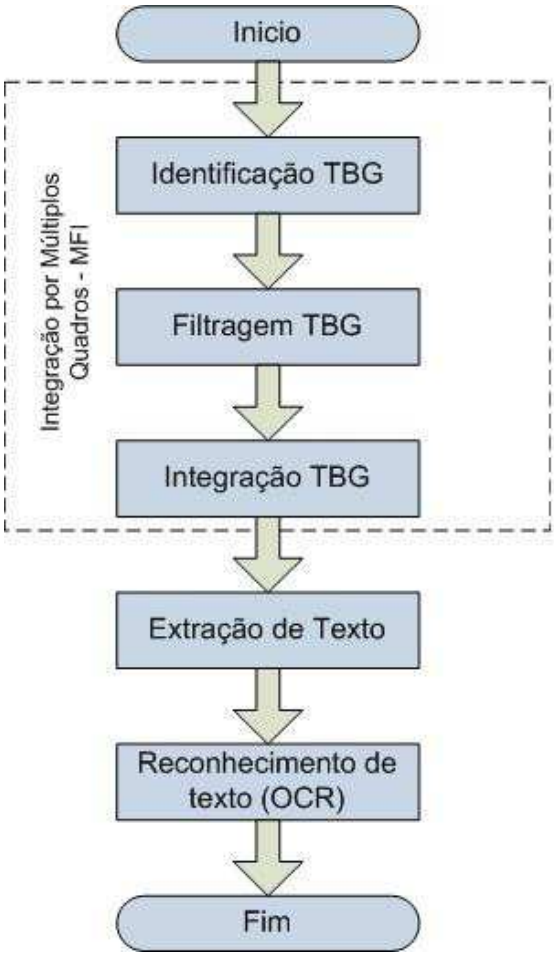


Figura 2.8 – Fluxograma do método MFI proposto por [Jian, 2009].

Na fase de identificação (Identificação TBG) são identificados os blocos de textos com o mesmo texto considerando a localização, distribuição de borda e o contraste do bloco de texto. Como os blocos de textos nos quadros dos vídeos são contínuos, estes são considerados os mesmos textos somente se tiverem três características semelhantes. A primeira é que o mesmo texto existindo em múltiplos quadros de vídeo geralmente mantém a mesma localização. A segunda é que o mapa da borda do bloco de texto da imagem principal contém as bordas do texto. A terceira é que o contraste do bloco de texto é determinado pela diferença entre o texto e o fundo, assim sendo o bloco com o mesmo texto deveria ter o contraste semelhante da imagem. Na fase de filtragem (Filtragem TBG) é medida a claridade do texto

usando o mapa de intensidade onde são selecionados os blocos com textos “limpos” para integração. O mapa de intensidade do texto é detectado pelo uso dos quatros detectores de intensidade do texto. A Figura 2.9 mostra os quatros detectores das intensidades do texto do mapa de intensidades utilizadas para detectar a intensidade dos textos na imagem.

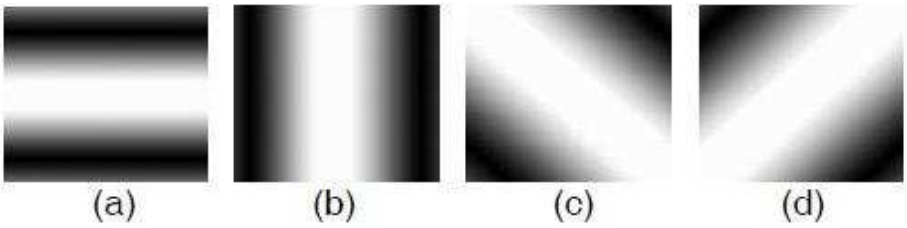


Figura 2.9 – Quatro detectores das intensidades do texto, (a) detector da intensidade do texto horizontal, (b) detector da intensidade do texto vertical, (c) detector da intensidade do texto da vertical esquerda e (d) detector da intensidade do texto da vertical direita.

Na fase de integração (Integração TBG) os blocos de textos são integrados utilizando as integrações médias e mínimas do texto e do fundo da imagem para obter o fundo e o texto limpo com alto contraste para o reconhecimento.

Para avaliar o desempenho da abordagem proposta por Jian [Jian, 2009], foi montado um banco de dados experimental contendo 10 vídeos da *web*, coletados de vários *sites* chineses famosos. A escolha por utilizar vídeos da *web* nesse trabalho foi porque geralmente as imagens são de fundos complexos, baixo contraste e texto borrado. Jian [Jian, 2009] rotulou manualmente as linhas de textos e contou a quantidade de caracteres chineses nos vídeos totalizando 1809 linhas de textos diferentes e 11312 caracteres chineses nos vídeos.

Três métricas são adotadas para a avaliação: Revocação, Precisão e Repetição. Um alto valor na Revocação indica a habilidade superior para reconhecer caracteres relevantes, enquanto alto valor na Precisão indica alta taxa de reconhecimento com os caracteres corretos. A Repetição é utilizada porque o último texto pode durar por muito tempo e pode ser detectado e reconhecido repetidamente. O desempenho para a abordagem do reconhecimento do texto é principalmente determinado pela chamada e precisão ao invés da repetição, porque o reconhecimento dos caracteres é muito mais importante que o reconhecimento de caracteres repetidamente. Estas métricas são definidas da seguinte forma:

- ✓ Revocação = $CN_{correto}$ / $CN_{verdadeiro}$.
- ✓ Precisão = $CN_{todoscorreto}$ / CN_{todos}.
- ✓ Repetição = $CN_{repetição}$ / CN_{todos}

sendo $CN_{todoscorreto} = CN_{correto} + CN_{repetição}$, $CN_{todoscorreto}$ é o número de caracteres reconhecido corretamente com os caracteres repetidos, $CN_{correto}$ é o número de caracteres reconhecido sem caracteres repetidos e $CN_{repetição}$ é o número de caracteres reconhecido corretamente e repetidamente. $CN_{verdadeiro}$.é o número de caracteres verdadeiro no fundo e CN_{todos}.é o número de caracteres reconhecido.

Os resultados obtidos pelo método proposto por Jian [Jian, 2009] foram os seguintes: a Revocação obteve uma taxa de 57,43 %, Precisão obteve uma taxa de 60,43 % e Repetição obteve uma taxa de 8,01 %.

Pratheeba, em [Pratheeba, 2010] propôs uma abordagem utilizando a detecção do texto baseado na morfologia e extração de cenas de vídeo complexas utilizando um mapa binário morfológico. O mapa é gerado pelo cálculo da diferença da operação morfológica de abertura e de fechamento da imagem. Após isto, as regiões candidatas são conectadas usando uma operação morfológica de dilatação. As regiões de texto são determinadas com base na ocorrência do texto em cada bloco candidato. As regiões de texto detectadas são localizadas com precisão usando a projeção de *pixels* texto no mapa binário morfológico e a extração de texto é finalmente realizada.

Com a finalidade de detectar as regiões de textos de um fundo complexo foi utilizada uma abordagem baseada na morfologia para extrair as características de alto contraste da imagem. Para montar o mapa binário morfológico Pratheeba [Pratheeba, 2010] utilizou as seguintes operações morfologicas:

Operação de fechamento

$$I(x,y) \bullet S_{m,n} = (I(x,y) \oplus S_{m,n}) \square S_{m,n} \tag{15}$$

Operação de abertura

$$I(x,y) \circ S_{m,n} = (I(x,y) \square S_{m,n}) \oplus S_{m,n} \tag{16}$$

Diferença (17)

$$D(I_1,I_2) = |I_1(x,y) - I_2(x,y)|$$

Limiarização

$$T(I(x,y)) = \begin{cases} 255, \text{ if } I(x,y) > T \\ 0 \text{ , Caso contrário} \end{cases} \tag{18}$$

Sendo *I(x,y)* a imagem de entrada em níveis de cinza, $S_{m,n}$ o elemento estruturante com o tamanho de m x n, sendo m e n probabilidades maior que zero. Além disso, $\oplus$ indica uma operação de dilatação e $\ominus$ indica uma operação de erosão.

Para se obter o mapa binário morfológico, as operações morfológicas de fechamento (15) e abertura (16) são executadas utilizando um elemento estruturante $S_{3,3}$. A diferença (17) é obtida pela subtração de ambas as imagens que são os resultados da etapa seguinte. Então um procedimento de limiarização (18) é aplicado seguido por um processo de rotulação para extrair os segmentos de texto. O parâmetro T, no procedimento de limiarização, é definido dinamicamente de acordo com o fundo da imagem. Esse parâmetro é responsável por determinar o valor limite da operação de binarização.

A Figura 2.10 mostra todo o processo da técnica baseada na morfologia para se extrair as características de contraste. A Figura 2.11 (b) mostra o resultado desse processo.

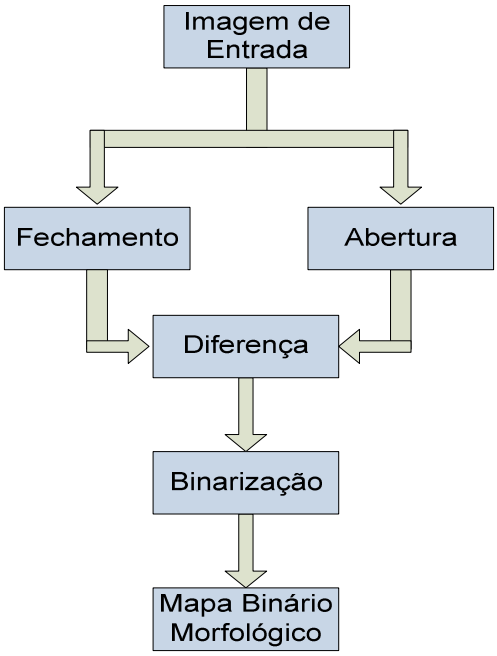


Figura 2.10 – Diagrama de blocos do método proposto para extrair as características de contraste [Pratheeba 2010].

Figura 2.11 – Geração do mapa binário morfológico, (a) imagem de entrada (b) mapa binário morfológico.

Utilizando uma operação morfológica de dilatação as regiões muito próximas podem ser facilmente conectadas enquanto as regiões mais distantes podem ficar isoladas. Pratheeba [Pratheeba, 2010] utilizou uma operação morfológica de dilatação com o elemento estruturante quadrado 7x7 na imagem binária obtida anteriormente (Figura 2.11(b)) para obter áreas comuns. A Figura 2.12(a) mostra o resultado do agrupamento das características. Se uma coluna de *pixels* consecutivos entre dois pontos diferentes de zeros na mesma linha é menor do que 5 % da largura da imagem, eles são preenchidos com 1s. Se os componentes conectados são menores que o valor do limiar, então são removidos. O valor do limiar é obtido empiricamente pela observação da região de tamanho mínimo do texto. Em seguida, cada componente conectado é redesenhado para ter as suas bordas suavizadas. Supondo que as regiões de texto geralmente são retangulares, uma caixa delimitadora retangular é gerada pela ligação de quatro pontos, que correspondem a (min_x,min_y), (max_x,min_y), (min_x,max_y), (max_x,max_y). As regiões candidatas já refinadas são mostradas na Figura 2.12(b).

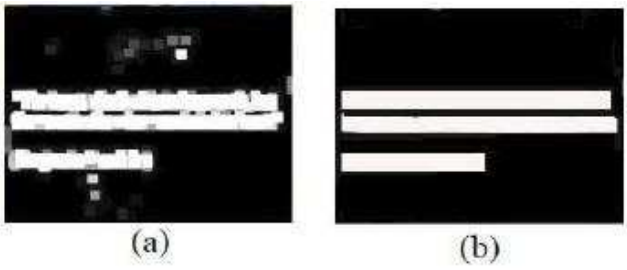


Figura 2.12 – Extração das regiões candidatas (a) componentes conectados através da dilatação (b) regiões candidatas suavizadas.

Baseado na observação da variação de intensidade que ao redor do *pixel* de transição é grande devido à complexa estrutura do texto, Pratheeba [Pratheeba, 2010] empregou DLBP (*Dominant Local Binary Pattern*) [Liao, 2009] para descrever a textura em torno do *pixel* de

transição. O DLBP efetivamente captura os padrões dominantes das texturas da imagem. Ao contrário da abordagem convencional LBP (*Local Binary Pattern*) [Ojala, 2002], no qual explora apenas de uma forma uniforme, dada uma textura da imagem. A abordagem da DLBP calcula a freqüência de ocorrência de todos os padrões invariantes definidos nos grupos de LBP. Estes padrões são, então, classificados em ordem decrescente. Os primeiros padrões com maior freqüência de ocorrência devem conter os padrões dominantes na imagem.

O LBP é uma ferramenta simples e muito eficiente para representar a consistência da textura utilizando somente o padrão da intensidade. O LBP forma um padrão binário utilizando o *pixel* corrente e todos os seus *pixels* vizinhos circulares e pode ser convertido para um número decimal. (Equação 19)

$$\mathrm{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i, \text{ sendo } s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad (19)$$

Sendo que, P e R indicam o número do *pixel* escolhido e o raio do círculo respectivamente, g_c e g_i indicam a intensidade do *pixel* corrente e os seus *pixels* vizinhos circulares.

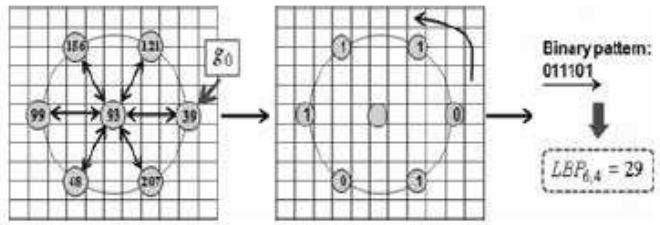


Figura 2.13 – Exemplo do calculo do LBP.

Obtem-se o padrão binário como mostrado na Figura 2.13 da seguinte forma: $\mathrm{LBP}_{6,4} = 29$ $(2^4+2^3+2^2+2^0)$.

O DLBP considera os padrões mais freqüentes que ocorreram em uma imagem. Percebe-se que a abordagem DLBP é mais confiável para representar a informação do padrão dominante nas imagens, evitando assim o problema de explorar apenas de uma forma uniforme.

Para se obter o valor DLBP é aplicada a operação LBP para cada *pixel* de transição em cada região candidata usando sua 8 vizinhança. Então é calculado o número de diferentes

DLBPs para considerar a variação da intensidade ao redor dos *pixels* de transição, definindo assim a probabilidade de ser texto. Essas informações são armazenadas em um vetor de características. Se a probabilidade de uma região candidata for maior, indica que um valor pré-definido à região correspondente é finalmente determinado como uma região de texto. O valor do limiar da probabilidade é definido empíricamente. A região de texto detectada é mostrada na Figura 2.14.

Figura 2.14 – Detecção das regiões candidatas.

A abordagem proposta por Pratheeba [Pratheeba, 2010] foi testada utilizando vídeos da vida real. Como não existe um banco de dados de vídeos padrão, foi criado um banco com 15 seqüências de vídeo MPEG-1 com uma resolução de 320 x 240 totalizando 5299 quadros. Todos os textos tiveram orientação horizontal. O banco de dados contém uma larga variedade de vídeos capturados de canais de televisão, incluindo comerciais e noticiário (nacional e estrangeiro). Uma larga variedade de fontes de texto, línguas e cores são representadas nos vídeos. A seqüência de vídeos foi capturada em 30 quadros por segundo

A fim de confirmar a eficiência do método na detecção e extração dos textos, a precisão é calculada utilizando a probabilidade do erro (PE) conforme a Equação (20).

$$PE = P(T)P(B/T) + P(B)T(T/B) \quad (20)$$

Sendo P(T) e P(B) indicam a probabilidade de serem *pixels* de texto e *pixels* do fundo nas imagens com fundo verdadeiro, respectivamente. P(B/T) indica a probabilidade de erro para classificar *pixels* de textos como *pixels* de fundo. P(T/B) indica a probabilidade de erro para classificar *pixels* de fundo como *pixels* de texto. A probabilidade média de erro PE (somatório das probabilidades de erro / quantidade de amostras) resultou em uma probabilidade de erro (PE) de 0, 0726.

Palaiahnakote em [Palaiahnakote, 2008], propôs a exploração de novas características de borda como retidão para a eliminação de bordas não significantes. Partes dos textos segmentados de um quadro são utilizadas para detectar a fronteira exata das linhas de texto nas imagens de vídeo. Para segmentar parte de um texto completo, o método introduz a seleção de um bloco de texto candidato de uma imagem.

O método proposto encontra blocos de texto candidato com base em algumas regras heurísticas. A projeção do perfil das bordas da imagem e a informação sobre o alinhamento do texto são usadas para detecção do bloco de texto com poucos falsos alarmes. Segundo Palaiahnakote [Palaiahnakote, 2008], o método é rápido e utiliza um mapa de bordas do quadro de vídeo para detectar o bloco de texto, sendo essa a grande vantagem proposta pelo método em relação aos métodos anteriores.

São duas regras para identificar o bloco de texto candidato, após dividir a imagem em 16 blocos de mesmo tamanho 64 x 64 *pixels* mostrados na Figura 2.15. Palaiahnakote [Palaiahnakote, 2008] escolheu o tamanho do bloco de 64 x 64 *pixels* porque espera-se que estes contenham parte texto. O método utiliza filtro aritmético e filtro da mediana para derivar suas regras. Esses filtros são conhecidos por remover ruídos da imagem (Figura 2.16).

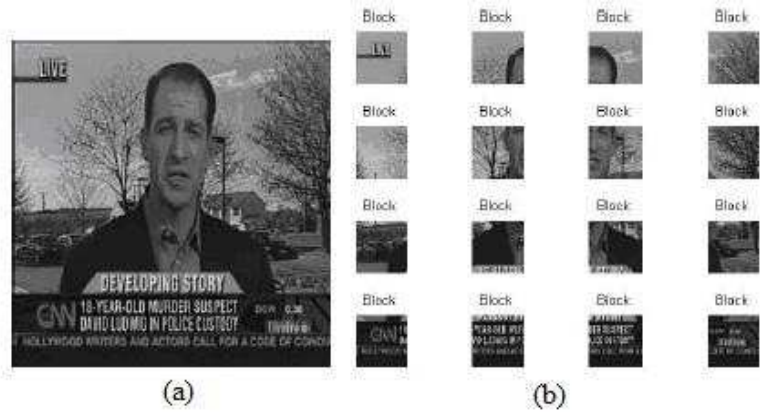


Figura 2.15 – (a) Imagem em níveis de cinza, (b) 16 blocos da imagem.

Figura 2.16 – (a) Imagem níveis de cinza, (b) imagem filtrada pela média e (c) imagem filtrada pela mediana.

As bordas são detectadas utilizando o detector de bordas de Canny, sendo que as bordas que contêm menos do que quatro *pixels* são eliminadas. Os resultados da detecção dos textos antes e depois da eliminação da borda são apresentados nas Figuras 2.17 (a) até (e). Sendo que (a) refere-se à parte segmentada, (b) e (c) são os resultados da detecção da borda por *Canny* antes a após a eliminação da borda respectivamente, (d) e (e) são os resultados da detecção do texto antes e após a eliminação da borda. A Figura 2.17(e) apresenta as linhas de textos que são propriamente detectadas com quatro caixas delimitadoras na imagem, enquanto a Figura 2.17(d) mostra as linhas de textos com duas grandes caixas delimitadoras. Portanto, a eliminação da borda ajuda a melhorar o desempenho do método. Os efeitos da eliminação podem ser vistos na Figura 2.17(c) em comparação com a Figura 2.17(b). No entanto, algumas vezes isso elimina os caracteres de texto quando estes estão ligados uns aos outros. Isso pode levar a falsos alarmes que pode ser notado na Figura 2.17 (e) onde alguns caracteres não são cobertos pela caixa delimitadora.

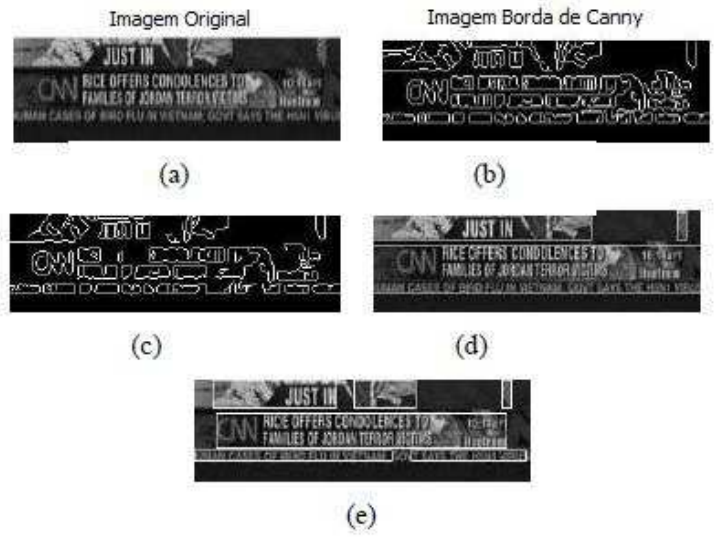


Figura 2.17 – Imagem com as caixas delimitadoras (*bounding box*) para as linhas de textos com e sem eliminação de bordas.

Em alguns casos, o método não consegue segmentar o texto completo parte por causa do fundo complexo ou pela presença de textos isolados. Por exemplo, a Figura 2.18 (a) e 2.18(c) refere-se à imagem original e 2.18(b) e 2.18(d) refere-se à segmentação incompleta. Já a Figura 2.19 mostra o resultado proposto pelo método.

Figura 2.18 – Falha no resultado da segmentação.

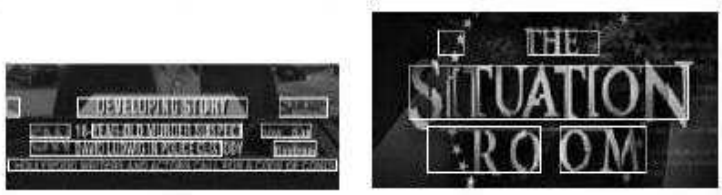


Figura 2.19 – Resultado do método proposto.

Palaiahnakote [Palaiahnakote, 2008] avaliou o desempenho do algoritmo de seleção de bloco de textos candidatos considerando a precisão como métrica. A precisão é definida como o número de imagens para cada qual o bloco de texto candidato foi corretamente escolhido dividido pelo número total de imagens. O método identifica bloco de textos candidatos com sucesso para 93 imagens de um total de 101. Portanto a precisão foi de 92 %. Em alguns casos, existe a necessidade de escolher dois blocos de textos candidatos quando a imagem contém textos em diferentes partes.

Os blocos de textos detectados são representados por suas caixas delimitadoras (*bounding boxes*). Para julgar se o bloco de texto detectado está correto, Palaiahnakote [Palaiahnakote, 2008] manualmente contou os blocos de texto verdadeiros que aparecem nas imagens do banco de dados. Também foi rotulado manualmente cada um dos blocos de texto detectados como uma das seguintes categorias:

- ✓ Bloco de texto detectado verdadeiramente: um bloco detectado que contém texto.
- ✓ Bloco de texto detectado falsamente: um bloco detectado que não contém texto.
- ✓ Bloco de texto com perda de dados: um bloco de texto detectado que não incluiu algum caractere.

- ✓ Bloco de texto com imprecisão no limite: um bloco de texto detectado verdadeiramente no qual o seu limite é mais largo que a caixa delimitadora do bloco de texto.

Baseado no número de blocos em cada uma das categorias mencionados acima, as métricas para avaliar o desempenho do método são calculadas da seguinte forma:

- ✓ Taxa de detecção = número de bloco de textos detectado verdadeiramente / número de blocos de textos existentes.
- ✓ Taxa de falso positivo = número de bloco de textos detectados falsamente / número de blocos de textos detectados.
- ✓ Taxa de dados perdidos = número de blocos de textos com perda de dados / número de blocos de textos detectados verdadeiramente.
- ✓ Taxa de imprecisão no limite = número de blocos de textos com imprecisão no limite / número de bloco de textos detectados verdadeiramente.

Os resultados obtidos pelo método proposto foram os seguintes: taxa de detecção 89,5%, taxa de falso positivo 10,6%, taxa de dados perdidos 17,1% e taxa de imprecisão no limite 10,4%.

Xiaoqing em [Xiaoqing, 2006] propôs um método de multi-escala baseado em bordas para extração de textos de imagem complexas. O método proposto é baseado no fato que as bordas são características confiáveis de texto independentemente da cor/intensidade, disposição e orientação. A proposta de Xiaoqing [Xiaoqing, 2006] consiste em três estágios, detecção da região de texto candidatos, localização da região de texto e extração de caracteres.

A detecção de regiões de texto candidatas, tem como objetivo construir um mapa de características utilizando três importantes propriedades da borda: aresta da borda, densidade e a variação da orientação. O mapa de características é uma imagem em escala de cinza com o mesmo tamanho da imagem de entrada onde a intensidade do *pixel* representa a possibilidade de texto. Xiaoqing [Xiaoqing, 2006] usou a magnitude da segunda derivada da intensidade como medida para a aresta da borda permitindo uma melhor detecção de picos de intensidade que normalmente caracteriza textos na imagem. A densidade da borda é calculada baseada na média da aresta da borda dentro de uma janela. Considerando a eficácia e a eficiência, quatro orientações (0°, 45°, 90°, 135°) são usadas para avaliar a variância da orientação, sendo 0°

indica direção horizontal, 90° indica a orientação vertical, 45° e 135° são duas direções diagonais respectivamente. A operação de convolução com um operador de bússola (mostrado na Figura 2.20) resulta em quatro imagens de intensidade de bordas orientadas E (θ), (θ $\in$ {0, 45, 90, 135}), as quais contêm todas as propriedades da bordas exigidas no método proposto.

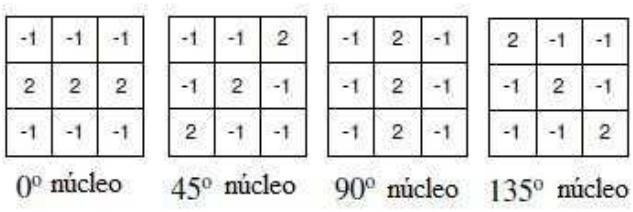


Figura 2.20 – Operador bússola.

O detector de borda é realizado utilizando a estratégia de multi-escala, onde imagens de multi-escala são produzidas por pirâmides Guassianas [Burt, 1981] e sucessivamente filtradas pelo filtro passa-baixa, reduzindo-as nas direções verticais e horizontais. Regiões com texto poderão ter valores significativamente mais altos para a média da densidade das bordas, resistência e variância das orientações do que as regiões de não textos. Xiaoqing [Xiaoqing, 2006] explorou essas três características para gerar um mapa que suprima as falsas regiões e melhore as verdadeiras regiões candidatas. Esse procedimento é descrito na Equação 20.

$$fmap(i,j) = \bigoplus_{s=0}^{n} \sum_{\theta} N\{ \sum_{x=-c}^{c} \sum_{y=-c}^{c} E(s,\theta, i+x, j+y) \times W(i,j)\} \qquad (20)$$

Sendo o *fmap* o mapa de característica de saída, $\oplus$ uma operação de adição em escala, *n* é o mais alto nível da escala, que é determinado pela resolução (tamanho) da imagem de entrada. Foram utilizadas duas escalas para imagens com a resolução 640 x 480. $\theta \in \{0, 45, 90, 135\}$ que estão em diferentes orientações e *N* é a operação de normalização. (*i,j*) são coordenadas do *pixel* da imagem. *W(i, j)* é a largura para o *pixel (i,j)*, cujo valor é determinado pelo numero de orientações das bordas dentro da janela. O tamanho da janela é determinado por uma constante *c*.

Normalmente, textos embutidos na imagem aparecem em grupos. Assim, as características de agrupamento podem ser usadas para localizar as regiões de textos. Uma vez que a intensidade do mapa de características representa a possibilidade de texto, um simples limiar global pode ser empregado para destacar aqueles com altas possibilidades de serem

regiões de texto, resultado em uma imagem binária. O operador morfológico de dilatação pode facilmente conectar regiões muito próximas deixando regiões distantes isoladas. O método proposto utilizou a operação morfológica de dilatação com o elemento estruturante quadrado 7 x 7 na imagem binária obtida anteriormente para unir áreas referidas como texto.

O objetivo do método é extrair caracteres binários das regiões de textos localizadas para que possam ser repassado diretamente para uma ferramenta de reconhecimento de caracteres (OCR). No método proposto, Xiaoqing [Xiaoqing, 2006] utilizou *pixels* de caracteres brancos uniformes em um fundo totalmente preto pelo uso da Equação 21.

$$T = \bigcup_{i=1,\ldots} \overline{|SUB_i|}_z \qquad (21)$$

Sendo *T* o texto extraído da imagem de saída binária, $\bigcup$ é uma operação de união, SUB_i são sub-imagens da imagem original, sendo que *i* indica o numero de sub-imagens. Sub-imagens são extraídas de acordo com as caixas delimitadoras na localização das regiões de textos. $\overline{|\cdot|}_z$ é o algoritmo de limiarização que segmenta as regiões de textos em caracteres branco em um fundo preto. Os resultados do método são mostrados na Figura 2.21.

Figura 2.21 – Imagens com diferentes tamanhos de fontes, (a) imagens originais e (b) imagens destacando apenas os textos.

Para avaliar o desempenho do método proposto Xiaoqing [Xiaoqing, 2006] utilizou 75 imagens de teste de quatro tipos: capa de livros, imagens de rótulos de objetos, placas de identificação e imagens ao ar livre. As imagens possuem diferentes tamanhos de fontes, cores, orientação, alinhamento e projeção de perspectiva sob diferentes condições de iluminação.

A taxa de precisão obtida por essa abordagem foi de 91,8 % de sucesso.

2.5. Considerações Finais.

Nesse capitulo foram apresentados alguns conceitos relacionados à localização de texto em imagens e vídeos, assim como os principais trabalhos que estão diretamente relacionados ao trabalho proposto de localização de códigos de identificação de vagões em trem.

Os resultados da extração dos textos das imagens de vídeos podem ser utilizados em vários tipos de aplicações. Devido à complexidade do problema cada autor apresenta uma abordagem diferente.

O que dificulta a avaliação dos métodos é o fato que cada autor possui o próprio banco de imagens até mesmo pelo fato de não existir um banco de imagens padrão.

Capítulo 3

Método Proposto

Este capítulo apresenta o método proposto para a localização de código de identificação de vagões de trem em cenas de vídeo. A abordagem proposta busca considerar o menor número de restrições possíveis e está dividida em quatro etapas, a saber: Pré-processamento, Segmentação, Detecção do Bloco de Texto e Pós-processamento.

No Pré-processamento o vídeo é segmentado em múltiplos quadros (imagens) e cada quadro é transformado de colorido para tons de cinza. Na etapa de Segmentação são propostas duas abordagens: na primeira abordagem a imagem é submetida a um único processo de limizarização, já na segunda, a imagem é submetida à três técnicas de limiarização cujos resultados são combinados.

Na etapa de Detecção de Bloco de Texto utiliza-se um filtro tipo passa-alta (Laplaciano), com uma máscara 3x3, normalizando-se o resultado para o intervalo [0,1]. A detecção do texto tem como base a diferença do máximo gradiente. Finalmente no Pós-processamento é empregado um filtro baseado no fator de compacidade dos componentes conexos encontrados da imagem. O fator de compacidade é a razão entre a área e o perímetro de um componente.

A Figura 3.1 apresenta um diagrama das etapas que são realizadas no processo de localização do código de identificação de vagões.

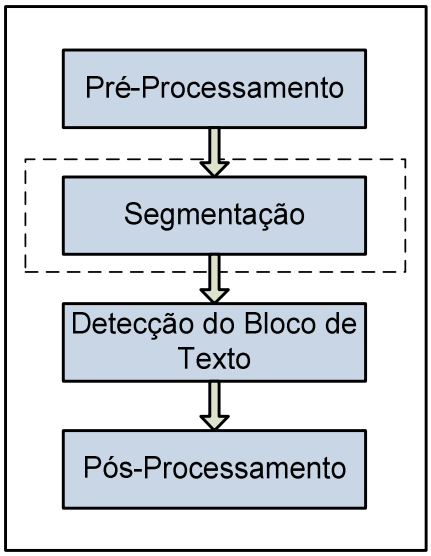


Figura 3.1 – Diagrama do método proposto.

3.1. Pré-processamento

O primeiro passo no Pré-processamento é separar o vídeo em múltiplos quadros (imagens), onde cada quadro representa uma imagem do vagão em movimento. A imagem é convertida para escala de cinza com o intuito de reduzir a quantidade de informação sem perder o código de interesse. As cores no modelo RGB são descritas pela indicação da quantidade de vermelho (*Red*), verde (*Green*) e azul (*Blue*) que contêm. Cada uma pode variar entre o mínimo 0 (completamente escuro) e máximo 255 (completamente intenso). Caso todos os canais estejam no mínimo, o resultado é preto. Caso estes estejam no máximo, o resultado é branco. A conversão em níveis de cinza foi feita pela média simples dos valores dos canais RGB da seguinte forma: (R + G + B) /3.

3.2. Segmentação

Na etapa de Segmentação são propostas duas abordagens: a primeira delas consiste em avaliar três técnicas de limiarização separadamente. A primeira técnica é a de multi-limiarização, proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994], a segunda técnica é a de limiarização local adaptativa proposta por Bernsen [Bernsen, 1995] e a terceira também se trata de uma técnica de limiarização local adaptativa sendo a limiarização de média móvel de Wellner [Parker, 1996]. A segunda abordagem proposta nessa etapa consiste em combinar os resultados das três técnicas de limiarização.

Na primeira abordagem os testes demonstraram que os melhores resultados foram obtidos com os seguintes parâmetros: 2 (dois) níveis para a multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994], 35 (trinta e cinco) para o contraste na limiarização local adaptativa de Bernsen [Bernsen, 1995] e 5 % (cinco) para a porcentagem de média móvel de Wellner [Parker, 1996].

A Figura 3.2 apresenta um exemplo de uma imagem de vagão após o processo de multi-limiarização N.Papamarkos e B. Gatos [Papamarkos, 1994] utilizando 2 (dois) níveis.

Figura 3.2 – Imagem após o processo de multi-limiarização proposta N.Papamarkos e B. Gatos [Papamarkos, 1994] utilizando 2 (dois) níveis.

A Figura 3.3 apresenta um exemplo de uma imagem após o processo de limiarização local adaptativa proposta por Bernsen [Bernsen, 1995] utilizando o parâmetro 35 para o contraste.

Figura 3.3 – Imagem após o processo de limiarização local adaptativa proposta Bernsen [Bernsen, 1995] utilizando 35 (trinta e cinco) para o contraste.

A Figura 3.4 apresenta um exemplo de uma imagem após o processo de limiarização local adaptativa utilizando a porcentagem de média móvel de Wellner [Paker, 1996] utilizando 5 (cinco) % para o parâmetro.

Figura 3.4 – Imagem após o processo de limiarização local adaptativa utilizando a porcentagem de média móvel de Wellner [Paker, 1996] utilizando 5 (cinco) % de parâmetro.

Conforme já descrito, a segunda abordagem proposta para a fase de Segmentação consiste em: utilizar as três técnicas de limiarização em conjunto detectando os contornos

produzidos pelas imagens limiarizadas. A idéia principal dessa abordagem é resolver o problema que ocorre quando o processo de limiarização perde total ou parcialmente o código do vagão.

A Figura 3.5 apresenta o diagrama da segunda abordagem da etapa de Segmentação

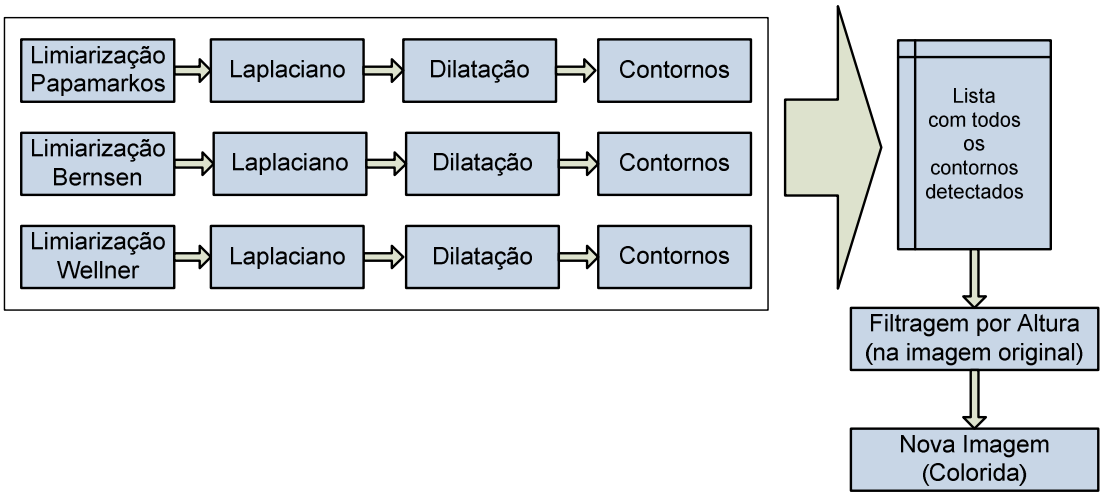


Figura 3.5 – Diagrama da segunda abordagem da etapa de Segmentação.

Após realizar uma análise detalhada de várias imagens, detectou-se que, em grande parte, as falhas na localização estão relacionadas ao processo de limiarização que em determinados casos perde total ou parcialmente o código do vagão. Espera-se que a combinação dos resultados de três abordagens diferentes de limiarização minimize este problema. Observa-se na Figura 3.5 que os contornos derivados de cada imagem limiarizada são combinados em uma lista de contornos que sofre, posteriormente, um processo de filtragem.

A região de texto geralmente tem um grande número de descontinuidades (transições entre o texto e o fundo). Para detectar essas descontinuidades cada uma das três imagens limiarizada é filtrada utilizando uma máscara 3x3 do Laplaciano [Trung, 2009]. Essa máscara produz dois valores para cada borda, positivos e negativos. Para eliminar os valores negativos da imagem filtrada por Laplaciano é necessário que a imagem seja normalizada para o intervalo [0, 1].

Sob as imagens do filtro Laplaciano normalizadas é aplicada uma operação morfológica de dilatação de 1 (uma) interação com o elemento estruturante quadrado 3x3. A dilatação das imagens é realizada com o intuito de conectar elementos a fim de detectar os contornos.

A Figura 3.6 apresenta o resultado da imagem dilatada a partir do filtro Laplaciano de uma imagem que foi limiarizada pelo processo de multi-limiarização N.Papamarkos e B. Gatos [Papamarkos, 1994].

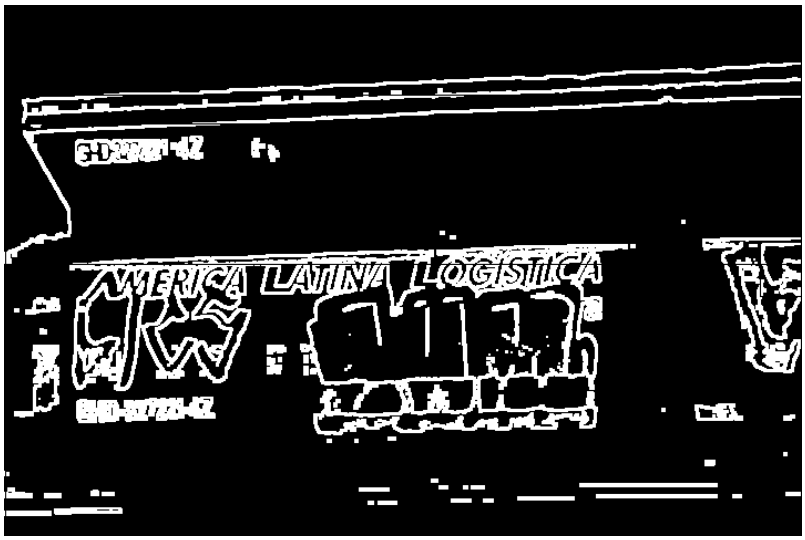


Figura 3.6 – Imagem após a operação morfológica de dilatação em uma imagem filtrada anteriormente por Laplaciano.

Na Figura 3.6, embora o código do vagão apareça borrado, a região do mesmo será detectada no processo de detecção de contornos.

A Figura 3.7 apresenta os contornos detectados a partir da imagem na Figura 3.6.

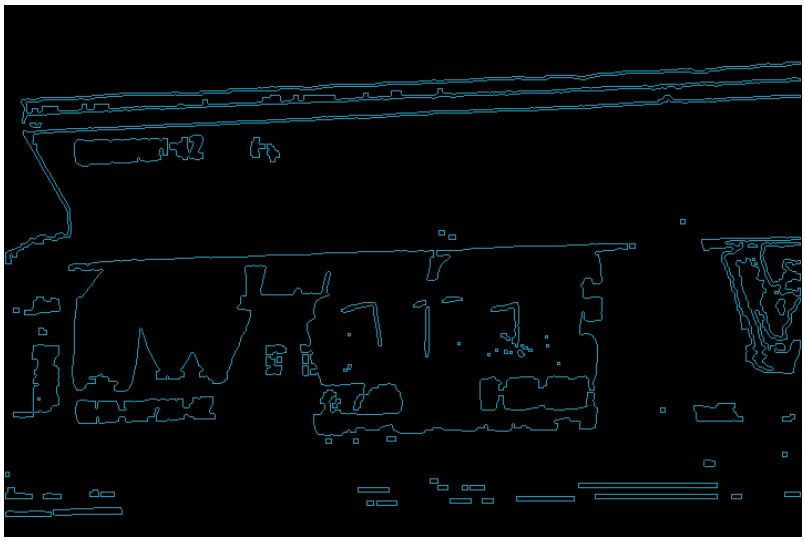


Figura 3.7 – Imagem dos contornos detectados.

Percebe-se na Figura 3.7 a região do código do vagão localizada em conjunto com os outros contornos da imagem dilatada.

A Figura 3.8 apresenta o resultado da imagem dilatada a partir do filtro Laplaciano de uma imagem que foi limiarizada pelo processo de limiarização local adaptativa proposto por Bernsen [Bernsen, 1995].

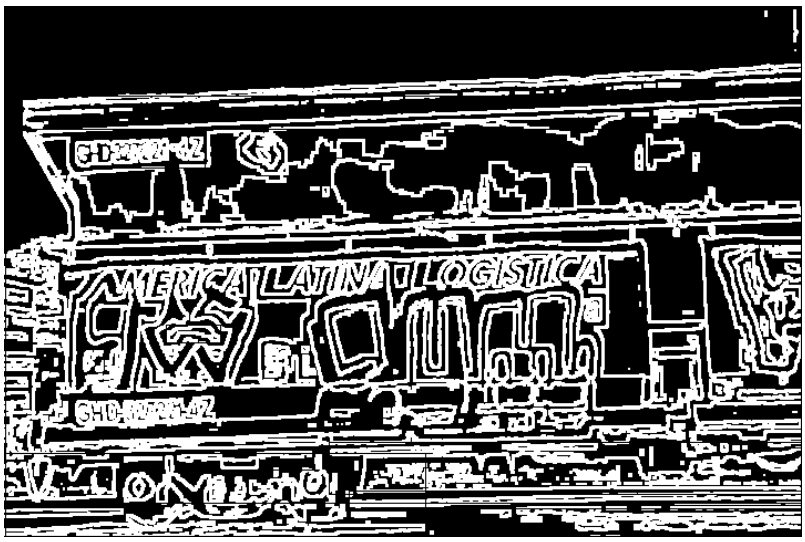


Figura 3.8 – Imagem após a operação morfológica de dilatação em uma imagem filtrada anteriormente por Laplaciano.

A Figura 3.9 apresenta os contornos detectados a partir da imagem apresentada na Figura 3.8.

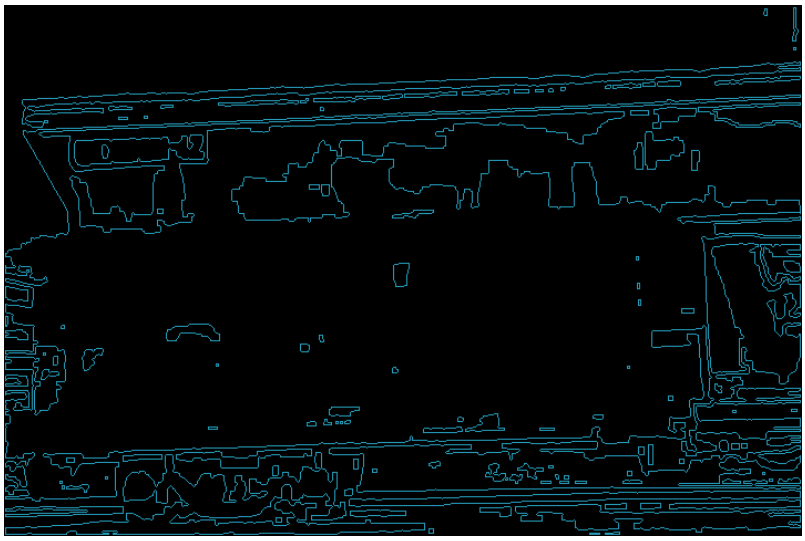


Figura 3.9 – Imagem dos contornos detectados.

A Figura 3.10 apresenta o resultado da imagem dilatada a partir do filtro Laplaciano de uma imagem que foi limiarizada utilizando a porcentagem de média móvel de Wellner [Paker, 1996].

Figura 3.10 – Imagem após a operação morfológica de dilatação em uma imagem filtrada anteriormente por Laplaciano.

A Figura 3.11 apresenta os contornos detectados a partir da imagem na Figura 3.10.

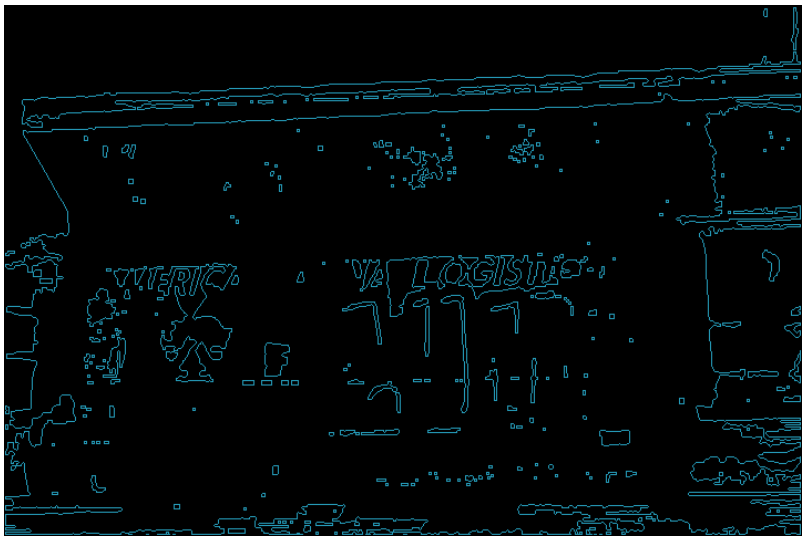


Figura 3.11 – Imagem dos contornos detectados.

Percebe-se que a Figura 3.11 não apresenta o contorno do código do vagão. Isso ocorre pelo fato que na dilatação (Figura 3.10) a região do código conecta-se com uma outra região.

As informações dos contornos das imagens são armazenadas em uma lista única. De posse de todas as informações dos contornos detectados é realizada uma filtragem para eliminar as partes da imagem que não interessam no processo de detecção dos códigos de identificação dos vagões. O intuito principal dessa filtragem é eliminar os contornos com altura muito pequena ou muito grande não representando assim regiões com códigos de identificação de vagões.

Embora os códigos dos vagões possam apresentar fontes de diferentes tamanhos, estes sempre se apresentam na orientação horizontal. De posse dessa informação é realizada a filtragem dos contornos pela altura. As alturas menores que 10 e maiores que 50 *pixels* serão descartados. Esses valores foram obtidos através de experimentos realizados. Observou-se que em várias amostras (vagões diferentes), os códigos de identificação dos vagões possuíam uma altura entre 10 e 50 *pixels*.

A filtragem dos contornos é realizada na imagem original (imagem de entrada) gerando uma nova imagem colorida com o fundo preto.

A Figura 3.12 apresenta uma nova imagem colorida gerada sem as partes referentes aos contornos eliminados. Mantendo, em uma primeira filtragem, apenas os candidatos a serem códigos de vagões.

Figura 3.12 – Imagem gerada após os filtros de altura.

3.3. Detecção do Bloco de Texto

Conforme descrito anteriormente, regiões de textos podem possuir um grande número de descontinuidades entre a imagem e o fundo (transições entre o texto e o fundo). Para detectar essas descontinuidades a imagem é filtrada pelo filtro passa-alta do Laplaciano utilizando uma máscara 3x3[Trung, 2009], apresentada na Figura 3.13.

0	1	0
1	-4	1
0	1	0

Figura 3.13 – Máscara do Laplaciano utilizado no método proposto.

A máscara produz dois valores para cada borda. A imagem filtrada pelo Laplaciano contém valores positivos e negativos, sendo que as transições entre os valores (cruzamento entre os zeros) correspondem às transições entre o texto e o fundo. Para eliminar os valores negativos da imagem filtrada por Laplaciano é necessário que a imagem seja normalizada para o intervalo [0, 1].

A fim de detectar os blocos de textos é utilizada a diferença do máximo gradiente (MGD - *Maximum Gradient Difference*) entre os valores máximos e mínimos dentro de janelas de tamanho 1 x N [Wong, 2003]. O valor de MGD no *pixel* (i,j) é calculado a partir da imagem normalizada filtrada por Laplaciano f. A Equação 22 mostra a fórmula do MGD:

$$MGD(i, j) = \max(f(i, j-t)) - \min(f(i, j-t)) \qquad (22)$$

sendo $t \in \left[-\frac{N-1}{2}, \frac{N-1}{2}\right]$.

O mapa MGD é obtido pelo movimento da janela sobre a imagem [Trung, 2009]. O tamanho da janela sugerida pelo método é 11.

Na etapa anterior foram apresentadas duas abordagens de Segmentação, nessa etapa de Detecção do Bloco de Texto são apresentados exemplos referentes às abordagens acima. As Figuras 3.14 e 3.15 dizem respeito à primeira abordagem adotada na etapa de Segmentação e as Figuras 3.16 e 3.17 dizem respeito à segunda abordagem adotada na etapa de Segmentação.

A Figura 3.14 apresenta a imagem filtrada pelo filtro passa-alta do Laplaciano normalizado em uma imagem de vagão multi-limiarizada pela técnica N.Papamarkos e B. Gatos [Papamarkos, 1994]

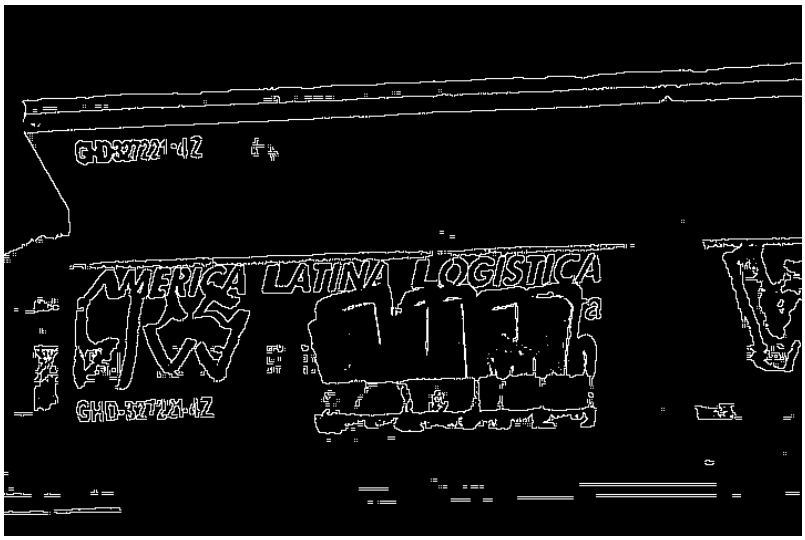


Figura 3.14 – Imagem filtrada por Laplaciano em uma imagem multi-limiarizada por N.Papamarkos e B. Gatos [Papamarkos, 1994].

A Figura 3.15 apresenta uma imagem após o processo de MGD da imagem filtrada por Laplaciano (Figura 3.14).

Figura 3.15 – Imagem MGD da imagem filtrada por Laplaciano.

A Figura 3.16 apresenta a imagem filtrada pelo filtro passa-alta do Laplaciano normalizado em uma imagem gerada a partir do filtro de seleção por altura (segunda abordagem da etapa de Segmentação).

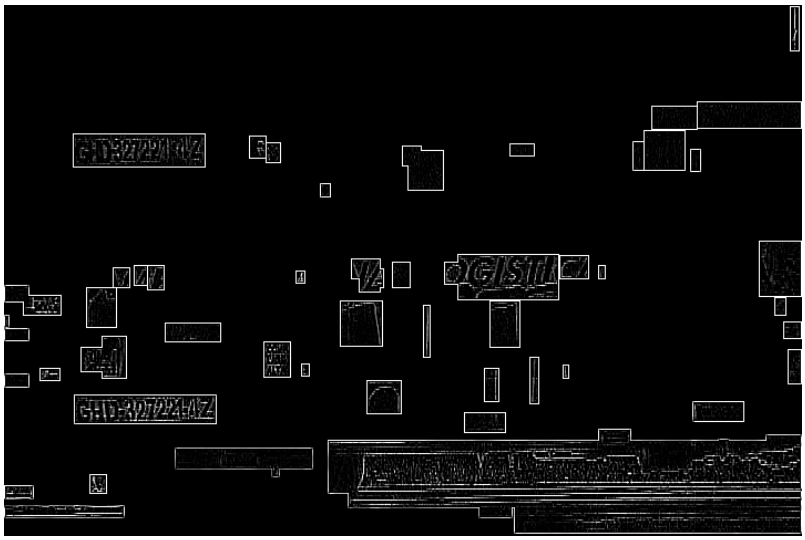


Figura 3.16 – Imagem filtrada por Laplaciano na imagem gerada a partir do filtro por altura.

A Figura 3.17 apresenta uma imagem após o processo de MGD da imagem filtrada por Laplaciano (Figura 3.16)

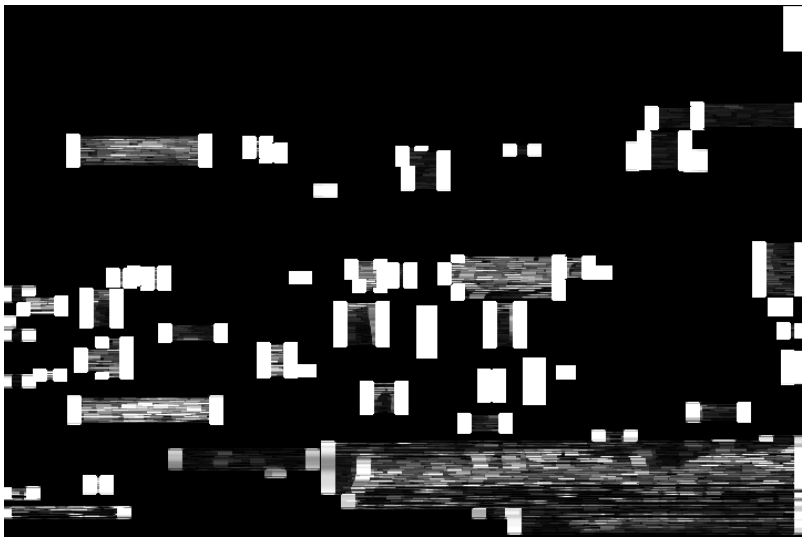


Figura 3.17 – Imagem MGD da imagem filtrada por Laplaciano a partir da seleção por altura.

3.4. Pós-processamento

A etapa de Pós-processamento consiste em empregar um filtro baseado no fator de compacidade dos componentes conexos encontrados da imagem. O fator de compacidade é a razão entre área (*A*) e o perímetro (*P*) de um conjunto conforme a Equação 23:

$$fc = \frac{2\pi A}{P^2} \tag{23}$$

No passo anterior, são detectados os blocos candidatos a serem códigos de vagões. Para cada bloco candidato é encontrado o seu contorno. De posse do contorno é calculado o fator de compacidade do bloco de texto.

Experimentos demonstram que os melhores resultados obtidos foram utilizando o intervalo de fator de compacidade entre 0,06 e 0,25. Caso o bloco de texto esteja entre esse intervalo torna-se um candidato a código de vagão.

A Figura 3.18 apresenta os contornos da imagem referente à primeira abordagem adotada na etapa de Segmentação.

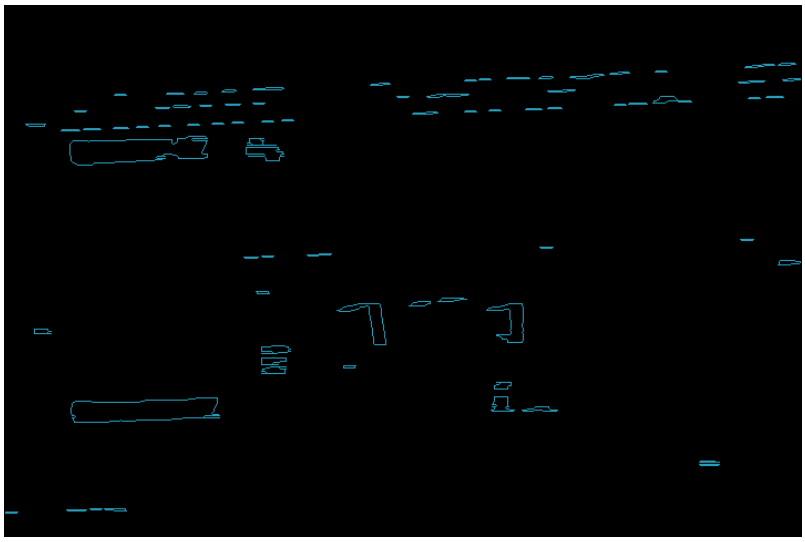


Figura 3.18 – Contornos da imagem filtrada pelo fator de compacidade referente à primeira abordagem adotada na etapa de Segmentação.

A Figura 3.19 apresenta os contornos da imagem referente à segunda abordagem adotada na etapa de Segmentação.

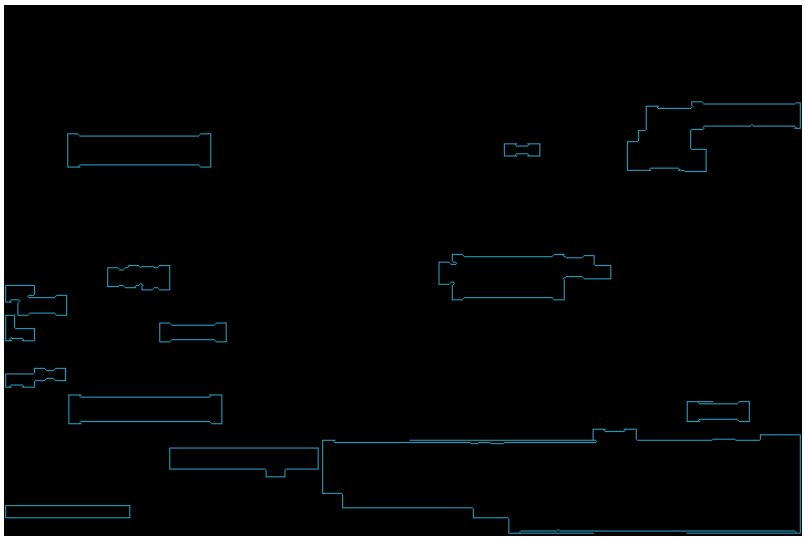


Figura 3.19 – Contornos da imagem filtrada pelo fator de compacidade referente à segunda abordagem adotada na etapa de Segmentação.

Ainda na etapa de Pós-processamento, os contornos filtrados pelo fator de compacidade são transformados em retângulos. Os retângulos contêm o posicionamento dos blocos de textos, nesse caso os possíveis códigos de identificação dos vagões.

Uma última filtragem é realizada utilizando propriedades geométricas. Retângulos que contenham uma largura menor que 65 *pixels* e maior que 200 *pixels* ou retângulos com altura maior a 50 *pixels* e menor que 15 *pixels* são descartados. Ainda nessa linha caso a altura do retângulo seja maior que a largura do mesmo esse retângulo também é descartado. Esses valores foram obtidos empiricamente nos experimentos realizados.

A Figura 3.20 apresenta os retângulos informando os possíveis códigos de identificação dos vagões, imagem referente à primeira abordagem adotada na etapa de Segmentação.

Figura 3.20 – Imagem com os candidatos a código de identificação dos vagões referente à primeira abordagem adotada na etapa de Segmentação.

A Figura 3.21 apresenta os retângulos informando os possíveis códigos de identificação dos vagões, imagem referente à segunda abordagem adotada na etapa de Segmentação.

Figura 3.21 – Imagem com os candidatos a código de identificação dos vagões referente à segunda abordagem adotada na etapa de Segmentação.

3.5. Considerações Finais

Nesse capítulo foi apresentada a descrição do método proposto, a qual visa realizar a localização dos textos em vídeo, em particular, de códigos de identificação de vagões.

O sistema foi desenvolvido na plataforma *Windows* utilizando a ferramenta de desenvolvimento *Visual Studio 2008* com a linguagem de programação C++. Também foi utilizada a biblioteca *Open Source Computer Vision (OpenCV)* que possui um conjunto de funções de manipulação de imagens.

O próximo capítulo traz a descrição dos experimentos realizados bem como os resultados alcançados com o método proposto.

Capítulo 4

Experimentos Realizados

Os experimentos apresentados neste capítulo visam avaliar o método proposto para a localização de códigos de vagões de diferentes modelos presentes em vídeos digitais. Ao todo foram realizados cinco experimentos. Os quatro primeiros avaliam as diferentes estratégias de segmentação previstas no capítulo anterior, a saber: a) uso da multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994]; b) uso da limiarização local adaptativa proposta por Bernsen [Bernsen, 1995]; c) uso da porcentagem de média móvel de Wellner [Parker, 1996]; e por fim, d) o uso da combinação dos contornos obtida a partir das imagens geradas nas três limiarizações citadas.

Nestes quatro primeiros experimentos, utilizou-se para avaliação do método 116 imagens (vagões) extraídas a partir de vídeos de trem filmados diretamente nas estações. Estas imagens correspondem a 116 vagões com diferentes códigos. Todas essas imagens têm a garantia que o código completo do vagão (série do vagão + código do vagão + dígito verificador) esteja na imagem. As imagens têm a resolução de 720 x 480 *pixels*.

O quinto experimento consiste em aplicar o método proposto utilizando a segunda abordagem da etapa de segmentação (utilizando os contornos das imagens limiarizadas em conjunto) diretamente no vídeo processando todos os seus quadros. Neste experimento considera-se que a localização é correta se ao menos uma vez o código de determinado vagão foi detectado nos múltiplos quadros onde este aparece. Considera-se correta a localização que permita recuperar ao menos o código do vagão sem a série.

4.1. Uso da multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994]

Nessa seção é apresentado o resultado obtido com o uso, da multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994]. Para cada etapa do processo será apresentada a imagem resultante.

A Figura 4.1 apresenta um dos vagões utilizados no processo de localização do código de identificação.

Figura 4.1 – Imagem original do vagão tanque para ser localizado o código de identificação.

A etapa de Pré-processamento consiste em transformar a imagem em níveis de cinza. A Figura 4.2 apresenta a imagem do vagão tanque em níveis de cinza.

Figura 4.2 – Imagem do vagão tanque em níveis de cinza.

Na etapa de segmentação, o processo consiste em multi-limiarizar a imagem utilizando a técnica proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994] utilizando 2 níveis. A Figura 4.3 apresenta a imagem multi-limiarizada.

Figura 4.3 – Imagem do vagão tanque multi-limiarizada por N.Papamarkos e B. Gatos [Papamarkos, 1994].

A próxima etapa a ser seguida é a etapa de Detecção do Bloco de Texto. Essa etapa consiste em aplicar o filtro do Laplaciano na imagem multi-limiarizada e aplicar o processo de MGD *Maximum Gradient Difference*.

A Figura 4.4 apresenta a imagem após o filtro do Laplaciano normalizado para [0, 1].

Figura 4.4 – Imagem do vagão tanque após o filtro por Laplaciano (normalizado).

Na seqüência é aplicado o processo de MGD *Maximum Gradient Difference.* A Figura 4.5 apresenta a imagem após esse processo.

Figura 4.5 – Imagem do MDG *Maximum Gradient Difference* aplicado ao vagão tanque.

A etapa de Pós-processamento consiste em detectar os contornos e realizar o filtro pelo fator de compacidade. A Figura 4.6 apresenta os contornos filtrados pelo fator de compacidade.

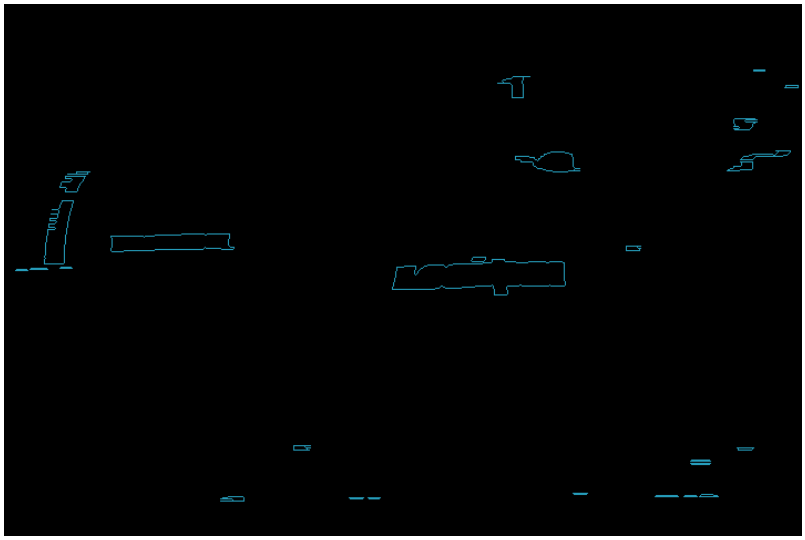


Figura 4.6 – Imagem dos contornos filtrados pelo fator de compacidade do vagão tanque.

Ainda na etapa de Pós-processamento, o último passo a ser dado consiste em apresentar a partir dos contornos filtrados pelo fator de compacidade os retângulos da localização. Como comentado anteriormente, uma última filtragem é realizada utilizando propriedades geométricas. Retângulos que contenham uma largura menor que 65 *pixels* e maior que 200 *pixels* ou retângulos com altura maior a 50 *pixels* e menor que 15 *pixels* são descartados. Ainda nessa linha caso a altura do retângulo seja maior que a largura do mesmo esse retângulo também é descartado.

A Figura 4.7 apresenta o resultado final método proposto.

Figura 4.7 – Resultado do método proposto utilizando a multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994].

Os melhores resultados obtidos para essa abordagem são apresentados na Tabela 4.1.

Parâmetros utilizados			
Tamanho da Janela MGD	**11**		
Multi-limiarização Papamarkos	**2**		
Fator Compacidade	**0,06**	a	**0,25**
Tipo dos Vagões	Graneleiro + Tanque + Plataforma		
Resultados			
Total de amostras	116		
Acertos	**77**	**66,38%**	
Erros	**39**	**33,62%**	
	116	100,00%	

Tabela 4.1 – Resultado da abordagem proposta utilizando multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994].

4.2. Uso da limiarização local adaptativa proposta por Bernsen [Bernsen, 1995]

Nessa seção é apresentado o resultado obtido do método proposto, em uma imagem do vagão utilizando na etapa de Segmentação a limiarização local adaptativa proposta por Bernsen [Bernsen, 1995] Para cada etapa do processo será apresentada a imagem resultante.

A Figura 4.8 apresenta o vagão para o qual o método proposto executa o processo de localização do código de identificação do vagão.

Figura 4.8 – Imagem original do vagão graneleiro para ser localizado o código de identificação.

A etapa de Pré-processamento consiste em transformar a imagem em níveis de cinza. A Figura 4.9 apresenta a imagem do vagão em níveis de cinza.

Figura 4.9 – Imagem do vagão graneleiro em níveis de cinza.

Na etapa de segmentação, o processo consiste em limiarizar a imagem utilizando a técnica de limiarização local adaptativa proposta por Bernsen [Bernsen, 1995] utilizando o valor de 35 para o contraste. A Figura 4.10 apresenta a imagem limiarizada por Bernsen [Bernsen, 1995].

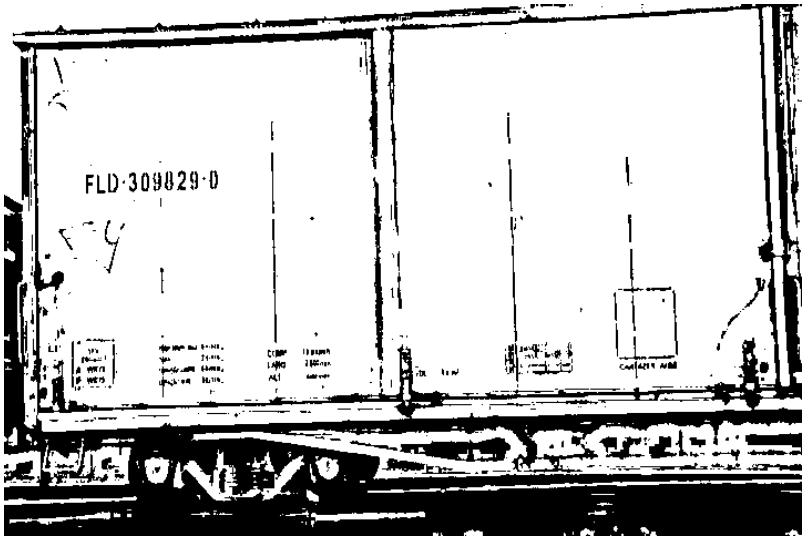


Figura 4.10 – Imagem do vagão graneleiro limiarizada por Bernsen [Bernsen, 1995].

A próxima etapa a ser seguida é a etapa de Detecção do Bloco de Texto. Essa etapa consiste em aplicar o filtro do Laplaciano na imagem limiarizada e aplicar o processo de MGD *Maximum Gradient Difference*.

A Figura 4.11 apresenta a imagem após o filtro do Laplaciano normalizado para [0, 1].

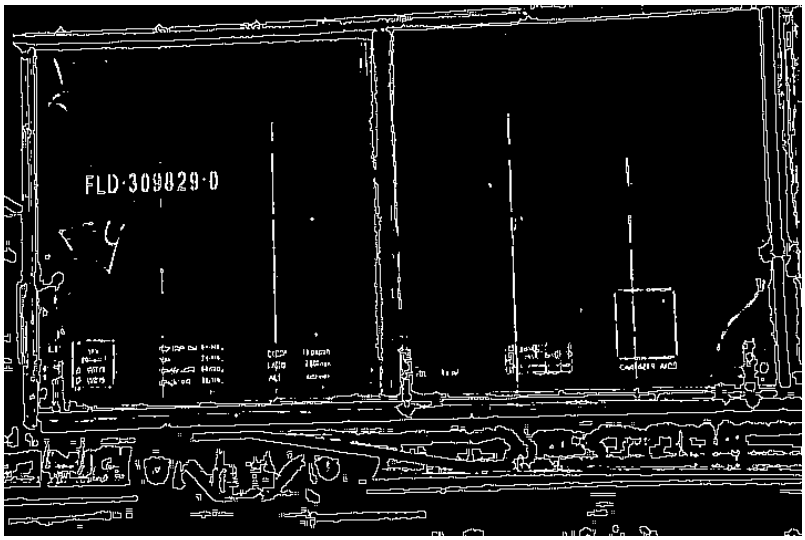


Figura 4.11 – Imagem do vagão graneleiro após o filtro por Laplaciano (normalizado).

Na seqüência é aplicado o processo de MGD *Maximum Gradient Difference.* A Figura 4.12 apresenta a imagem após esse processo.

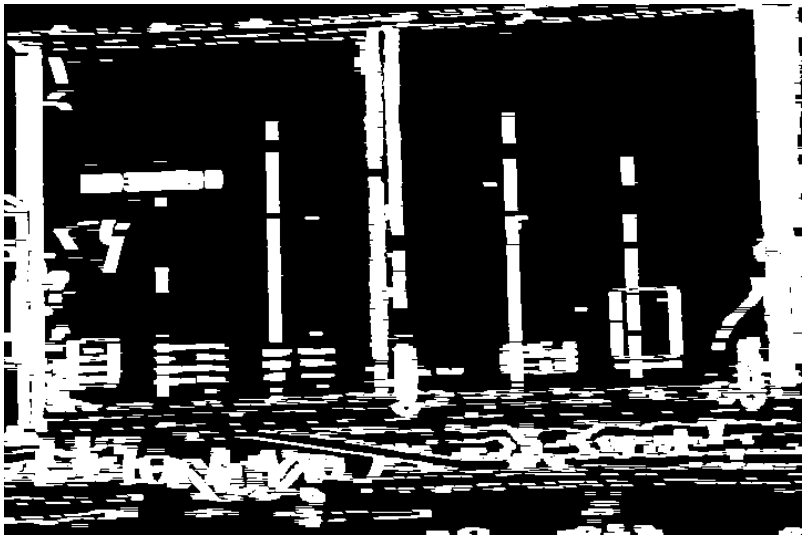


Figura 4.12 – Imagem do MDG *Maximum Gradient Difference* aplicado ao vagão graneleiro.

A etapa de Pós-processamento consiste em detectar os contornos e realizar o filtro pelo fator de compacidade. A Figura 4.13 apresenta os contornos filtrados pelo fator de compacidade.

.

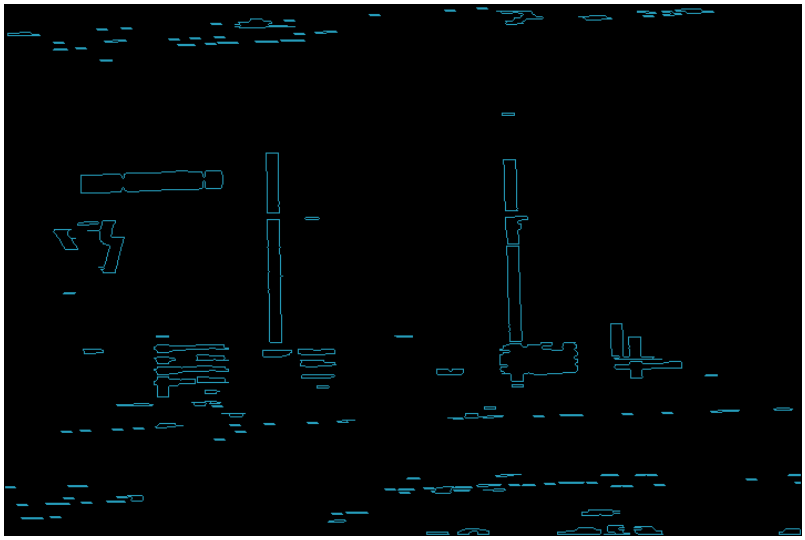


Figura 4.13 – Imagem dos contornos filtrados pelo fator de compacidade do vagão graneleiro.

Ainda na etapa de Pós-processamento, o último passo a ser dado consiste em apresentar a partir dos contornos filtrados pelo fator de compacidade os retângulos da localização. Como comentado anteriormente, uma última filtragem é realizada utilizando as propriedades geométricas.

A Figura 4.14 apresenta o resultado final do método proposto.

Figura 4.14 – Resultado do método proposto utilizando a limiarização local adaptativa proposta por Bernsen [Bernsen, 1995] .

Os melhores resultados obtidos para essa abordagem são apresentados na Tabela 4.2.

Parâmetros utilizados			
Tamanho da Janela MGD	**11**		
Limiarização Bernsen	**35**		
Fator Compacidade	**0,06**	a	**0,25**
Tipo dos Vagões	Graneleiro + Tanque + Plataforma		
Resultados			
Total de amostras	116		
Acertos	**27**	**23,28%**	
Erros	**89**	**76,72%**	
	116	100,00%	

Tabela 4.2 – Resultado da abordagem proposta utilizando multilimiariação proposta por Bernsen [Bernsen, 1995].

4.3. Uso da porcentagem da média móvel de Wellner [Paker, 1996]

Nessa seção é apresentado o resultado obtido do método proposto, em uma imagem do vagão utilizando na etapa de Segmentação a porcentagem de média móvel de Wellner [Paker, 1996]. Para cada etapa do processo será apresentada a imagem resultante.

A Figura 4.15 apresenta o vagão para o qual o método proposto executa o processo de localização do código de identificação do vagão.

Figura 4.15 – Imagem original do vagão graneleiro para ser localizado o código de identificação.

A etapa de Pré-processamento consiste em transformar a imagem em níveis de cinza. A Figura 4.16 apresenta a imagem do vagão em níveis de cinza.

Figura 4.16 – Imagem do vagão graneleiro em níveis de cinza.

Na etapa de segmentação, o processo consiste em limiarizar a imagem utilizando a técnica de limiarização pelo porcentual de média móvel de Wellner [Parker, 1996] utilizando 5 (cinco) % de parâmetro.

A Figura 4.17 apresenta a imagem limiarizada pela porcentagem da média móvel de Wellner [Paker, 1996].

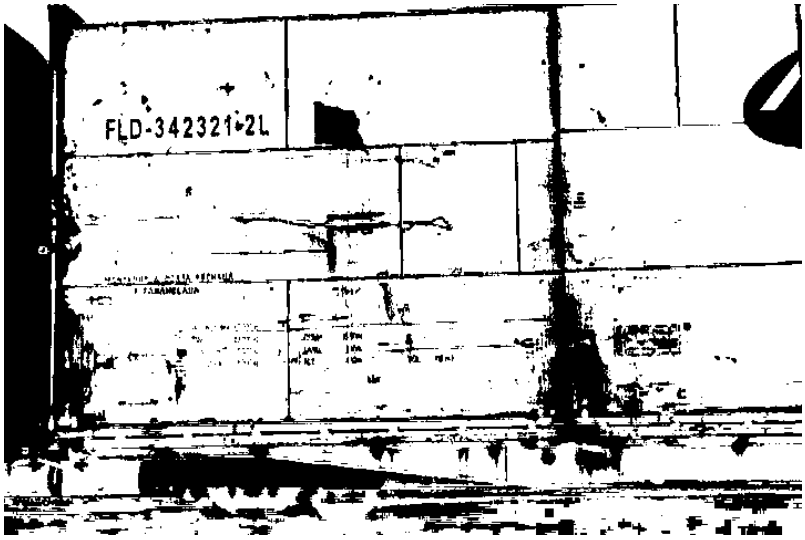


Figura 4.17 – Imagem do vagão graneleiro limiarizada pela porcentagem da média móvel de Wellner [Paker, 1996].

A próxima etapa a ser seguida é a etapa de Detecção do Bloco de Texto. Essa etapa consiste em aplicar o filtro do Laplaciano na imagem limiarizada e aplicar o processo de MGD *Maximum Gradient Difference*.

A Figura 4.18 apresenta a imagem após o filtro do Laplaciano normalizado para [0, 1].

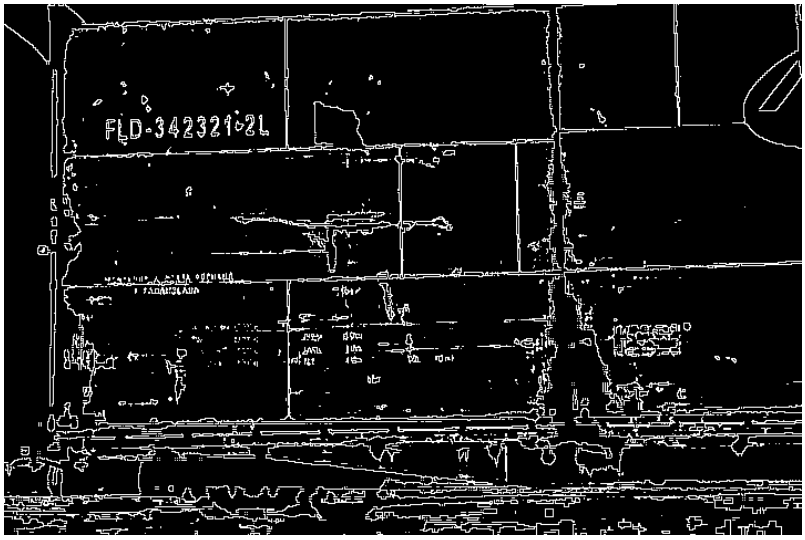


Figura 4.18 – Imagem do vagão graneleiro após o filtro por Laplaciano (normalizado).

Na seqüência é aplicado o processo de MGD *Maximum Gradient Difference.* A Figura 4.19 apresenta a imagem após esse processo.

Figura 4.19 – Imagem do MDG *Maximum Gradient Difference* aplicado ao vagão graneleiro.

A etapa de Pós-processamento consiste em detectar os contornos e realizar o filtro pelo fator de compacidade. A Figura 4.20 apresenta os contornos filtrados pelo fator de compacidade.

Figura 4.20 – Imagem dos contornos filtrados pelo fator de compacidade do vagão graneleiro.

A Figura 4.21 apresenta o resultado final após a última filtragem do Pós-processamento.

Figura 4.21 – Resultado do método proposto utilizando a limiarização por média móvel de Wellner [Paker, 1996].

Os melhores resultados obtidos para essa abordagem são apresentados na Tabela 4.3.

Parâmetros utilizados			
Tamanho da Janela MGD	**11**		
Porcentagem da média móvel de Wellner	**5**		
Fator Compacidade	**0,06**	a	**0,25**
Tipo dos Vagões	Graneleiro + Tanque + Plataforma		
Resultados			
Total de amostras	116		
Acertos	**25**	**21,55%**	
Erros	**91**	**78,45%**	
	116	100,00%	

Tabela 4.3 – Resultado da abordagem proposta utilizando a porcentagem da média móvel de Wellner [Paker, 1996].

4.4. Uso da combinação dos contornos das imagens limiarizadas

Nessa seção é apresentado o resultado obtido para uma imagem de vagão utilizando na etapa de Segmentação a combinação dos contornos das imagens limiarizadas pelas três técnicas de segmentação.

A Figura 4.22 apresenta o vagão para o qual o método proposto executa o processo de localização do código de identificação do vagão.

Figura 4.22 – Imagem original do vagão graneleiro para ser localizado o código de identificação.

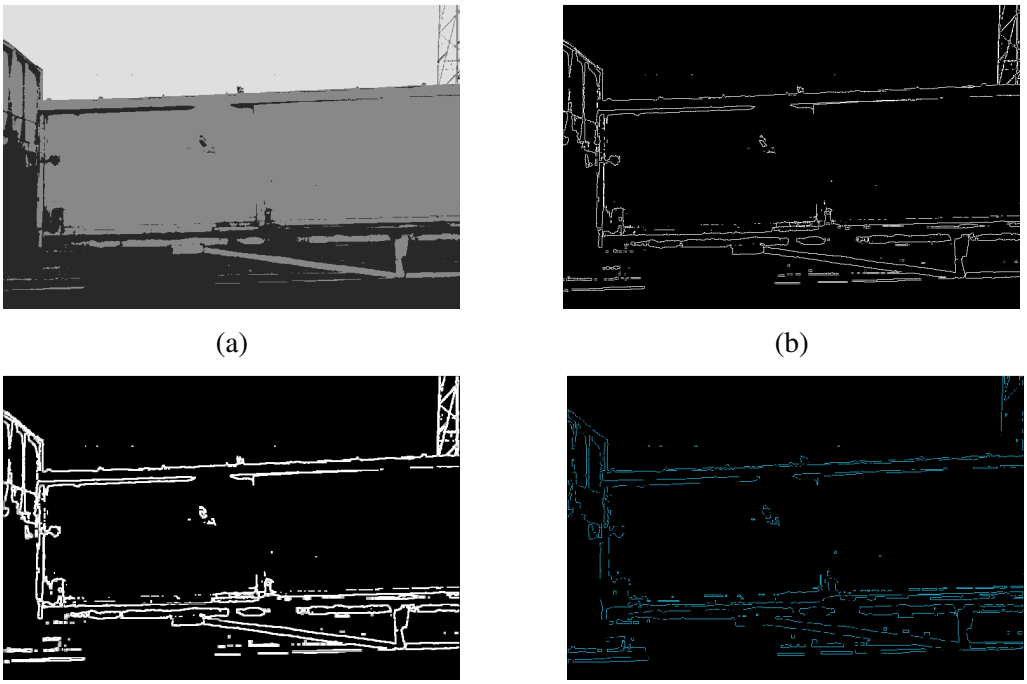
A etapa de Pré-processamento consiste em transformar a imagem em níveis de cinza. A Figura 4.23 apresenta a imagem do vagão tanque em níveis de cinza.

Figura 4.23 – Imagem do vagão graneleiro em níveis de cinza

Neste experimento a Segmentação consiste em limiarizar a imagem utilizando as técnicas propostas por N.Papamarkos e B. Gatos [Papamarkos, 1994], Bernsen [Bernsen, 1995] e a porcentagem de média móvel de Wellner [Parker, 1996] separadamente.

Para cada imagem limiarizada á aplicado um filtro por Laplaciano utilizando a máscara 3x3. Com a imagem filtrada por Laplaciano normalizada é executado uma operação morfológica de dilatação de 1 (uma) interação utilizando o elemento estruturante quadrado 3x3. De posse disso é executado um processo de localização de contornos. Os contornos são armazenados em uma lista única.

A Figura 4.24 apresenta a primeira etapa da detecção. A Figura 4.24 (a) apresenta a imagem multi-limiarizada por N.Papamarkos e B. Gatos [Papamarkos, 1994], a Figura 4.24 (b) apresenta a imagem filtrada por Laplaciano (normalizado), a Figura 4.24 (c) apresenta a imagem após uma operação de dilatação de 1 (uma) interação na imagem filtrada por Laplaciano e por fim a Figura 4.24 (d) apresenta os contornos da imagem.

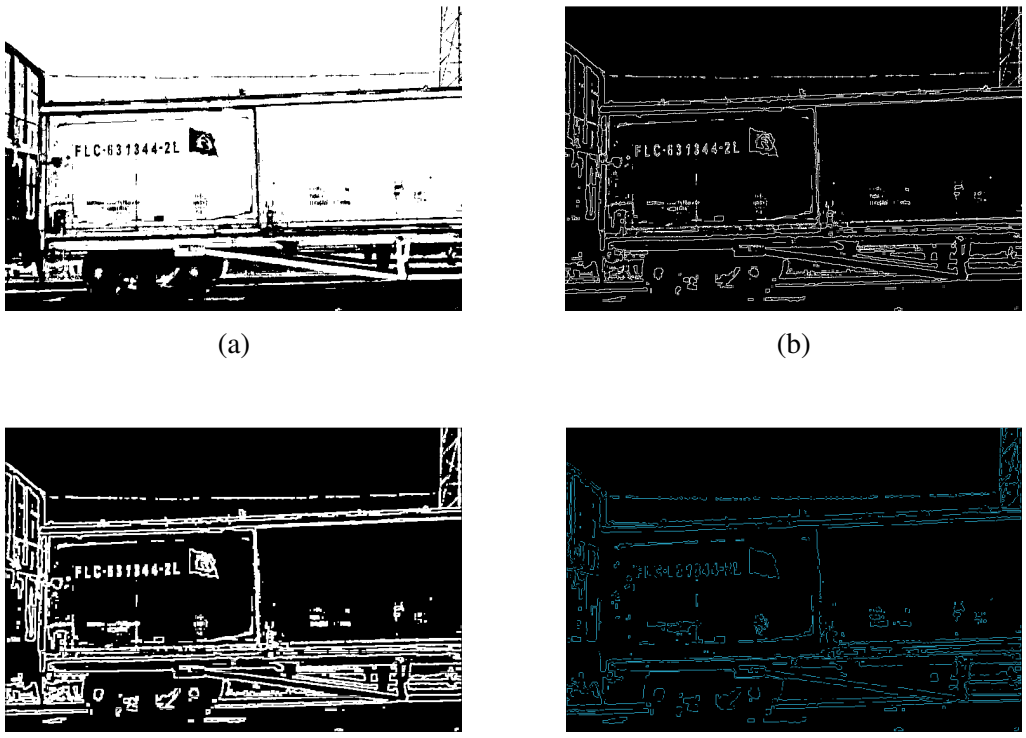


(a) (b)

(c) (d)

Figura 4.24 – Apresenta a primeira etapa de detecção, (a) imagem multi-limiarizada por Papamarkos [Papamarkos, 1994], (b) imagem filtrada por Laplaciano (normalizada)m (c) imagem dilatada e (d) contornos detectados.

A Figura 4.25 apresenta a segunda etapa da detecção. A Figura 4.25 (a) apresenta a imagem limiarizada por Bernsen [Bernsen, 1995], a Figura 4.25 (b) apresenta a imagem filtrada por Laplaciano (normalizado), a Figura 4.25 (c) apresenta a imagem após uma operação de dilatação de 1 (uma) interação na imagem filtrada por Laplaciano e por fim a Figura 4.21 (d) apresenta os contornos da imagem.

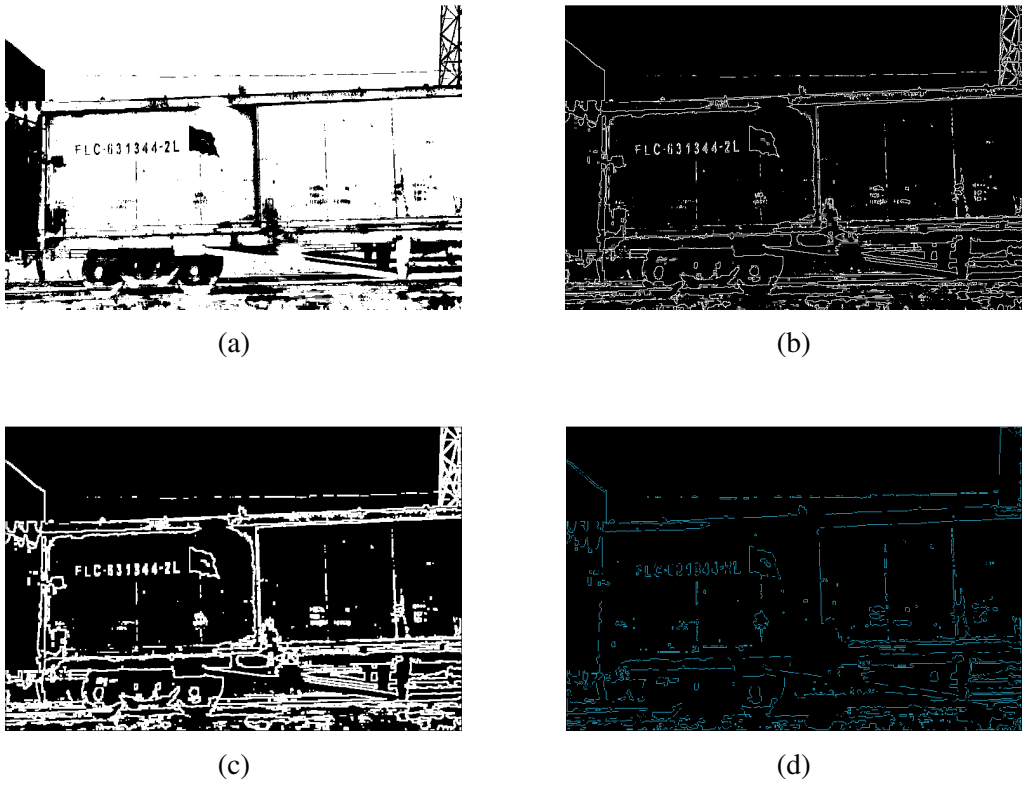


(a) (b)

(c) (d)

Figura 4.25 – Apresenta a segunda etapa de detecção, (a) imagem limiarizada por Bernsen [Bernsen, 1995], (b) imagem filtrada por Laplaciano (normalizada)m (c) imagem dilatada e (d) contornos detectados.

A Figura 4.26 apresenta a terceira etapa da detecção. A Figura 4.26 (a) apresenta a imagem limiarizada pelo percentual de média móvel de Wellner [Paker, 1996], a Figura 4.26 (b) apresenta a imagem filtrada por Laplaciano (normalizado), a Figura 4.26 (c) apresenta a imagem após uma operação de dilatação de 1 (uma) interação na imagem filtrada por Laplaciano e por fim a Figura 4.26 (d) apresenta os contornos da imagem.



(a) (b)

(c) (d)

Figura 4.26 – Apresenta a terceira etapa de detecção, (a) imagem limiarizada pelo porcentual de média móvel de Wellner [Paker, 1996], (b) imagem filtrada por Laplaciano (normalizada)m (c) imagem dilatada e (d) contornos detectados.

Após os contornos serem detectados, é realizada uma filtragem por altura. Esta filtragem consiste em coletar da lista de contornos respeitando os critérios estabelecidos para gerar uma nova imagem colorida com o fundo preto. Experimentalmente, estabeleceu-se que contornos com altura menor que 5 *pixels* e maior que 10 *pixels* serão ignorados.

A Figura 4.27 apresenta a nova imagem colorida gerada a partir do filtro por altura aplicado na lista de contornos.

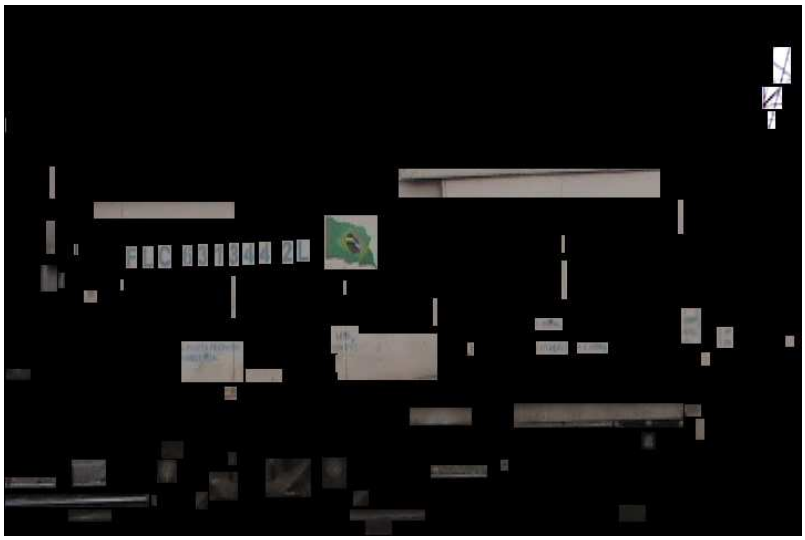


Figura 4.27 – Imagem gerada a partir do filtro por altura.

A nova imagem gerada á convertida para níveis de cinza. A Figura 4.28 apresenta a imagem “recortada” em níveis de cinza.

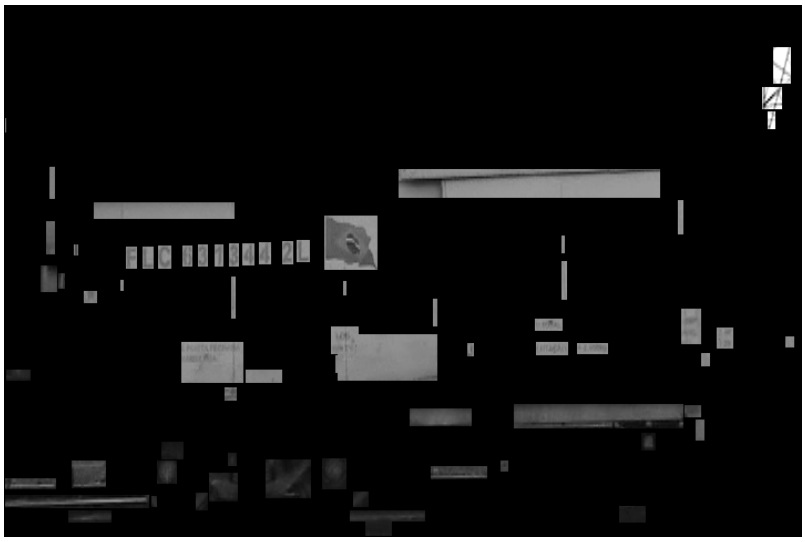


Figura 4.28 – Imagem “recortada” em níveis de cinza.

A próxima etapa a ser seguida é a etapa de Detecção do Bloco de Texto. Essa etapa consiste em aplicar o filtro do Laplaciano na imagem em níveis de cinza e aplicar o processo de MGD *Maximum Gradient Difference*.

A Figura 4.29 apresenta a imagem após o filtro do Laplaciano normalizado para [0, 1].

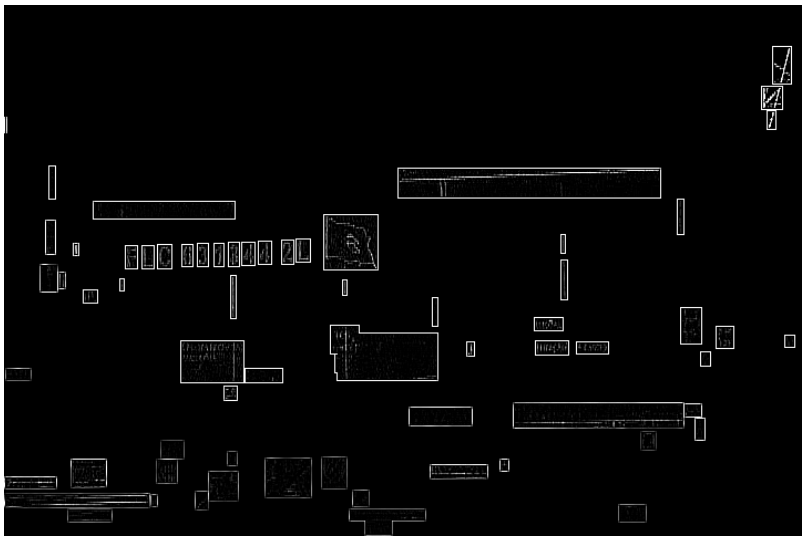


Figura 4.29 – Imagem do vagão graneleiro após o filtro por Laplaciano (normalizado).

Na seqüência é aplicado o processo de MGD *Maximum Gradient Difference,* para conectar os componentes A Figura 4.30 apresenta a imagem após esse processo.

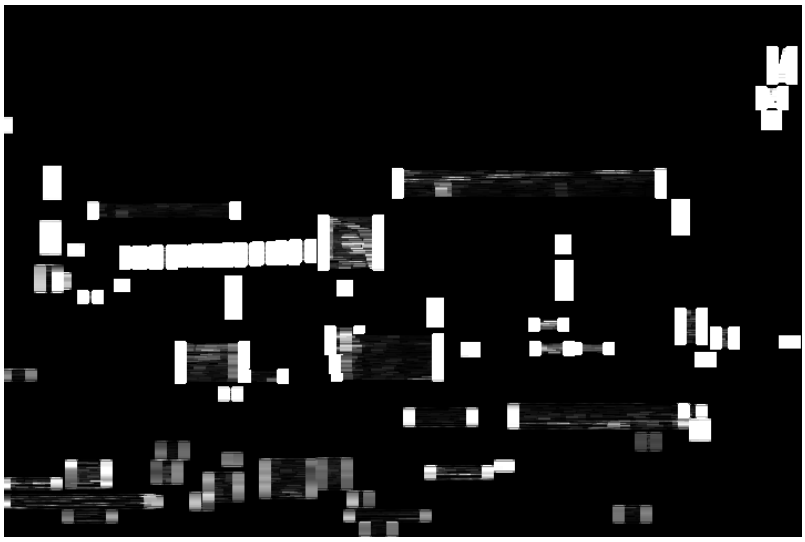


Figura 4.30 – Imagem do MDG *Maximum Gradient Difference* aplicado ao vagão graneleiro.

A etapa de Pós-processamento consiste em detectar os contornos e realizar a filtragem pelo fator de compacidade. A Figura 4.31 apresenta os contornos filtrados pelo fator de compacidade.

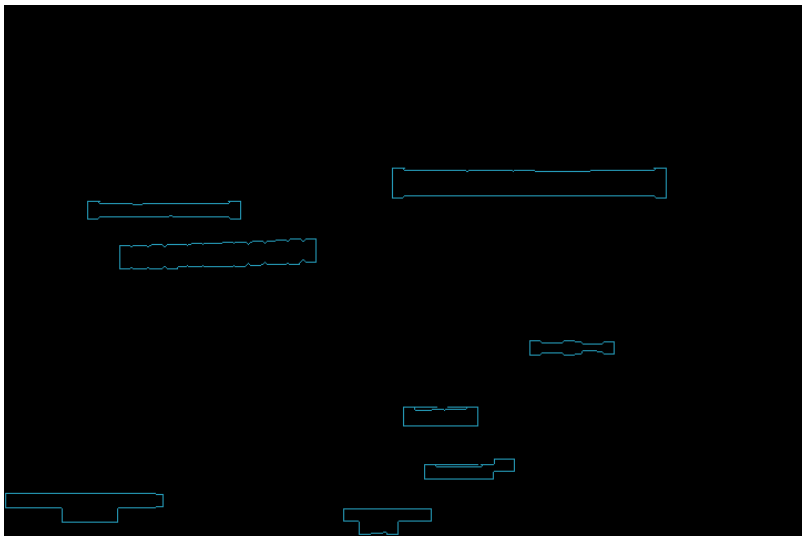


Figura 4.31 – Imagem dos contornos filtrados pelo fator de compacidade do vagão graneleiro.

Ainda na etapa de Pós-processamento, o último passo a ser dado consiste em apresentar a partir dos contornos filtrados pelo fator de compacidade os retângulos da localização. Como comentado anteriormente, uma última filtragem é realizada utilizando as propriedades geométricas.

A Figura 4.32 apresenta o resultado final do método proposto.

Figura 4.32 – Resultado da localização do código de identificação do vagão.

Os melhores resultados obtidos para essa abordagem são apresentados na Tabela 4.4

Parâmetros utilizados				
Tamanho Janela MGD	**11**			
Multilimiarização Papamarkos	**2**			
Limiarização Bernsen	**35**			
Média móvel de Wellner	**5**			
Fator Compacidade	**0,06**	a	**0,25**	
Tipo dos Vagões	Graneleiro + Tanque + Plataforma			
Resultados				
Total de amostras	116	**%**	**Média dos Blocos Candidatos**	**Desvio Padrão**
Acertos	**95**	**81,90%**	5,04	2,61
Erros	**21**	**18,10%**	4,62	2,40
	116	100,00%		

Tabela 4.4 – Resultado do método proposto utilizando a abordagem de detecção dos contornos das imagens limiarizadas.

4.5. Experimento baseado em Vídeo.

Nessa seção a busca pelo código é feita no vídeo sendo considerado como correta a localização, se o código foi encontrado ao menos uma vez nos múltiplos quadros onde este aparece.

Nesse experimento é utilizado o mesmo processo da seção anterior. No entanto, a grande diferença consiste em avaliar se em algum instante (nos múltiplos quadros dos vídeos) o método proposto foi capaz de localizar pelo menos uma vez o código de identificação do vagão.

Foram avaliados 2582 quadros onde aparecem 116 vagões diferentes. Os resultados obtidos para essa abordagem são apresentados na Tabela 4.5.

Parâmetros utilizados				
Tamanho Janela MGD	**11**			
Multilimiarização Papamarkos	**2**			
Limiarização Bernsen	**35**			
Média móvel de Wellner	**5**			
Fator Compacidade	**0,06**	a	**0,25**	
Tipo dos Vagões	Graneleiro + Tanque + Plataforma			
Resultados				
Total de amostras	116	**%**	**Média dos Blocos Candidatos**	**Desvio Padrão**
Acertos	**98**	**84,48%**	5,03	2,59
Erros	**18**	**15,52%**		
	116	100,00%		

Tabela 4.5 – Resultado do método proposto utilizando a abordagem de detecção dos contornos das imagens limiarizadas detectando no mínimo uma vez o código de identificação do vagão em múltiplos quadros.

Conforme apresentado na Tabela 4.5 a taxa de acerto (conseguiu localizar o código de identificação do vagão pelo menos uma vez) foi de 84,48 % e o erro (não conseguiu localizar o código de identificação) foi de 15,52 %. O número médio de blocos candidatos (blocos de textos candidatos a serem mandados para uma ferramenta de OCR) foi de 5,03 por quadro processado, e o desvio padrão foi de 2,59.

4.6. Considerações Finais

Nesse capitulo foi apresentado os resultados obtidos pelo método proposto levando em conta as 4 (quatro) abordagens para aplicação do método diretamente na imagem e uma abordagem para aplicação do método em vídeos.

Na abordagem que aplicou o método diretamente nas imagens selecionadas (imagens que garantiam a presença do código do vagão), os melhores resultados foram obtidos pela abordagem que utiliza a combinação dos contornos das imagens limiarizadas obtendo uma taxa de acerto na localização dos códigos de identificação dos vagões de 81,90%.

Quando aplicado o método proposto submetendo as imagens a um único processo de limiarização, os melhores resultados foram obtidos utilizando a abordagem que utiliza a técnica de multi-limiarização proposta por N.Papamarkos e B. Gatos [Papamarkos, 1994] com uma taxa de acerto na localização dos códigos dos vagões de 66,38 % indicando ser essa a melhor abordagem.

Tanto a abordagem que utiliza a técnica de limiarização loca adaptativa proposta por Bernsen [Bersen, 1995] como a abordagem que utiliza a porcentagem da média móvel de Wellner [Paker, 1996] apresentaram resultados insatisfatórios com uma taxa de acerto de 23,28 % e 21,55 % respectivamente.

Na abordagem que aplicou o método proposto utilizando as três técnicas de limiarização em conjunto diretamente nos vídeos foi possível observar uma melhora na taxa de acerto de 81,90% para 84,48%.

O gráfico da Figura 4.33 resume os resultados obtidos nos experimentos realizados.

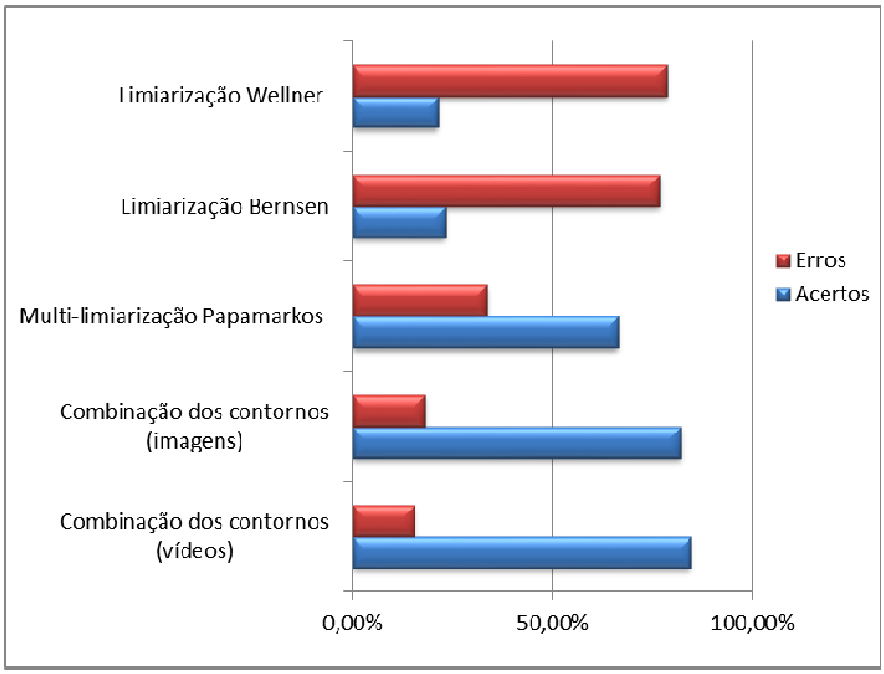


Figura 4.33 – Gráfico com resumo dos resultados obtidos.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho foi proposto e avaliada um método baseado em técnicas de via~so computacional para a localização de códigos de vagão em vídeos. O método proposto representa uma maneira simplificada de localizar os códigos em vagões sem impor restrições significativas às variáveis envolvidas no problema.

Os experimentos demonstraram que a combinação dos resultados de deferentes técnicas de limiarização é promissora, uma vez que possibilitou minimizar o impacto negativo da perta do código durante o processo de segmentação das imagens. O método utilizando a combinação dos contornos das imagens limiarizadas obteve uma taxa de acerto de 81,90%. Já o melhor resultado obtido utilizando apenas uma técnica de limiarização foi de 66,38%, demonstrando assim que a combinação dos contornos obtidos a partir de diferentes imagens limiarizadas possibilitou um ganho considerável em termos de localização correta.

Quando aplicado diretamente nos vídeos a fim de localizar pelo menos um código de vagão em múltiplos quadros, observou-se uma melhora adicional. A taxa de acerto passou de 81,90% para 84,48%. A razão está na possibilidade de utilizar a busca do código em vários quadros consecutivos.

Embora a grande vantagem do método proposto é de não impor restrições significativas à variáveis envolvidas no problema, sua desvantagem é o esforço computacional. O fato de se utilizar a filtragem por Laplaciano várias vezes acarreta um alto custo computacional.

Como uma proposta de estudo para trabalhos futuros, além da busca de redução do custo computacional, pode-se citar uma melhor investigação de técnicas para a eliminação de bloco de texto falsos.

Embora o método consiga localizar os códigos de identificação, a presença de falsos positivos pode encarecer o processo de leitura automática. A utilização do fator de compacidade como uma filtragem final nessa etapa se mostrou ineficiente. Uma alternativa interessante seria utilizar outros recurso após a filtragem do fator de compacidade, como uma análise da textura de cada bloco.

Referências Bibliográficas

[Li, 2000] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," IEEE Trans. Image Process., vol. 9, no. 1, Jan. 2000, pp.147–156.

[Jain, 1998] A. K. Jain and B. Yu, "Automatic text location in images and video frames," Pattern Recognit., vol. 31, no. 12, 1998, pp. 2055–2076.

[Zhong, 1999] Y.Zhong, H.-J.Zhang, A.K.Jain, Automatic caption localization in compressed video, in:Proceedings of the International Conference on Image Processing, vol.2, 1999, pp.96–100.

[Wolf, 2002] C. Wolf, J.-M. Jolion, F. Chassaing, Text localization, enhancement and binarization in multimedia documents, in: Proceedings of the 16th International Conference on Pattern Recognition, vol. 2, 2002, pp. 1037–1040.

[Zhang, 2003] D.Q. Zhang, B.L. Tseng, S.F. Chang, Accurate overlay text extraction for digital video analysis, in: Proceedings of the International Conference on Information Technology Research and Education, August 2003, pp. 233–237.

[Lienhart, 2002] R. Lienhart, A. Wernicke, Localizing and segmenting text in images, videos and web pages, IEEE Trans. Circuits Syst. Video Technol. 12 (4) (2002) 256–268.

[Gao, 1983] X. Gao, X. Tang, et al., Automatic news video caption extraction and recognition, in: K. S. Leung et al. (Eds.), Proceedings of the Lecture Notes in Computer Science 1983, Second International Conference on Intelligence Data Engineering and Automated Learning Data Mining, Financial Engineering, Intelligence Agents, Hong Kong, 2000, pp. 425–430.

[Wernicke, 2000] A. Wernicke, R. Lienhart, On the segmentation of text in videos, in Proceedings of the IEEE International Conference on Multimedia Expo, vol. 3, 2000, pp. 1511–1514.

[Cai, 2002] M. Cai, J. Song, M.R. Lyu, A new approach for videotext detection, in: Proceedings of the International Conference on Image Processing, September 2002, Rochester, NY, pp. 117–120.

[Antani, 2000] S. Antani, D. Crandall, and R. Kasturi, "Robust extraction of text in video," in Proc. 15th Int. Conf. Pattern Recognit., vol. 1, 2000, pp. 831–834.

[Lyu 2005] Lyu, M.R., Jiqiang Song, Min Cai, "A comprehensive method for multilingual video text detection, localization, and extraction", IEEE Trans. Circuits Syst. Video Technol., Volume 15, Issue 2, Feb. 2005, pp. 243 – 255.

[Hua, 2002] X. S. Hua, P. Yin and H. J. Zhang, "Efficient Video Text Recognition Using Multiple Frame Integration", International Conference on Image Processing, 2002.

[Chen, 2001] D. Chen, K. Shearer, and H. Bourlard, "Text enhancement with asymmetric filter for video OCR," in Proc. 11th Int. Conf. Image Anal. Process., 2001, pp. 192–197.

[Sato, 1998] T. Sato, T. Kanade, E. K. Hughes, and M. A. Smith, "Video OCR for digital news archive," in Proc. IEEE Workshop Content-Based Access Image Vídeo Database, 1998, pp. 52–60.

[Chun, 1999] B. T. Chun, Y. Bae, and T.-Y. Kim, "Text extraction in videos using topographical features of characters," in Proc. IEEE Int. Fuzzy Syst. Conf., vol. 2, 1999, pp. 1126–1130.

[Kwak, 2000] S. Kwak, K. Chung, Y. Choi, “Video Caption Image Enhancement for an Efficient Character Recognition”, in Proc. 15th Int. Conf. Pattern Recognit., vol. 2, 2000, pp. 2606–2609.

[Trung, 2009] Trung Quy Phan, Palaiahnakote Shivakumara and Chew Lim Tan, “A Laplacian Method for Video Text Detection” in 2009 10th International Conference on Document Analysis and Recognition

[Jian, 2009] Jian Yi, Yuxin Peng, and Jianguo Xiao, “Using Multiple Frame Integration for the Text Recognition of Vídeo”, in 2009 10th International Conference on Document Analysis and Recognition

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

[Wong, 2003] E. K. Wong and M. Chen, “A new robust algorithm for video text extraction”, Pattern Recognition 36, 2003, pp. 1397-1406.

[Pratheeba, 2010] T.Pratheeba , Dr.V.Kavitha and S.Raja Rajeswari,
“Morphology Based Text Detection and Extraction from Complex Video Scene”, International Journal of Engineering and Technology Vol.2(3), 2010, 200-206

[Palaiahnakote, 2008] Palaiahnakote Shivakumara, Weihua Huang and Chew Lim Tan, “Efficient Video Text Detection using Edge Features”, 2008 IEEE.

[Xiaoqing, 2006] Xiaoqing Liu and Jagath Samarabandu, “Multiscale Edge-Based Text Extraction From Complex Images”, 2006 IEEE, ICME 2006.

[Liao, 2009] S. Liao, Max W. K. Law, and Albert C. S. Chung,” “Dominant Local Binary Patterns for Texture Classification”, IEEE Transactions on Image Processing, Vol. 18, No. 5, May 2009.

[Ojala, 2002] T. Ojala, M. Pierikainen, and T. Maenpaa, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp. 971–987, Jul. 2002.

[Papamarkos, 1994] N.Papamarkos and B. Gatos – A New Approach for multilevel Threshod Selection, 1994 CVGIP Graphical Models And Image Processing Vol 56, No. 5, September, pp 357-370

[Bernsen, 1995] Bernsen, J., Dynamic Thresholding of gray-level images, Proc. Eighth Int'l Conf. on Pattern Recognition, Paris, France, oct 1986, pp 1251-1255, Trier, O.D. and Jain A.K., ”Goal-Directed Evaluation of Binarization Methods”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol.17, no.12, december 1995, pp.1191-1201.

[Parker, 1996] Parker J.R., Algorithms for Image Processing and Computer Vision, John Wiley and Sons, pp 145-148, 1996.

[Burt, 1981] P.J. Burt. Fast filter transforms for image processing. Computer Graphics and Image Processing, 1981.