PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

Universidade Federal do Rio de Janeiro

DISSERTAÇÃO DE MESTRADO

Segmentação de caracteres tipográficos em imagens complexas

Diego Gouvêa Macharete Trally

Universidade Federal do Rio de Janeiro

Curso de Mestrado

Orientador: Antonio Carlos Gay Thomé

Ph.D

Rio de Janeiro

2011

Instituto de Matemática

Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais

Diego Gouvêa Macharete Trally

Segmentação de caracteres tipográficos em imagens complexas

Volume único

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Informática.

Orientador: Antonio Carlos Gay Thomé

Rio de Janeiro

2011

T758 Trally, Diego Gouvêa Macharete

Segmentação de caracteres tipográficos em imagens complexas / Diego Gouvêa Macharete Trally, 2011

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais, 2011

Orientador: Antonio Carlos Gay Thomé

1. Segmentação de Caracteres - Teses. 2. Processamento de imagens - Teses. 3. OCR – Teses.

I.Thomé, Antonio Carlos Gay (Orient.). II. Universidade Federal do Rio de Janeiro. Programa de Pós-Graduação em Informática. III. Título.

CDD:

Diego Gouvêa Macharete Trally

Segmentação de caracteres tipográficos em imagens complexas

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática, Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências em Informática.

Rio de Janeiro, 26 de Agosto de 2011

_____

Antonio Carlos Gay Thomé, Ph.D., DCC-IM/UFRJ – Orientador

_____

Adriano Joaquim de Oliveira Cruz, Ph.D., DCC-IM/UFRJ

_____

Jose Antonio dos Santos Borges, Doutor, DCC-IM/UFRJ

_____

Marley Maria Bernardes Rebuzzi Vellasco, Ph.D., PUC-RIO

Dedicatória

Dedico este novo marco em minha vida mais uma vez à Camila, agora minha esposa, pois sem ela eu não chegaria tão longe e, mesmo que chegasse, nada disso teria sentido.

Agradecimentos

Ao professor Thomé, pelos ensinamentos, pelo apoio e pela dedicação e acima de tudo por acreditar em mim e na concretização deste trabalho.

À minha esposa Camila, pela sua assistência incondicional, pela compreensão e principalmente por me ajudar a seguir em frente e não me desviar do caminho.

A meus pais, por me fazerem chegar até aqui, e por compreenderem minha falta de tempo nesses últimos anos.

Ao pessoal do trabalho, pelo apoio oferecido para a conclusão do mestrado.

E aos meus amigos, pelas ajudas, idéias e momentos de descontração que também foram indispensáveis para a finalização desta etapa.

Resumo

Este trabalho apresenta o desenvolvimento de um sistema de segmentação de caracteres em imagens complexas com o objetivo de se criar um leitor robusto. O trabalho foi desenvolvido sobre imagens de veículos em ambientes não controlados, contendo textos em diversos locais além da própria placa do automóvel. Assume-se que os caracteres em uma imagem são monocromáticos e com isso utiliza-se uma técnica de particionamento da imagem conhecida como *split & merge*, capaz de gerar componentes conexos baseado na informação do espaço de cores da imagem. Uma abordagem desta técnica utilizando uma representação em árvore binária foi aplicada. Para refinar a divisão dos componentes conexos gerados, desenvolveu-se uma técnica de reconstrução do plano de fundo e de reagrupamento. O sistema apresentou um desempenho de 79,70% de acerto no conjunto de desenvolvimento. Para fins de comparação utilizou-se um banco de imagens público e o sistema apresentou um desempenho de 64,56%.

Abstract

This paper presents the development of a system for character segmentation in complex images with the objective of creating a robust reader system. The study was conducted on images of vehicles in unconstrained environments, containing texts in various locations besides the license plate of the car. It is assumed that the characters in a image are monochromatic and it is applied a image partitioning technique known as split & merge, capable of generating connected components based on the information of the image color space. An approach of this technique using a binary tree representation is applied. To refine the division of the generated connected components, a technique was developed for background reconstruction and regrouping. The system provided a performance of 79.70% on the development set. For comparison purposes a public image database was used and the system achieved a performance of 64.56%.

Índice de figuras

Índice de tabelas

GLOSSÁRIO

BB – *Bounding Box* – É o menor retângulo que envolve um determinado objeto.

LPR – *Licence Plate Recognition* (Reconhecimento de placas de automóveis)

MLP – Multi-Layer Perceptron

OCR – *Optical Character Recognition* (Reconhecimento Óptico de Caracteres)

S&M – *Split And Merge* – Técnica de identificação de grupos de uma imagem por cores.

SVM – Suporte Vector Machine

Sumário

1 Introdução

Desde o início, os computadores surgiram com o objetivo de simplificar e automatizar as tarefas do homem. Porém, muitas destas tarefas ainda requerem a habilidade humana, dado seu grau de complexidade, inexistência de tecnologia ou devido às diferenças intrínsecas entre a capacidade humana e a capacidade computacional. Com o desenvolvimento de novas tecnologias essa diferença de capacidade vem diminuindo cada vez mais. Porém, apesar de todo o avanço da informática, o computador ainda não consegue imitar eficientemente boa parte das capacidades do homem, como por exemplo, sua grande capacidade de identificar e reconhecer padrões de formas e símbolos, como objetos, pessoas, texto, etc. Isso explica o grande interesse da comunidade científica em pesquisas que consigam simular tais capacidades.

No início de sua utilização, um computador era apenas uma ferramenta de trabalho limitada ao processamento de texto e operações aritméticas, agregando posteriormente a função de entretenimento. Nesse cenário a máquina era utilizada para facilitar ou dar condições de trabalho ao seu operador humano. O computador era apenas um armazenador de dados, não tendo a capacidade de interpretá-los. Por exemplo, embora tivesse a possibilidade de gravar e reproduzir sons, utilizando microfones e caixas de som, não era capaz de interpretar sons atribuindo-lhes significado (emulando a audição) nem de sintetizar textos em sons, como fazemos quando falamos. Atualmente existem diversos aplicativos para reconhecer e sintetizar sons, mostrando a evolução da tecnologia nesta área e a busca pela capacidade de emular as capacidades humanas através de uma máquina.

Outro exemplo que pode ser citado é o sentido da visão e a capacidade de localizar e reconhecer padrões, bem como extrair o seu significado. O exemplo mais claro é a capacidade do ser humano de ler. A comunicação escrita é, junto à falada, a forma de transmissão de informação mais utilizada, e este tipo de informação se encontra em todo o lugar, não apenas em documentos arquivados dentro de um computador. Para interpretar a informação escrita utilizamos nossa capacidade de visão e a grande habilidade humana de reconhecer padrões para identificar as letras, e depois nosso cérebro se responsabiliza por juntá-las e dar significado a cada uma delas e/ou ao seu conjunto. Simular em um computador essa habilidade de identificar textos é uma área de pesquisa antiga, e é conhecida como reconhecimento óptico de caracteres (OCR).

1.1 Reconhecimento Óptico de Caracteres

O OCR surgiu há pouco mais de seis décadas (MORI, SUEN e YAMAMOTO, 1992), primeiramente como um aplicativo para identificar textos obtidos através de digitalizadores de documentos, os *scanners*. Sua aplicação se resumia apenas a documentos bem específicos, escritos sobre folhas brancas, com letras pretas de tamanho e fonte bem definidos. Com o avanço tecnológico tanto dos computadores como das câmeras fotográficas ou de vídeo, o OCR se tornou muito mais amplo, englobando a automatização de diversos processos que antes dependiam integralmente da ação humana. Exemplos desses processos são:

- **Controle de processos industriais** – Onde o computador reconhece textos que servem para guiar uma linha de produção, ou para verificar a qualidade de um produto.

- **Reconhecimento de placas de veículos (LPR)** – O computador localiza e reconhece a placa de um automóvel, que tem diversas aplicações, como controle de acesso, monitoramento e segurança, aplicação de multas, etc.
- **Leitura automática de formulários** – Busca reconhecer a escrita humana em um formulário de forma a automatizar o cadastro ou a catalogação de algum dado, como por exemplo, registro em concursos, ou reconhecimento de CEP.
- **Digitalização de documentos** – A primeira aplicação prática de um OCR. Tem como objetivo transformar uma imagem de um documento em texto para o computador. Com isso o arquivo do documento passa a ocupar menos espaço, pois não é uma imagem, além de se tornar possível sua indexação e a busca por palavras dentro do documento.
- **Catalogação de imagens** – Parte do mesmo princípio da digitalização de documentos, porém é mais complexo pois trabalha com quaisquer imagens, diferente das fotos de documentos obtidas em ambiente controlado. Busca extrair os textos contidos na imagem de forma que essa informação ajude na identificação e indexação da imagem. Por exemplo, listar imagens com o nome de uma certa loja.

Apesar da pesquisa nesse ramo existir a mais de meio século, ainda hoje há desafios a serem superados. Isso porque, com os avanços da tecnologia, as demandas de diferentes soluções vão surgindo, além de exigências mais rígidas na robustez, na precisão e na velocidade de um algoritmo de reconhecimento de caracteres. O OCR evoluiu ao longo das décadas de um simples leitor de documentos e passou a abranger problemas maiores, como ler textos em qualquer imagem. É nesse contexto que se

encontra esse trabalho, que se propõe a localizar e segmentar textos em imagens complexas. Neste trabalho, se define por imagem complexa como qualquer imagem capturada em ambiente natural não-controlado, podendo conter um número elevado de objetos, sejam eles caracteres ou não. Nesse tipo de imagem não há nenhuma restrição quanto ao número de caracteres, nem ao seu tamanho, formato, inclinação ou cor. A Figura 1.1 mostra dois exemplos de imagens complexas, a primeira é uma foto de uma porta de vidro que reflete o cenário ao fundo, e a segunda é uma foto da traseira de um veículo, onde pode ser observados efeitos como sombra e reflexo.

Figura 1.1 - Exemplos de imagens complexas

Basicamente um sistema de OCR, assim como o ser humano, divide seu processo em três etapas: A aquisição da imagem, a segmentação e o reconhecimento. A aquisição é a etapa responsável por capturar a percepção visual do ambiente e transformá-la em dados de forma a serem armazenados e interpretados em um computador. Isso pode ser feito através de uma foto ou de um vídeo. Uma foto normalmente apresenta uma resolução melhor, o que significa mais nitidez e precisão nas informações que serão disponibilizadas. Porém, uma imagem maior leva mais tempo para ser gerada e processada. Para o caso de aplicações que se propõem a serem executadas em tempo real, isso implica na necessidade de equipamentos mais rápidos e consequentemente mais caros. A opção de utilizar câmeras de vídeo dificulta o processamento, pois devido

ao princípio de formação da imagem nesse tipo de câmera, ela se encontra mais sujeita a ruídos e deformações na imagem. Tanto em uma quanto em outra forma de aquisição, a imagem está sujeita a distorções, ruídos e efeitos de iluminação que dificultam o processamento da imagem. Reflexos e sombras são exemplos destes efeitos que atrapalham o processamento e podem ser difíceis de serem corrigidos.

A segmentação de caracteres é uma das principais etapas, pois é a que trata da localização e extração dos caracteres da imagem. O seu desempenho pode afetar sensivelmente todas as etapas posteriores. É nessa etapa que o OCR extrai da imagem as informações desejadas, tendo idealmente ao final do seu processo recortes da imagem apenas onde contenham caracteres. Essas sub-imagens variam de acordo com o tipo de entrada que a camada de reconhecimento espera receber. Pode ser recortes contendo apenas um caractere, ou uma palavra inteira.

Alguns trabalhos dividem a camada de segmentação em duas etapas: uma responsável por uma análise macro da imagem, extraindo uma ou mais regiões de interesse que possui alta probabilidade de conter caracteres, e uma etapa que recebe essa(s) imagem(s) e localiza e extrai os caracteres propriamente ditos(GUINGO, STIEBLER e THOMÉ, 2004)(CHEN, BOURLARD e THIRAN, 2001). Para localizar e extrair os caracteres da imagem adquirida, essa camada se utiliza de diversas técnicas de processamento de imagem. Muitos trabalhos realizam primeiro um tratamento na imagem, de forma a tornar mais uniforme o tipo de imagem a ser processada ou de corrigir/diminuir distorções e ruídos. Pode ser feita também um tratamento posterior, ao longo do algoritmo, para corrigir falhas mais localizadas. Após esse pré-tratamento a imagem passa por uma série de transformações, buscando extrair diversas informações

para identificar regiões de interesse. Ao final do processamento obtém-se a região com o(s) caractere(s) que pode passar por uma etapa de pós-processamento para prepará-la para a camada posterior.

A terceira etapa de um OCR é a de reconhecimento, responsável por identificar e dar significado ao símbolo recebido da etapa anterior. É a responsável em analisar as características do segmento de imagem recebido e identificar qual símbolo ou palavra este representa. Diversas são as características utilizadas, desde características estatísticas, geométricas, morfológicas ou alguma outra análise que o pesquisador julgar conveniente, sempre buscando uma maior taxa de acerto e uma maior robustez, ou buscando aceitar símbolos diversos e multilíngües. Normalmente se utiliza uma técnica de aprendizado de máquina para decidir qual caractere é mais provável de possuir as características extraídas, ou uma técnica de casamento de padrões. Nessa etapa espera-se que o caractere chegue da forma mais uniforme possível, para que seu tamanho e inclinação não atrapalhem o processo de reconhecimento.

1.2 Sistemas de reconhecimento de placas de veículos

Um sistema de LPR tem como objetivo localizar e reconhecer a placa do veículo e tem diversas aplicações, principalmente nas áreas de controle de acesso e segurança. Algumas das aplicações deste tipo de sistema são:

- Controle de acesso de veículos – Este tipo de sistema, utilizado em estacionamentos e condomínios, tem como objetivo identificar se o veículo possui permissão para trafegar no local instalado. O sistema obtém a placa do carro e busca em um banco de dados a informação de acesso daquele veículo. Assim pode-se identificar se o veículo possui acesso àquele lugar, ou no caso de

estacionamentos públicos, cadastra aquele veículo no banco de dados anotando seu horário de entrada e saída.

- Radar eletrônico / Multas – Um sistema de LPR pode automatizar o processo de aplicação de multas em lugares que possuem um radar ou outro sistema de detecção de infração de trânsito. Ao detectar a infração, o sistema tira uma fotografia do automóvel infrator e passa pelo sistema de LPR para identificar a placa do veículo e registrar a multa na base de dados do departamento de trânsito.
- Monitoração das ruas – Um sistema de LPR capaz de identificar as placas por uma câmera de vídeo pode ser utilizado para monitoração contínua das ruas, podendo identificar e alarmar para a polícia ou departamento de trânsito por onde carros roubados (ou outro tipo de carro que se queira localizar) passaram, facilitando a busca e apreensão.

Os sistemas de localização de placas normalmente se baseiam nas características únicas das placas para encontrá-las na imagem e extrair sua licença. Por exemplo, as placas no Brasil para veículos de passeio são cinza com sete caracteres pretos. Esse tipo de característica levou ao desenvolvimento de algoritmos bastante específicos para localizar este tipo de placa. Uma adaptação a esse modelo deve ser feita para poder contemplar outras placas brasileiras, como as de carros de serviço (taxis e caminhões) que são vermelhas com letras brancas. Este mesmo programa não poderia ser utilizado na Argentina, por exemplo, onde a placa possui um padrão diferente do brasileiro. Daí surge a necessidade de se utilizar um leitor robusto, capaz de identificar caracteres

independentes de sua cor de fundo, que pode ser utilizado em qualquer imagem de veículo de qualquer país.

1.3 Motivação

Conforme descrito nas sessões anteriores, a pesquisa sobre como emular as capacidades humanas através de computadores, apesar de antiga, permanece desafiadora, dentre elas a visão e o reconhecimento de padrões que nos dão a capacidade de ler em qualquer ambiente. No cenário mais atual da pesquisa sobre OCR, se destacam os leitores robustos, que se propõem a identificar textos em qualquer tipo de imagem.

Neste cenário, este trabalho busca desenvolver um leitor robusto direcionado para aplicações de LPR e comparar seu desempenho com outros trabalhos relacionados à segmentação de caracteres em imagens complexas.

O principal problema que um leitor robusto tem que lidar é a falta de informação com relação ao que se deseja buscar. Não se sabe a quantidade de caracteres contidas na imagem (ou se existem), e também não se possui nenhum outro tipo de informação que facilite a busca, como cor do fundo, cor da letra, posição esperada, entre outras.

Outro desafio está no fato de a imagem não ser controlada, o que significa que pode possuir uma complexidade de objetos elevada e o leitor deve ser capaz de identificar, nesse cenário, apenas os caracteres, sendo que muitas vezes objetos podem ser confundidos com letras, como por exemplo, um farol redondo de um carro pode se passar pela letra “O”, ou o espaçamento de uma grade que pode ser facilmente confundido como uma sucessão de letras “I”.

1.4 Objetivo

Este trabalho tem como objetivo desenvolver a etapa de segmentação de um leitor robusto. O sistema deve ser capaz de segmentar todos os caracteres legíveis de uma imagem complexa. Os caracteres a serem considerados nesse trabalho são apenas os numerais (0-9) e aqueles pertencentes ao alfabeto latino (A-Z, inclusive K,W e Y). Símbolos (&, %, $, +, -, /, etc.) não serão considerados, bem como acentos gráficos, o que significa que uma vogal acentuada que tenha sido segmentada sem o acento não será considerada como um erro.

Apesar de se propor a extrair caracteres em qualquer tipo de imagem, o foco deste trabalho é identificar caracteres em imagens contendo veículos e ser utilizado em sistemas de LPR, com o objetivo de identificar os caracteres da placa, bem como outros caracteres existentes em um veículo e que possam facilitar sua identificação (por exemplo, número de um ônibus).

1.5 Organização da dissertação

O capítulo 2 apresenta uma revisão sobre a área de processamento de imagens que servirá como base teórica para o entendimento dos capítulos posteriores. Nesse capítulo serão apresentados os conceitos de formação de uma imagem digital, assim como as técnicas mais conhecidas de manipulação e modificação da imagem.

O capítulo 3 traz um levantamento bibliográfico sobre o estado da arte em segmentação de caracteres e em leitores robustos. Serão apresentados alguns dos trabalhos existentes nessa linha de pesquisa, bem como os tipos de técnicas abordadas para cada tipo de segmentação.

No capítulo 4 é descrito o desenvolvimento feito neste trabalho, descrevendo a abordagem de segmentação por cores e a técnica de *split* & *merge*, que foram adotadas como base para este trabalho, além das outras técnicas desenvolvidas nesse trabalho.

O capítulo 5 apresenta os resultados deste trabalho em seus diversos testes. Além disso, será apresentada os resultados do sistema utilizando as imagens da competição de leitura robusta ICDAR, que ocorreu nos anos de 2003 e 2005 e disponibilizou um banco de imagens para trabalhos desenvolvidos na área de leitura robusta. Estes resultados serão comparados com os obtidos em outros trabalhos inscritos nesta competição, de forma a comparar o desempenho do sistema frente a outros trabalhos publicados.

O capítulo 6 encerra este trabalho apresentando as conclusões finais geradas a partir deste estudo, apresentando uma visão crítica do sistema desenvolvido apresentando suas vantagens e limitações, além de sugestões de melhorias para trabalhos futuros.

2 Processamento Digital de Imagens

A área de processamento de imagens diz respeito ao conjunto de operações realizadas sobre uma imagem de modo a transformá-la e/ou extrair informações dela. Nesse capítulo será feita uma apresentação desta área de conhecimento, passando por conceitos básicos e operações mais comuns, que servirão de base para o entendimento dos trabalhos descritos no capítulo seguinte, além das técnicas utilizadas e desenvolvidas neste trabalho.

2.1 Conceitos básicos de imagem

Por imagem defini-se como uma função bidimensional f(x,y) onde x e y são coordenadas espaciais do plano, e a amplitude de f em um dado ponto (x,y) é chamado de intensidade ou nível de cinza da imagem naquele ponto (GONZALES e WOODS, 2002). Neste trabalho, é convencionado que o eixo X representa o eixo horizontal, enquanto o eixo Y representa o eixo vertical, sendo a origem das coordenadas sempre o canto superior esquerdo da imagem, conforme mostra a Figura 2.1.

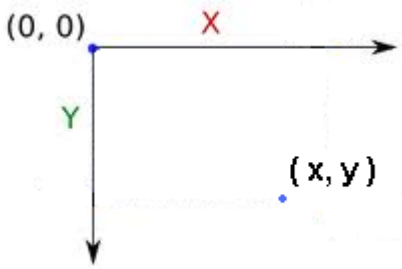

Figura 2.1 - Eixo das coordenadas de uma imagem

Quando *x,y* e *f* são quantidades finitas e discretas, chamamos de imagem digital. O elemento primário que compõe a imagem é conhecido como pixel (forma compacta de

Picture Element, do inglês, ou elemento da imagem), que é dado como a menor divisão da imagem. A Figura 2.2 mostra uma imagem real (esquerda) e sua representação digital (direita). Os quadrados em forma de grade são os pixels da imagem. Como em qualquer quantização, é possível perceber que há perda de informação na imagem digital. Essa perda pode ser reduzida aumentando a quantidade de informação contida na imagem digital, ou seja, aumentando o número de pixels que compõem a imagem digitalizada. A quantidade de pixels pode estar limitada pelo equipamento utilizado na digitalização, pelo espaço em disco ocupado pela imagem, pelo erro aceitável da imagem digitalizada, ou pela quantidade de informação que se está disposto a processar, pois quanto maior a imagem, mais lento será o algoritmo de processamento de imagem.

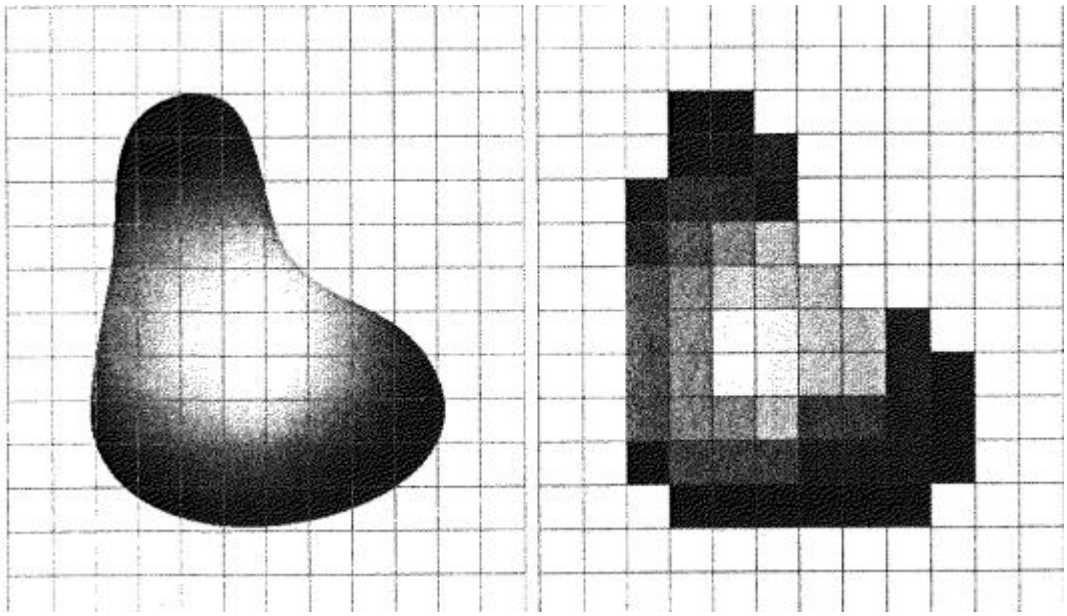

Figura 2.2 - Representação de uma imagem na forma digital em uma matriz de 14x12 pixels(GONZALES e WOODS, 2002)

2.1.1 Espaço de cores

Uma imagem pode ser representada por mais de um plano, de acordo com o espaço de cores utilizado. Espaço de cores é a especificação de um sistema de

coordenadas e um subespaço contido nesse sistema onde cada cor é representada por um único ponto (GONZALES e WOODS, 2002). Os espaços de cores mais utilizados são:

- RGB (*Red, Green and Blue* – vermelho, verde e azul), utilizado em monitores e imagens bitmaps por serem mais intuitivos e representarem exatamente a captação do hardware que obtém a imagem,
- CMY (*Cyan, magenta, yellow*) ou CMYK (*Cyan, magenta, yellow, black* – Ciano, magenta, amarelo e preto), utilizado em impressoras pois descrevem melhor as propriedades subtrativas dos pigmentos,
- HSI ou HSV (*Hue, saturation, Intensity/Value* – Matiz, saturação e intensidade), que se aproxima mais da forma como o ser humano descreve e interpreta as cores, além de ter a vantagem de desacoplar a informação de cor e de intensidade.

A Figura 2.3 mostra a representação destes três espaços de cores descritos. A primeira mostra o RGB, com suas cores básicas sendo utilizadas de forma aditiva para formar as outras cores. A segunda imagem representa o espaço CMY e sua propriedade subtrativa para formar as outras cores do espectro. A terceira imagem mostra a representação do espectro de cores no HSV representada por um cone de cores, onde a altura do cone representa sua intensidade luminosa, o raio do cone representa a saturação e o ângulo representa a matiz. Aqui é possível ver claramente a separação entre a intensidade luminosa e a cor, pois dada uma altura qualquer do cone, todas as cores representadas possuem a mesma luminosidade, sendo as cores representadas apenas pelos valores de H e S.

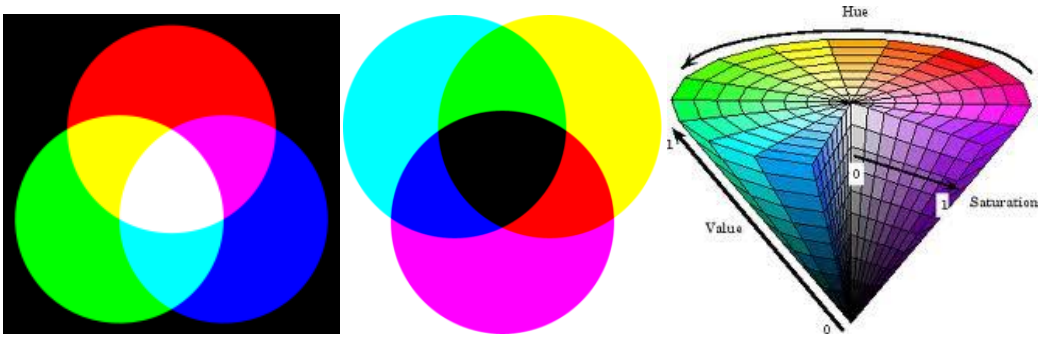

Figura 2.3 - Representação gráfica dos espaços RGB, CMYK e HSV

Quando é utilizada uma imagem em tons de cinza, utiliza-se uma redução do espaço de cores para apenas uma dimensão. Nesse caso normalmente é utilizada apenas a informação de intensidade luminosa que é percebida pelo olho humano. A conversão de uma cor no espaço RGB para nível de cinza é feita baseada na percepção do olho para cada cor primária, que é 30% de vermelho, 59% de verde e 11% de azul, conforme a Equação 2.1 explicita.

$$L(x,y) = 0.3 * R(x,y) + 0.59 * G(x,y) + 0{,}11 * B(x,y) \qquad 2.1$$

Nessa redução do espaço de cores, ocorre perda de informação, de forma que um ponto em um espaço de cores qualquer não é unicamente mapeado no espaço de tons de cinza. De fato, pela equação 2.1 pode-se fixar os valores de duas variáveis e calcular a terceira para se obter o mesmo valor de intensidade. A Figura 2.4 mostra uma imagem colorida que ao ser convertida para tons de cinza passa a apresentar apenas um nível de cinza, perdendo a distinção entre as cores.

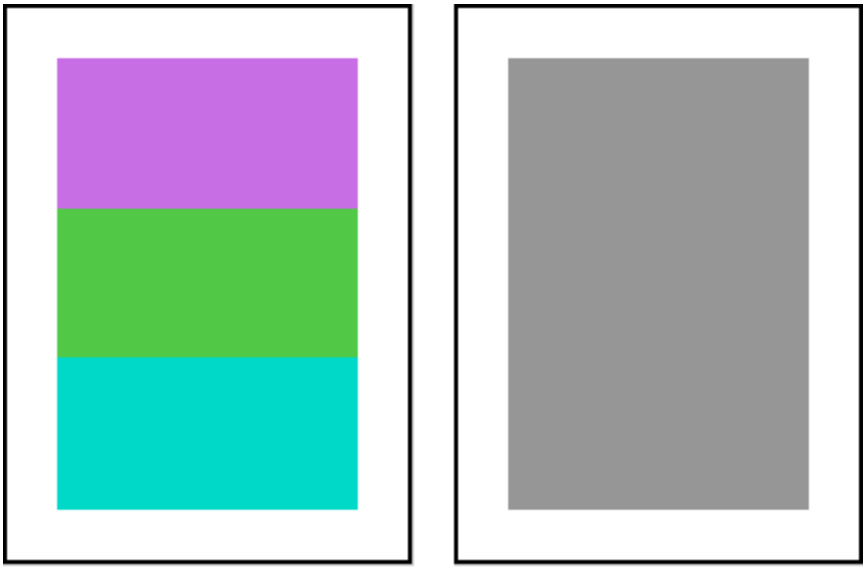

Figura 2.4 - Perda de unicidade causada pela conversão de uma imagem colorida para tons de cinza

2.1.2 Redução do espaço de cores - Binarização

Uma técnica muito utilizada em processamento de imagens, que pode ser compreendida como uma técnica de redução do espaço de cores, é a binarização (*thresholding*, em inglês). Consiste na conversão de uma imagem qualquer para apenas dois níveis de cores, representados por preto e branco, ou zero e um. Para possibilitar uma melhor visualização, quando tratarmos de imagem binária será invertido o conceito e apresentaremos a cor branca representando valor zero e a cor preta representando valor um.

A binarização é utilizada para se separar regiões de interesse da imagem. Não há uma regra específica para a operação, embora a técnica de *thresholding* considere que pixels acima de um limiar assumam valor um e pixels abaixo desse limiar recebam o valor zero. De forma mais ampla, qualquer regra booleana pode ser aplicada para se obter uma imagem binarizada. A Figura 2.5 mostra dois exemplos de binarização de uma placa de automóvel vermelha convertida para tons de cinza. Na primeira binarização foi

escolhido um limiar fixo escolhido com pouco critério e na segunda foi utilizado um método automático de escolha de limiar.

Figura 2.5 - Exemplos de binarização: a) imagem original, b) imagem binarizada com limiar fixo de 152, c) imagem binarizada com limiar calculado pelo método de Otsu.

O grande problema nesta técnica é a escolha correta do limiar de binarização. Este valor varia dependendo da imagem e do objeto que se deseja buscar, podendo algumas vezes nem existir, sendo o ideal utilizar uma faixa de valores. Diversos trabalhos foram feitos para se buscar de forma automática esse limiar ideal, dentre eles o mais conhecido é o método de Otsu (OTSU, 1979), que assume que a imagem está dividida entre primeiro plano e o plano de fundo e traça o histograma da imagem. Idealmente o histograma consiste de dois picos, representando esses dois planos, e o limiar seria o vale entre esses dois picos. Em imagens reais esses picos não estão bem definidos assim como o vale. O método busca identificar o ponto que minimize a variância dos pixels dentro de cada classe, encontrando então o limiar ótimo. A Figura 2.5 c mostra a imagem binarizada através do limiar calculado pelo método de Otsu, onde se pode observar uma imagem com menos informação irrelevante se comparada à primeira imagem.

2.2 Relações básicas entre pixels

Muitas técnicas de processamento de imagens trabalham em nível de pixel, ou seja, é analisado cada pixel e sua relação com outros pixels para formar um novo pixel na

imagem processada. Por isso, antes de serem apresentadas as técnicas mais comuns, deve ser conhecida a relação existente entre os pixels.

2.2.1 Vizinhança

Vizinhança de pixels diz respeito à quais pixels no entorno do pixel em questão são considerados vizinhos do mesmo. Considerando a representação do pixel por um quadrado, define-se que vizinhos são aqueles "quadrados" que possuem pelo menos um ponto em comum. Dado um pixel *p(x,y),* são considerados vizinhos horizontais os pixels que se encontram na mesma linha, uma coluna antes e uma coluna depois de *p*, isto é, *p(x-1,y)* e *p(x+1,y)*. Da mesma forma, são considerados vizinhos verticais aqueles que estão na mesma coluna, a uma linha acima e uma linha abaixo de *p*, ou seja, *p(x,y-1)* e *p(x,y+1)*. Os vizinhos horizontais e verticais são aqueles pixels que possuem um lado em comum com o pixel analisado. Quando falamos em vizinhança de 4 pixels- $N_4(p)$ -, nos referimos aos quatro vizinhos horizontais e verticais do pixel.

Ao redor do pixel, temos ainda aqueles pixels que possuem apenas um vértice em comum com o quadrado que representa o pixel em questão. São os pixels *p(x-1,y-1), p(x+1,y-1), p(x-1,y+1)* e *p(x+1,y+1)*, e são considerados os vizinhos diagonais do pixel *p* - $N_D(p)$. A união entre a vizinhança de 4 e a vizinhança diagonal resulta na vizinhança de 8 – $N_8(p)$ – que engloba todos os pixels ao redor do pixel *p*. A Figura 2.6 exemplifica a vizinhança do pixel *P*.

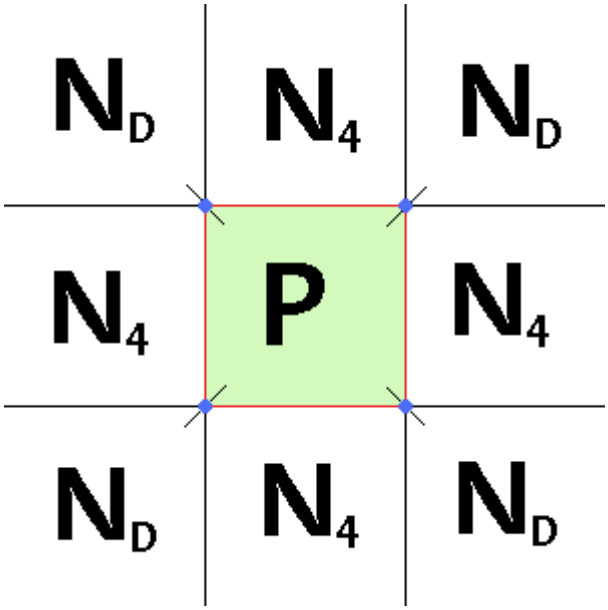

Figura 2.6 - Vizinhança entre pixels. Os pixels que possuem apenas um ponto em comum são vizinhos diagonais, os que possuem dois são vizinhos de 4. Os que possuem pelo menos um ponto são vizinhos de 8

2.2.2 Componentes conexos

Dois pixels estão conectados se são vizinhos entre si e se atendem a uma regra de conectividade definida, tipicamente uma restrição à diferença entre suas luminosidades. Um grupo conexo é definido como sendo um conjunto de pixels de uma imagem onde todo pixel está conectado a pelo menos um pixel deste conjunto. Um grupo conexo é representado em uma imagem por um rótulo que representa o número deste grupo. Por ser composto de pixels vizinhos entre si, o tipo de vizinhança adotada impacta no resultado obtido com este método. A Figura 2.7 mostra um exemplo de grupos conexos e a influência da vizinhança no resultado. A primeira figura mostra a imagem binária original. Nesse caso a regra de conectividade é o valor do pixel ser diferente de zero. A segunda e a terceira imagens são grupos conexos obtidos com conectividade de 8 e 4 pixels, respectivamente. Cada grupo é representado por uma cor diferente para melhor visualização sendo que a cor branca representa o fundo, ou pixels que não pertencem a nenhum grupo. Nota-se que utilizando uma vizinhança de 4 foram encontrados dois

grupos, enquanto considerando uma vizinhança de 8 os dois grupos são conectados por um pixel na diagonal que une esses dois grupos.

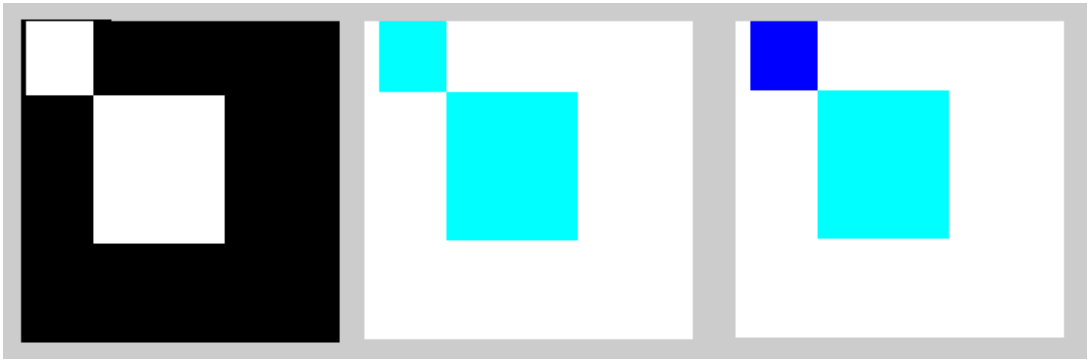

Figura 2.7 - Exemplo de grupos conexos. A primeira imagem é a imagem original, a segunda representa os grupos em conectividade de 8 e a terceira os grupos em conectividade de 4

2.3 Operações de melhoria da imagem

Existem diversas operações que podem ser aplicadas sobre a imagem de forma a torná-la visualmente melhor, o que pode ajudar o desempenho de um algoritmo de segmentação quando utilizado corretamente. Essas operações envolvem desde mudanças no tamanho quanto a melhorias no contraste, na luminosidade ou na aplicação de filtros para diminuir ruídos ou borrões na imagem. Essas técnicas serão apresentadas a seguir.

2.3.1 Ampliação e redução

Muitas vezes é necessário ampliar uma imagem, seja para melhor visualização ou para facilitar o processamento da imagem. Outras vezes é necessário reduzir uma imagem, ou por ela ser muito grande e não caber na tela, ou por conter informação em excesso que não é necessária para a aplicação, podendo trabalhar com imagens menores ganhando em tempo de execução do algoritmo. As técnicas de ampliar e reduzir imagens são bastante parecidas, sendo que uma cria novos pixels na imagem e a outra

remove. A chave destas técnicas está na escolha do valor do novo pixel na imagem ampliada, ou daquele que substituirá um conjunto de pixels na imagem reduzida.

Como um exemplo imagine uma imagem de 256x256 pixels. Para ampliá-la em um fator de 1.5 (384 x 384 pixels) a forma mais simples é dividi-la em uma grade de 384x384 pixels, o que resulta em uma divisão menor que um pixel. Escolhe-se então o pixel mais próximo na imagem original deste pixel da imagem dividida, e o pixel novo assume o mesmo valor de luminância. Essa técnica é conhecida como vizinho mais próximo (*nearest neighbor*) e é a mais rápida e simples existente, porém cria um efeito quadriculado na imagem pois apenas replica os valores dos pixels.

Técnicas mais sofisticadas levam em consideração mais de um vizinho mais próximo para calcular o valor do novo pixel. Utiliza a interpolação bi-linear realizando a média ponderada dos 4 vizinhos das diagonais para calcular o valor do novo pixel. A Figura 2.8 mostra o resultado das duas técnicas para uma mesma imagem aumentada em três vezes. Enquanto a técnica do vizinho mais próximo tende a deixar a imagem mais quadriculada, deixando alguns pixels “maiores” que outros quando o fator de amplificação não é inteiro, a técnica da interpolação bi-linear deixa as bordas mais suaves, embaçando a imagem. A técnica bi-linear resulta em uma imagem melhor de se visualizar, porém é mais custosa computacionalmente que a do vizinho mais próximo.

Figura 2.8 - Técnicas de ampliação da imagem: a) a imagem original, b) imagem ampliada em 3x por *nearest neighbor*, c) imagem ampliada em 3x por *bilinear interpolation*

2.3.2 Transformações em potência

Transformações em potência são utilizadas para corrigir a luminosidade de uma imagem e seguem uma fórmula bastante simples descrita na equação 2.2

$$s = cr^{\gamma} \quad \textbf{2.2}$$

Nessa equação temos que s é o valor do pixel após a transformação, c é uma constante positiva arbitrária, r é o valor atual do pixel e γ é uma constante positiva que define a curva de transformação.

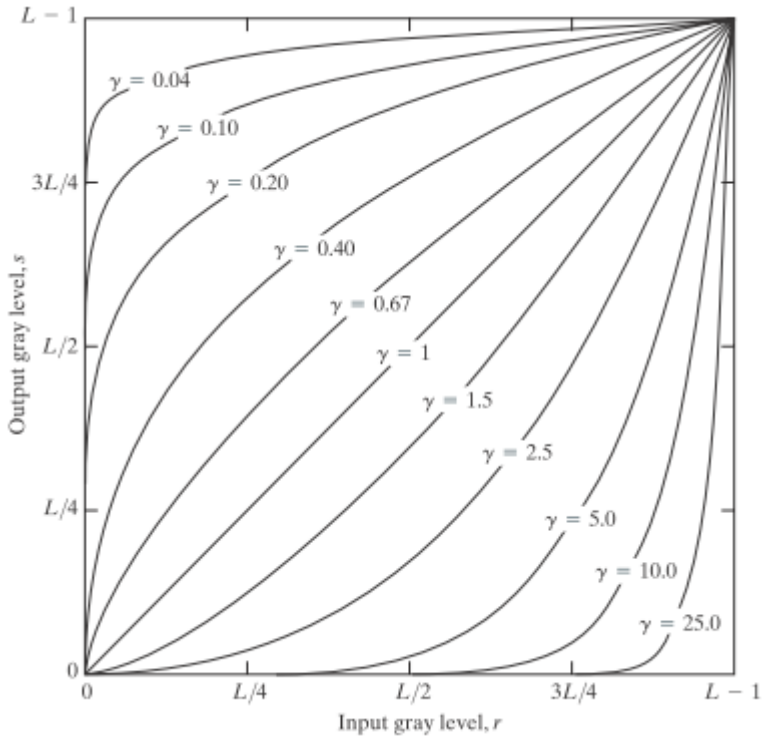

Figura 2.9 - Curva da transformação de potência para diferentes valores de γ (GONZALES e WOODS, 2002)

A Figura 2.9 mostra as curvas de transformações para diversos valores de γ. Observa-se que valores de $\gamma < 1$ irão mapear a maioria dos pontos em uma região mais clara da imagem, enquanto que valores de $\gamma > 1$ mapeiam os pixels em pontos mais escuros.

Diversos aparelhos de captura, exibição e impressão de imagens têm resposta de luminosidade seguindo uma exponencial. O processo para corrigir essa resposta exponencial está na transformação de potência, com a escolha certa do parâmetro γ, daí essa transformação também se tornou conhecida como correção de gama (GONZALES e WOODS, 2002). Por exemplo, dispositivos de exibição de imagens que funcionam por meio de tubos de raios catódicos (CRT) têm uma resposta de intensidade/tensão elétrica seguindo uma exponencial de expoente entre 1.8 e 2.5, que significa que as imagens exibidas são mais escuras que a imagem real. Para corrigir isso, uma correção de gama deve ser feita com um fator de $1/\gamma$ de forma a anular a distorção indesejada.

2.3.3 Equalização de histograma

O histograma de uma imagem consiste em uma função discreta $h(r_k) = n_k$, onde r_k é o k-ésimo nível de cinza e n_k é o número de pixels na imagem que possuem esse nível de cinza. O histograma pode ser considerado como uma distribuição da probabilidade de cada nível de cinza ocorrer. Um histograma que possua uma distribuição muito estreita significa que possui muita concentração de luminosidades próximas, ou seja, um baixo contraste. Seguindo o mesmo raciocínio, um histograma largo possui os tons de cinza mais bem distribuídos, logo possui um contraste melhor.

A equalização de histograma busca maximizar o contraste da imagem transformando o histograma da imagem em uma distribuição uniforme ocupando todo o espaço

disponível(GONZALES e WOODS, 2002). Devido à natureza discreta das imagens digitais, não é possível redistribuir uniformemente todos os níveis de cinza de uma imagem, causando picos em alguns pontos do histograma. Outra técnica mais simples computacionalmente, porém com um resultado inferior, é o alongamento de histograma, que consiste apenas em trazer o valor mínimo de luminância para zero, e o valor máximo para 255, distribuindo proporcionalmente os pontos intermediários. Assim, para se gerar o histograma alongado h' a partir de um histograma h, basta aplicar a Equação 2.3.

$$h' = (h - \min(h)) * \frac{255}{\max(h) - \min(h)} \qquad \textbf{2.3}$$

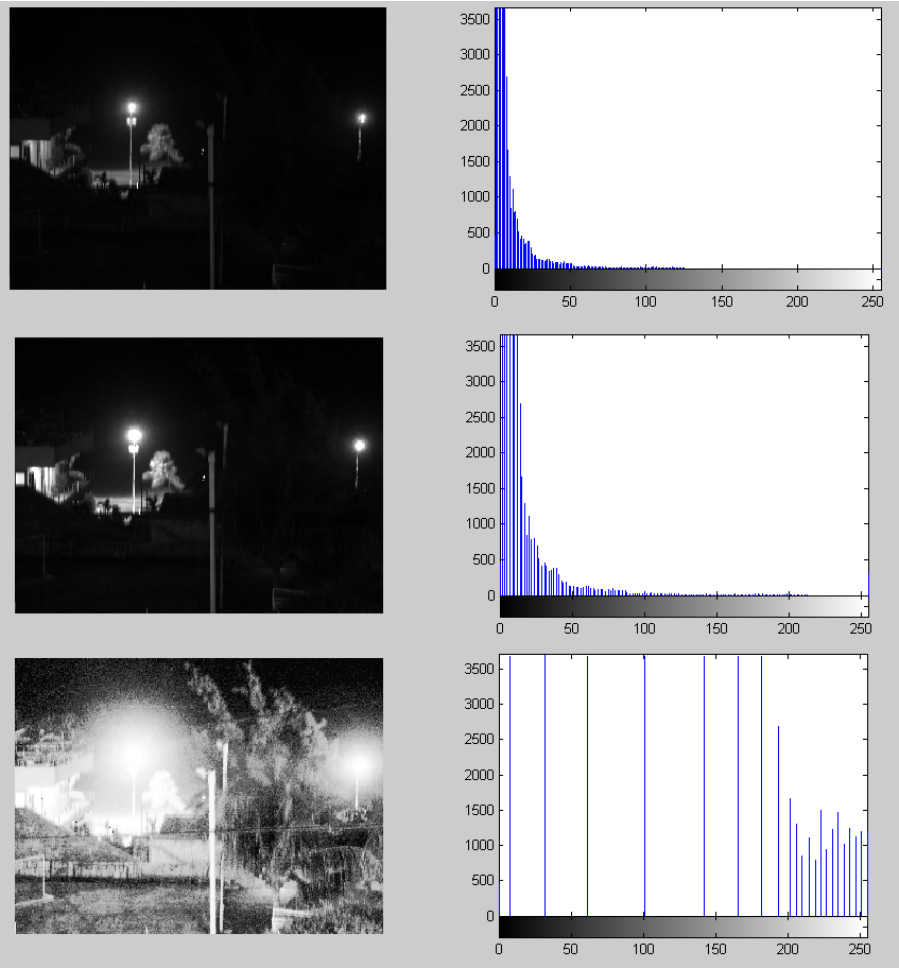

Figura 2.10 - Exemplos de transformações do histograma: a) imagem original; b) histograma alongado; c) histograma equalizado

O alongamento de histograma é ineficaz quando há pontos próximos aos limites do pixel (0 e 255), sendo necessário outro tipo de tratamento na imagem. A equalização de histograma é uma poderosa ferramenta para aumentar o contraste em uma imagem, mas deve ser utilizada com cuidado pois pode estragar algumas imagens. A Figura 2.10 mostra exemplos das técnicas apresentadas de manipulação de histograma. A primeira imagem mostra a imagem original e seu histograma. Observa-se que é uma imagem escura, caracterizada por um forte pico na região inicial do histograma e pouca distribuição nas regiões mais claras. A segunda imagem apresenta o resultado do alongamento de histograma. Observa-se que o espaçamento entre os picos aumentou e a imagem resultante está um pouco mais clara que a original. A terceira imagem é o resultado da equalização de histograma e é possível ver que a imagem ficou mais clara, porém vários objetos perderam a definição e a imagem ficou degradada.

2.4 Operações morfológicas

Operadores morfológicos são aqueles que aplicados à imagem alteram a sua forma. Embora possam ser utilizados em imagens coloridas ou em tons de cinza, sua aplicação mais usual é em imagens binárias e as operações básicas serão descritas nessa seção. As operações mais conhecidas são: dilatação, erosão, abertura e fechamento.

Cada operação morfológica é feita utilizando um elemento estruturante que pode possuir diversas formas e tamanhos, dependendo da aplicação desejada. As operações são feitas realizando a convolução entre a imagem e o elemento estruturante. Alguns exemplos de elementos estruturantes são exibidos na forma de matriz na Figura 2.11.

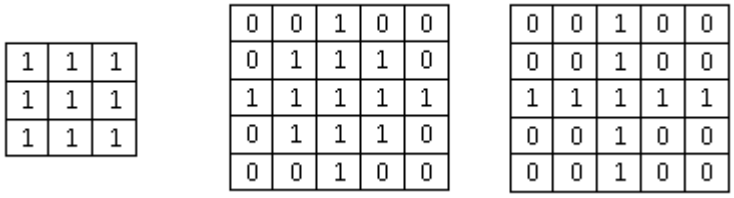

Figura 2.11 - Exemplos de elementos estruturantes: a) quadrado, b) diamante, c) cruz

2.4.1 Dilatação

A operação de dilatação pode ser descrita conforme a equação 2.4. Dada uma imagem no espaço Z^2, a dilatação da imagem A utilizando o elemento estruturante B será dada pelo conjunto de pontos z onde a reflexão de B ($\widehat{B}$) translada em z se intercepta com algum ponto de A(GONZALES e WOODS, 2002). Com isso temos como resultado uma imagem com sua região de fronteira expandida de acordo com o elemento estruturante utilizado.

$$A \oplus B = \{z | \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A\} \qquad \textbf{2.4}$$

A dilatação é utilizada para conectar pontos próximos e fechar buracos na imagem, que podem ser conseqüência de um processamento anterior, como uma binarização com um limite mal especificado. A Figura 2.12 exemplifica a aplicação do operador de dilatação utilizando um elemento estruturante quadrado de 3x3, mostrado na Figura 2.11. Note que na imagem original (esquerda) havia uma falha no traço horizontal da letra “A” que foi corrigido pela dilatação. Ao mesmo tempo, todos os traços da imagem ficaram mais grossos. Como conseqüência do aumento da região de fronteira, alguns objetos que antes não estavam conectados podem passar a fazer parte do mesmo grupo conexo, o que nem sempre pode ser o desejado. Por isso a dilatação deve ser utilizada com cuidado.

Figura 2.12 - Dilatação de uma imagem binária com um elemento estruturante quadrado de 3x3. Em destaque as áreas com falhas que foram corrigidas pela dilatação.

2.4.2 Erosão

Dado uma imagem A e um elemento estruturante B, a erosão de A por B é definida nos termos da equação 2.5 (GONZALES e WOODS, 2002). Em outras palavras, a erosão de A por B consiste no conjunto de pontos z em Z^2 tal que o conjunto B transladado de z esteja contido em A. Essa operação resulta em uma imagem mais fina, com uma região de fronteira menor.

$$A \ominus B = \{z|(B)_z \subseteq A\} \quad \textbf{2.5}$$

Este tipo de operação é utilizado para desconectar objetos e eliminar detalhes irrelevantes em questão de tamanho. A Figura 2.13 mostra a aplicação do operador de erosão em uma versão modificada da imagem do exemplo anterior. Nesta imagem, a letra “r” foi deslocada para a esquerda de forma a ficar conectada à letra “A”.

Foi utilizado o mesmo elemento estruturante quadrado de 3x3. Observa-se que após a erosão as letras foram desconectadas, porém a operação causou estragos na imagem, desconectando várias partes das letras. Isso ocorreu devido ao fato do elemento estruturante ser de um tamanho compatível com a largura das linhas mais finas das letras, assim não houve nenhum ponto nestas linhas onde todo o conjunto B estivesse contido na imagem A. Por esse motivo, a escolha de usar ou não a erosão, bem como a escolha do elemento estruturante deve ser feita com cuidado.

Figura 2.13- Operação de erosão aplicada a uma imagem.

2.4.3 Abertura e Fechamento

A dilatação é capaz de fechar falhas na imagem deixando seus limites maiores, engrossando os traços, e conseqüentemente arriscando conectar partes indesejadas, enquanto a erosão faz o oposto, reduzindo a região de fronteira e desconectando elementos. Conhecendo isso, podemos agrupar essas duas transformações de forma que a segunda operação tente anular o efeito negativo da primeira, obtendo assim um resultado com uma menor distorção da imagem original. Define-se como Abertura como uma transformação de erosão seguida de uma de dilatação, e Fechamento como uma transformação de dilatação seguida por uma erosão (GONZALES e WOODS, 2002).

A transformação de abertura busca desconectar elementos e a transformação de fechamento busca fechar buracos na imagem, corrigindo as imperfeições causadas pela binarização ou inerentes à própria figura. A Figura 2.14 mostra a aplicação destas transformações à imagem do exemplo anterior. Comparado aos resultados das transformações simples, estas transformações apresentam resultados mais próximos da imagem original

Figura 2.14 - Aplicação da transformação de abertura (meio) e fechamento (direita)

3 Estado da arte da Segmentação

3.1 Reconhecimento Óptico de Caracteres - OCR

O estudo de meios de localizar e reconhecer textos em imagens, e traduzi-los para linguagem escrita começou há muitas décadas. Na década de 50, David H. Shepard desenvolveu a primeira máquina comercial capaz de converter documentos impressos em linguagem de máquina. Essa foi a primeira aplicação comercial de OCR, altamente limitada pela tecnologia da época, que era capaz de ler apenas caracteres tipográficos de uma fonte, de cor preta, alinhados horizontalmente, igualmente espaçados e com fundo em folha de papel branca. Já na década de 60, as aplicações de OCR já estavam mais difundidas, variando desde leituras de números seriais de cupons até códigos postais de correio. Na década de 70, começou a se ampliar a aplicação de OCR para auxílio de deficientes visuais, com o intuito de fazer um computador ler em voz alta um texto para um cego. Nesta época também teve início as pesquisas sobre sistemas de OCR capazes de ler várias fontes, de forma a tornar o sistema mais robusto para diferentes textos. Até então apenas textos impressos eram considerados, embora a idealização do OCR sempre fosse de um sistema capaz de reproduzir a capacidade humana de leitura, que é altamente robusta, capaz de ler textos independente de fonte e tamanho e nas mais variadas condições de ambiente e orientação.

Desde então, o avanço dos computadores alavancou as pesquisas nesta área, gerando novas demandas e o desenvolvimento de novas técnicas para melhorar a robustez e a capacidade de generalização do programa. As aplicações passaram de simples leitores de documentos em formato padronizado para leitura de formulários, caracteres manuscritos, leitura de placas de automóveis, placas de sinalização,

documentos multicoloridos, capas de mídias, sites, documentos digitalizados com fontes diversas, e mais recentemente, fotografias de cenas reais contendo textos.

Os trabalhos sobre OCR se dividem em duas vertentes: Caracteres manuscritos e caracteres tipográficos. Embora o conhecimento utilizado em um tipo de aplicação muitas vezes possa ser aproveitado em outro, os trabalhos feitos sobre caracteres manuscritos não fazem parte do escopo desta pesquisa e não serão contemplados neste capítulo de revisão bibliográfica. Por caracteres tipográficos neste trabalho, entende-se todo caractere não-cursivo gerado de forma artificial por uma máquina (seja impresso por uma impressora, datilografado ou imposto artificialmente em uma imagem), e que possui fontes bem definidas, embora possa haver diversas fontes em uma mesma imagem e até em uma mesma palavra.

Dentro da linha de pesquisa de caracteres tipográficos há trabalhos feitos em cima de imagens estáticas e trabalhos em vídeos. Este último possui uma característica particular que é o eixo do tempo, ou seja, quadros sucessivos podem ser utilizados para refinar o resultado, embora por outro lado imagens estáticas podem ser obtidas em uma resolução normalmente superior à de um vídeo.

Embora existam muitos trabalhos desenvolvidos em OCR e segmentação de caracteres, falta na literatura revisões atualizadas sobre as abordagens de forma geral. O mais comum de se encontrar são *surveys* sobre aplicações específicas (por exemplo, LPR ou indexação de vídeo), talvez pela dificuldade de se classificar os tipos de trabalho existentes ou pela dificuldade de se criar uma comparação de resultados entre eles, pois a grande maioria destes trabalhos busca a resolução de problemas específicos e utilizam um banco de imagens próprio, tornando imprecisa a comparação de resultado.

Não é fácil desenvolver um sistema genérico de extração de informação textual, pois há muitas possíveis fontes de interferência presentes principalmente quando se extrai texto de fundo sombreado ou texturizado, de imagens de baixo contraste ou complexas, bem como de imagens que têm variações no tamanho da fonte, estilo, cor, orientação e alinhamento (JUNG, KIM e JAIN, 2004). Imagens complexas (imagens de cenas reais contendo texto – *scene text*) têm seu texto normalmente afetado por variações no ambiente e nos parâmetros da câmera, como iluminação, foco, movimento, distorções da lente e da formação da imagem, ruído, perspectiva, entre outros. Jung ET AL divide o problema de extrair texto de imagens em cinco etapas:

1. **Detecção:** Consiste no processo de identificar se a imagem contém ou não texto. É uma etapa necessária para se evitar executar todo o processamento pesado de localização até o reconhecimento. É essencial para trabalhos feitos sobre vídeos contínuos, visto que a quantidade de imagens processadas é considerável.
2. **Localização:** É a etapa de identificação da localização das regiões que contém caracteres dentro da imagem.
3. **Rastreamento:** É uma etapa utilizada apenas em localização em vídeos e que utiliza a informação da localização do texto de um quadro no quadro seguinte, visando à redução do tempo de processamento da localização.
4. **Extração:** Uma vez localizados os caracteres, eles precisam ser extraídos do fundo e tratados de forma a melhorar o desempenho do reconhecimento. Esta etapa também é conhecida na literatura como segmentação, embora a

localização também seja considerada como uma parte da segmentação de caracteres.

5. **Reconhecimento:** Nesta etapa o texto tem suas características extraídas e com base nelas é traduzido para texto armazenado no computador.

Alguns trabalhos focam apenas na parte de extração, assumindo que já estão recebendo uma imagem com texto e devem segmentá-la em caracteres para um OCR. Outros já focam nas etapas de localização e extração, e às vezes reconhecimento. Poucos trabalhos são focados na parte de detecção (essa etapa pode ser substituída pela localização, desde que sua capacidade de rejeição de falsos positivos seja alta), e outros poucos tratam da etapa de rastreio.

Para se localizar textos em imagens, os trabalhos normalmente adotam pelo menos uma de três abordagens, cada uma assume como verdadeiro uma característica para todos os textos (CHEN, BOURLARD e THIRAN, 2001). As abordagens comuns são:

1. **Abordagem por contraste:** Assume que todo o caractere, para ser legível, deve possuir um contraste em relação ao fundo onde este texto está inserido. As soluções desenvolvidas por essa abordagem normalmente utilizam detecção de bordas e operadores morfológicos para localizar os textos.
2. **Abordagem por regiões:** Assume que cada caractere possui apenas uma cor (monocromaticidade). Nessa abordagem, detectam-se os caracteres utilizando binarizações e identificação de componentes conexos, assumindo que cada caractere pertencerá apenas a um grupo ao final da execução.
3. **Abordagem por texturas:** Assume que as regiões que contém texto possuem características únicas de textura, podendo ser utilizado para diferenciar de

regiões sem texto. Regiões de texto possuem freqüência e informações de orientação, podendo então ser tratada como uma textura distinta (WU, MANMATHA e RISEMAN, 1997). Utiliza aplicações de filtros, comprimento de transição (CHEN e WU, 2009) e técnicas de aprendizado de máquina para gerar classificadores capazes de identificar essas características nas regiões. Técnicas de casamento de padrões também podem ser consideradas nessa categoria.

Cada abordagem possui vantagens e desvantagens, sendo muito comum encontrar trabalhos que utilizam mais de uma abordagem, sendo difícil classificá-los. Devido a essa dificuldade de classificação, os trabalhos apresentados serão divididos por propósito e não por técnicas empregadas.

Em seu trabalho, Chen (2001) busca extrair textos sobrepostos em imagens e vídeos (legendas e afins), e para isso detecta separadamente bordas horizontais e verticais e aplica um operador morfológico de dilatação com elementos estruturantes distintos para cada tipo de borda. As regiões que existirem em ambas as figuras de bordas dilatadas serão consideradas regiões candidatas, passando essas regiões por uma análise de textura, feita através de um classificador SVM (*Suport Vector Machine*). A idéia por trás do algoritmo é que regiões de texto possuem bordas próximas e dilatação é feita para unir essas bordas em uma única região. Se essa região existir nas duas imagens dilatadas, ou seja, for composta por várias bordas horizontais e verticais, é candidata a ser um texto. O classificador é utilizado apenas para remover eventuais pedaços da imagem que tenham as mesmas características de bordas, como grades e escadas. A técnica funciona bem para o caso de texto inserido artificialmente na

imagem, pois este é feito para ser legível dentro da imagem em questão e não sofre influência da iluminação, sombras, ângulo da câmera e outros fatores. Em cenas reais, onde temos todos esses tipos de ruídos, o algoritmo não seria tão eficiente. Em outro trabalho (CHEN, ODOBEZ e THIRAN, 2004), o autor refina o método de seleção de regiões candidatas e desenvolve um novo classificador, fazendo uma comparação entre classificadores SVM e redes neurais MLP para quatro diferentes conjuntos de características, concluindo que o SVM se comporta melhor em todos os casos.

3.2 Segmentação de documentos

A segmentação de documentos é um tema bastante abordado e diversos trabalhos foram escritos sobre o tema. Segmentar documento é relativamente mais simples do que imagens complexas, pois o documento ou já foi gerado no computador (para aplicações de leitor de telas para cegos) ou foi digitalizado através de um *scanner* que gera a imagem do documento em um ambiente bastante controlado, sofrendo pouca influência de iluminação, embora possa haver distorções de orientação e perspectiva (ao se digitalizar um livro, por exemplo, as letras próximas à margem interna do livro podem sofrer essa distorção). Além disso, documentos têm características espaciais (divisão em linhas, espaçamento bem definido e fontes constantes) que facilitam sua segmentação. Os desafios nesta área se encontram basicamente em segmentar documentos que contenham imagens, caracteres sobrepostos ou conectados, ser independente de idioma e robustez a ruídos.

Um método interessante foi desenvolvido para segmentar caracteres sobrepostos ou conectados (LEE, LEE e PARK, 1996). Muitas vezes a simples projeção vertical, utilizada em muitos trabalhos para dividir os caracteres, não é capaz de separar os caracteres

sobrepostos, degenerando os caracteres. Ele propôs uma divisão não linear utilizando uma técnica de busca de menor caminho para seccionar os caracteres agrupados. Um reconhecedor é utilizado junto ao sistema levando em consideração as subdivisões adjacentes, na tentativa de identificar e agrupar pedaços de caracteres que foram separados.

Tsujimoto e Asada (1992) descrevem em seu trabalho os principais componentes de um leitor de documentos. O trabalho é voltado principalmente para jornais e documentos divididos em colunas. Ele divide os documentos em blocos formados por componentes conexos adjacentes cujo espaço em branco entre eles seja menor que um valor pré-definido, e em seguida classificam esses blocos como linhas de texto, figuras, etc. para depois agrupar os blocos de texto que formam parágrafos de uma mesma coluna. A maior dificuldade deste trabalho está mais no estudo do agrupamento dos parágrafos de forma lógica do que na localização do texto, reforçando a idéia da simplicidade de se segmentar texto de documentos por conterem fundo homogêneo.

Outro trabalho buscou fazer a identificação de texto em documentos através da textura utilizando filtros de Gabor e Wavelets (NOURBAKHSH, PATI e RAMAKRISHNAN, 2006). Os autores mostram a invariância do seu método à orientação do texto. Pelos resultados exibidos, o sistema é capaz de discriminar texto e não texto quase na totalidade dos casos, errando apenas pequenas regiões e símbolos.

Algumas vezes, documentos digitalizados não possuem fundo homogêneo, e sim um fundo multicolorido, com formas e às vezes texto sobre texto (Figura 3.1), como é o caso de capas de livros e mídias, panfletos, entre outros. Neste caso o documento é conhecido como documento complexo. Para esse tipo de problema, Chen e Wu (2009)

desenvolveram um método de divisão da imagem em planos, separando-os através de binarização multinível através da análise do discriminante da imagem. O método identifica regiões homogêneas através de diferentes níveis de binarização, para então tratar cada plano separadamente, identificando as áreas de texto, aplicando melhorias nessas regiões e agrupando-as de volta na imagem filtrada apenas com os textos.

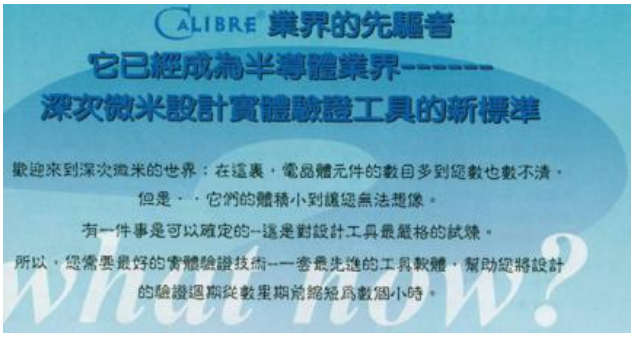

Figura 3.1 - Imagem de um documento complexo contendo texto em várias camadas

3.3 Reconhecimento de placas de licenciamento de veículos

Outra área de aplicação de OCRs que possui uma grande quantidade de trabalhos publicados é a de LPR. O reconhecimento de placas de automóveis é uma área em ascensão e de amplo interesse comercial. Pode ser considerado como um caso de imagem complexa, pois está sujeita a todas as interferências de uma imagem de cena real, porém possui a vantagem de se saber que tipo de texto se está procurando. As placas de automóveis são padronizadas em cada país, contêm fundo de cor bem específica e na maioria das vezes possuem um número definido de caracteres. No Brasil, por exemplo, as placas são cinza com sete caracteres pretos para veículos de passeio, e vermelhas com caracteres brancos para veículos de serviço, além de outras variações mais raras de serem encontradas. Várias são as informações que podem ser utilizadas para ajudar a localizar a placa. Há trabalhos que limitam a área de busca a apenas um

pedaço da imagem, pois as imagens aquisitadas pelo sistema seguem um padrão, então estatisticamente a placa só pode se encontrar naquela região (GUINGO, STIEBLER e THOMÉ, 2004). Neste trabalho, os autores utilizam um filtro de alta freqüência de forma a identificar pontos que apresentem um determinado contraste, o que inclui as bordas dos caracteres e dependendo da cor do pára-choque, a borda da placa. Então se busca a região com o tamanho pré-definido da placa que possua a maior densidade de pontos.

Shapiro ET AL (2004) também utiliza a detecção de bordas para localizar os caracteres de uma placa de veículo. Um filtro horizontal de tamanho M*N (M>N) é aplicado à imagem de bordas e uma imagem formada por 80% do valor do pixel da imagem original, utilizando a imagem filtrada como máscara, o que resulta em uma área clara na região da placa. Uma projeção vertical da imagem é feita, sendo a placa localizada no pico de maior intensidade. Após isso corrige sua inclinação detectada aplicando a transformada de Hough(GONZALES e WOODS, 2002, p. 587-597), e refina-se a segmentação no sentido horizontal. Ambos os sistemas utilizam apenas do alto número de bordas que regiões de texto produzem, porém isso não é suficiente para caracterizar uma placa, podendo outras regiões de texto serem detectadas no lugar da placa, ou regiões texturizadas que também produzem um grande número de bordas, como faróis e asfalto. Uma vez tendo a placa localizada e corrigida a sua orientação, a maioria dos trabalhos propõe a segmentação dos caracteres através de uma ou mais projeções verticais para dividir os caracteres (MARTINSKY, 2007)(ZHANG e ZHANG, 2003).

Os sistemas de LPR citados acima possuem a desvantagem de serem suscetíveis a outros textos presentes na imagem. Devido ao fato de apenas uma região candidata ser válida em toda a imagem, os sistemas podem escolher de forma errada

comprometendo o resultado. Nenhum trabalho estudado utiliza um leitor robusto para identificar todas as letras existentes na imagem e analisar a semântica da seqüência de caracteres em busca da placa.

3.4 OCR em vídeos

A indexação de vídeos através da detecção de textos sobrepostos nos vídeos (legenda, créditos e informações de forma geral) é outra aplicação de OCR de crescente interesse no meio acadêmico. Textos artificialmente inseridos em vídeos possuem a característica de estar sempre no mesmo plano da câmera, além de não sofrer influência de iluminação e o ruído existente provém apenas da compressão do vídeo. Isso resulta em uma série de vantagens na localização deste texto em relação às imagens de cena real. Lienhart e Stuber (1996) assumem, baseado na observação, que os caracteres artificiais em vídeos possuem as seguintes características:

- Caracteres são monocromáticos, apenas uma pequena parte não possui essa característica e não são de interesse para o trabalho.
- Caracteres são rígidos, ou seja, não sofrem alteração de forma, tamanho e orientação ao longo dos quadros. Novamente uma pequena parcela fora do escopo de seu trabalho pode ser encontrada.
- Caracteres têm restrições quanto ao seu tamanho. Não terão caracteres tão grandes quanto um quadro e nem tão pequenos que o tornem ilegíveis.
- Caracteres em vídeo são estacionários ou possuem um movimento linear.
- Caracteres contrastam com o fundo, pois o texto inserido artificialmente é gerado para ser lido de forma fácil, embora devido à compressão do vídeo isso possa não ser inteiramente verdade sempre.

- O mesmo caractere aparece em vários quadros consecutivos.

Em seu trabalho, Lienhart utiliza uma técnica de segmentação baseada em cor denominada *split & merge* que consiste de sucessivas divisões da imagem em quadrantes seguindo um determinado critério, e depois sucessivos agrupamentos desses quadrantes, formando grupos conexos baseado na cor dos objetos. Os grupos são então eliminados de acordo com suas dimensões e os quadros consecutivos são utilizados para refinar a localização. A imagem passa então por uma detecção de bordas fortes (aplicação de um filtro de Canny (CANNY, 1986) com um limiar alto) seguida por uma dilatação, que devem interceptar com as regiões candidatas detectadas. Mais uma heurística é utilizada para eliminar grupos e aqueles que sobram são considerados grupos de texto.

Sato ET AL (1999) aplica um filtro diferencial 3x3 nos quadros para obter as bordas verticais e aplica um filtro de suavização para eliminar bordas estranhas e conectar bordas de caracteres que não se conectaram. As regiões agrupadas são identificadas como passiveis de conter texto se a região candidata: i) possui tamanho maior que 70 pixels, ii) fator de preenchimento maior que 45%, iii) relação de aspecto maior que 0.75. Uma melhoria na imagem é feita utilizando vários quadros para melhorar a resolução. Caracteres são extraídos das regiões escolhidas aplicando um filtro de correlação para encontrar, independentemente, linhas a 0, 90, 45 e -45 graus. Essas imagens filtradas são agrupadas e binarizadas, resultando em uma imagem dos caracteres sem o plano de fundo. A partir daí, a projeção vertical é utilizada para selecionar pontos de corte e um reconhecedor é integrado à segmentação para definir quais cortes pertencem a um mesmo caractere.

Outra abordagem utiliza um método de binarização local adaptativa para fazer uma primeira limpeza na imagem, tentando deixar apenas áreas com texto nos quadros do vídeo (LYU, SONG e CAI, 2005). Os autores propuseram um método de binarização adaptativa que leva em consideração a complexidade do plano de fundo. Uma série de parâmetros são definidos para o funcionamento do algoritmo, que utiliza oito características do caractere, quatro independente de idioma, e quatro que variam entre os idiomas inglês e chinês estudados no artigo.

Embora as imagens utilizadas nestes leitores de vídeos sejam cenas reais, esses trabalhos não se enquadram na categoria de leitores robustos de imagens complexas devido a suas várias restrições impostas quanto ao tipo de texto esperado, embora o conhecimento utilizado possa ser reaproveitado em leitores robustos.

3.5 Leitores robustos

Os trabalhos classificados nessa sessão são aqueles que se propõem a realizar a leitura de textos em imagens complexas, aplicando pouca ou nenhuma restrição ao tipo de caractere ou conjunto de caracteres aceito pelo sistema.

LeBourgeis (1997) utilizou uma abordagem por textura utilizando o comprimento de transição (*run length*). *Run lengths* são definidos pela espessura dos traços dos caracteres e pelo espaçamento entre traços, caracteres e palavras. Um filtro foi desenvolvido para computar o comprimento de transição, que uma vez aplicado à imagem resulta em uma nova imagem com as regiões de texto destacadas. Para segmentar os caracteres aplica-se uma binarização automática e separa caracteres conectados utilizando a informação dos níveis de cinza.

Outro trabalho utiliza nove filtros derivativos de segunda ordem de gaussianas para identificar as texturas das regiões de texto na imagem (WU, MANMATHA e RISEMAN, 1997). As áreas de texto normalmente têm uma resposta maior a esses filtros. Uma vez identificadas as regiões, elas são agrupadas e são utilizados operadores morfológicos para corrigir buracos no texto. Para a seleção das regiões, bordas significativas são extraídas da imagem e depois filtradas para eliminar bordas que não pertençam ao texto. Esses traços são agrupados em regiões e novamente filtrados para eliminar mais falsos positivos. Após isso, as regiões restantes passam por uma etapa de alongamento, na tentativa de recuperar caracteres que não foram agrupados nas etapas anteriores.

Clark e Mirmehdi (2000) também utilizaram a abordagem por textura para identificar regiões de texto em imagens complexas. Os autores utilizam cinco medidas extraídas da imagem. Essas medidas são utilizadas como entrada de uma rede neural MLP de três camadas que classifica cada pixel como texto ou não texto. O sistema é capaz de identificar regiões de texto mesmo quando estes não são legíveis.

Yuille e Chen (2004) fizeram uma análise estatística das imagens utilizadas para definir quais características são relevantes para identificar textos. Então foram criados classificadores fracos utilizando as probabilidades conjuntas das características dentro e fora do texto. Esses classificadores fracos foram utilizados como entradas para um AdaBoost (HAYKIN, 1999) para treinar um classificador forte. Na prática foi feito uma cascata com quatro classificadores fortes contendo 79 características no total. Uma melhoria na imagem é feita para as imagens classificadas e um OCR comercial é utilizado para validar o trabalho. O trabalho utiliza resoluções relativamente grandes (2048x1536 pixels) e o algoritmo é executado em menos de 3 segundos. O primeiro conjunto de

características é baseado na observação das derivadas em x e y da imagem dos textos (horizontal e vertical). Em x a derivada tende a ser larga na região central enquanto em y são largas no topo e embaixo, e pequenas no centro. Com essa observação, a imagem é dividida em blocos para as derivadas x e y e a média e desvio padrão desses blocos são utilizados como características para o primeiro classificador. A segunda classe de características é baseada nos histogramas de intensidade, direção e intensidade do gradiente. Idealmente o histograma de uma região de texto deveria ter dois picos correspondentes ao fundo e ao texto, porém na prática não é isso que é observado. Utilizando um histograma conjunto de intensidade e derivada de intensidade é possível estimar as médias de intensidades do texto e do fundo. A Terceira e última classe é baseada na detecção de bordas, binarizando o gradiente de intensidades, seguido de junção de bordas. Por serem mais computacionalmente custosas, são utilizadas por ultimo apenas nas regiões aceitas pelos outros classificadores. Essas características utilizam a contagem de bordas estendida na imagem. O classificador forte foi então criado utilizando treinamento padrão de AdaBoost combinado com a abordagem de cascata de Viola e Jones que utilize pesos assimétricos. O trabalho apresentou resultados promissores, apresentando para um universo de 35 imagens reservadas para o teste, onde foram gerados cerca de 20.000.000 janelas que são passadas ao classificador, 118 falsos positivos e 27 falsos negativos. O algoritmo porém não é robusto em imagens de baixa resolução, nem onde o texto sofre influência de efeitos de luz e sombra (variação considerável do fundo para um mesmo texto), e tende também a classificar como texto regiões que possuam padrões repetitivos, como grades, janelas de prédios, etc.

Uma abordagem diferente é utilizada no trabalho de Zhang e Chang (2004). Ele utiliza um modelo gráfico não direcionado, chamado de Campos Aleatórios de Markov (MRF – Markov *random fields*). O estudo propõe uma busca por partes de objetos utilizando um MRF de ordem elevada. A detecção das regiões é feita utilizando a segmentação por cores através do algoritmo de agrupamento por deslocamento de médias (COMANICIU e MEER, 1997). Através do grafo de regiões adjacentes a MRF é formada adicionando a cada nó *i* uma variável de estado aleatória *X*. As características observadas incluem características de um nó, extraídas de cada nó *i*, e características de três nós, extraída de cada três regiões conectadas. Assim, a detecção de texto pode ser modelada como um problema de interferência probabilística dada todas as características observadas. A escolha da região como texto ou não texto é dada pela razão de verossimilhança das duas hipóteses opostas (x=1 sendo texto e x=0 sendo não-texto). Se esse valor for maior que um limiar, é considerado como texto.

Outro trabalho utiliza uma metodologia baseada puramente nas bordas(SAMARABANDU e LIU, 2005)(LIU e SAMARABANDU, 2006). Utiliza três parâmetros das bordas para fazer a detecção das regiões de texto: a força da borda, a densidade e a variância da orientação das linhas. Esses parâmetros são unificados em um mapa de características onde valores elevados representam regiões de caracteres. Esse mapa é então binarizado para apenas os valores representativos de texto serem considerados, e então passa por um operador morfológico para conectar regiões próximas.

Retornaz e Marcotegui (2007) aplicam uma técnica denominada "*ultimate opening*", que consiste em um operador residual que analisa a evolução de cada pixel de uma

imagem em tons de cinza sujeita a uma família de operações de abertura com tamanho crescente, que foi introduzido por Beucher (2007). A diferença entre duas aberturas consecutivas (resíduo) é considerada e o valor máximo de contraste do resíduo e o tamanho da abertura que produziu esse resíduo são armazenados. Os componentes conexos são obtidos analisando o resultado da binarização da imagem formada pelos resíduos máximos. Esse método só é capaz de detectar componentes conexos escuros, para detectar componentes claros o mesmo procedimento deve ser feito na imagem invertida. Uma eliminação de grupos pequenos que normalmente são detectadas em zonas de textura é feita realizando uma dilatação unitária e eliminando os grupos que se conectam a muitos outros. Várias características de geometria, largura de traço, regularidade de forma e de contraste são extraídas e enviadas a um classificador LDA (*Linear Discriminant Analysis*) que identifica grupos como texto ou não-texto.

Outro trabalho que utiliza a abordagem de cores utilizando o método do *split & merge* foi apresentado por Jafri, Boutin e Delp (2008). Neste trabalho, o *split & merge* é utilizado em imagens coloridas para se detectar fundos homogêneos. Os autores assumem que os textos estão sempre pelo menos parcialmente cercados por um fundo homogêneo e com isso busca por regiões que contenham buracos. Uma vez encontrada essas regiões, um teste de distância entre a cor desta região e a cor dos objetos contidos nos buracos é feita, sendo consideradas regiões válidas aquelas que possuírem uma distância acima de um limiar. O resultado é um método bastante simples para se detectar regiões com texto, mas não segmentá-las. O método, entretanto, é bastante limitado e sensível a variações na imagem. Imagens com baixo contraste não devem

apresentar bom desempenho, além de imagens com forte influência de iluminação (sombras, reflexos, etc.).

Outra aplicação utilizando operadores morfológicos foi feita utilizando *toggle mapping*, que se trata de um operador genérico que mapeia uma função em um conjunto de *n* funções (FABRIZIO, MARCOTEGUI e CORD, 2009). O trabalho apresenta a técnica denominada TMMS (*toggle mapping morphological segmentation*), que utiliza um conjunto de duas funções, erosão e dilatação da imagem em uma vizinhança *v*. O resultado é uma função s(x) que possui três valores, devido ao fato de o método gerar ruído *salt-and-pepper* para áreas homogêneas, passando então essas regiões a assumirem o valor "2" na "ternarização". As regiões de fronteira são então analisadas para detectar os caracteres. O artigo comparou o resultado do seu trabalho com outras técnicas de binarização (Niblack e Sauvola) e com o *Ultimate Opening*, apresentando resultados superiores aos três.

Para levantar o estado da arte de leitores robustos para imagens complexas, bem como fornecer uma referência para comparação de resultados e um banco de imagens público para os pesquisadores da área, foi criada em 2003 a competição de leitura robusta na International Conference on Document Analysis and Recognition- ICDAR (LUCAS, PANARETOS, *et al.*, 2003). A princípio a competição previa três categorias ligadas à leitura robusta: localização de texto, reconhecimento de caracteres e reconhecimento de palavras. Porém, apenas a localização de texto teve trabalhos inscritos. O banco de imagens da competição, disponível para *download* em http://algoval.essex.ac.uk/icdar/datasets.html, é composto por diversas imagens geradas em câmeras digitais, com resoluções variadas (desde 640x480 até 2048x1536) e

abrangendo diversas condições adversas na imagem, como por exemplo, texto sobre uma superfície de vidro que reflete um cenário, reflexo da luz saturando regiões próximas ao texto, texto em superfície não plana, entre outras. Alguns exemplos de imagens podem ser vistas na Figura 3.2.

Figura 3.2 - Exemplos de imagem da competição de leitura robusta do ICDAR2003

A competição teve quatro trabalhos inscritos e utilizou como métrica a média entre a precisão (número de regiões corretas dividido pelo número total de regiões retornado pelo programa) e o acerto (número de regiões corretas dividido pelo número de regiões de texto na imagem). As métricas da competição serão descritas com mais detalhes em outro capítulo deste trabalho. O resultado mostrou que ainda há um longo espaço para

ser percorrido no desenvolvimento de leitores capazes de identificar satisfatoriamente as regiões de texto em cenas reais. Os resultados da competição se encontram na Tabela 3.1, onde F é uma média ponderada entre precisão e acerto, e t é o tempo médio de execução do algoritmo. Mais informações a respeito dos trabalhos inscritos, dos resultados e deficiências de cada algoritmo são encontradas na publicação dos resultados da competição (LUCAS, PANARETOS, *et al.*, 2005).

Tabela 3.1 - Resultado do ICDAR2003

Sistema	Precisão	Acerto	F	t(s)
Ashida	0.55	0.46	0.50	8.7
HWDavid	0.44	0.46	0.45	0.3
Wolf	0.30	0.44	0.35	17.0
Todoran	0.19	0.18	0.18	0.3

No ano de 2005 houve uma segunda edição da competição (LUCAS, 2005) nos mesmos termos da edição anterior. A competição teve cinco competidores, e seu resultado é apresentado na Tabela 3.2. A grande contribuição da competição, além de fornecer uma visão geral da atual situação da pesquisa na área, foi de fornecer o banco de imagens. Alguns trabalhos estudados utilizam o banco, tornando possível a comparação dos resultados.

Tabela 3.2 - Resultados do ICDAR2005

Sistema	Precisão	Acerto	F	t(s)
Hinnerk Becker	0.62	0.67	0.62	14.4
Alex Chen	0.60	0.60	0.58	0.35
Qiang Zhu	0.33	0.40	0.33	1.6
Jisoo Kim	0.22	0.28	0.22	2.2
Nobuo Ezaki	0.18	0.36	0.22	2.8

4 Propostas de abordagem de Segmentação

Este capítulo apresenta o desenvolvimento deste trabalho para um sistema de localização de caracteres em imagens complexas. Primeiramente é feita uma abordagem de textos em imagens complexas, apresentando algumas das dificuldades e das considerações feitas para o desenvolvimento. Em seguida são apresentadas as diversas etapas que compõe o sistema criado.

4.1 Texto e imagem

A proposta deste trabalho é desenvolver um método de localização de texto em imagens complexas utilizando técnicas de processamento de imagens e/ou aprendizado de máquina. Na seção anterior foram vistos diversos trabalhos com objetivo semelhante, onde diversas abordagens foram apresentadas. Embora localizar caracteres não seja uma tarefa simples, existem algumas suposições que podem ser feitas de modo a simplificar o problema, são elas:

- **O caractere é monocromático**. Embora nem sempre isso seja verdade, pois podem existir efeitos artísticos no caractere de forma que ele possua mais de uma cor, porém a probabilidade de um caractere não ser monocromática é pequena. No caso de caracteres multicoloridos, imagina-se que o caractere será dividido em diversos. Uma fase posterior do algoritmo interpretaria esses segmentos e poderia agrupá-los em um único grupo. Caracteres sobre influência parcial da iluminação podem ter sua cor alterada, mas ainda assim serão considerados monocromáticos.
- **O caractere contrasta com o fundo**. Para que uma letra seja legível, deve possuir um contraste de forma que seja possível a identificação sua posição e

forma. Esse conceito não deve se deter apenas ao fundo onde o caractere está inserido, pois as bordas dos caracteres podem fornecer o contraste necessário para sua identificação, por exemplo, podemos ter um caractere na cor branca sobre um fundo branco, porém esse caractere possui um contorno escuro que contrasta com o fundo, o que o tornará legível.

- **Regiões de texto possuem características bastante específicas**. Uma alta densidade de bordas caracteriza uma região com vários objetos que contrastam entre si, que pode ser um texto ou uma região texturizada, como uma árvore ou o asfalto. Outra característica é o tamanho do traço e a distância entre bordas. Caracteres são formados por traços que normalmente possuem a mesma largura para uma mesma fonte e tamanho. O espaçamento entre os caracteres também costuma ser uniforme sobre as mesmas condições. Assume-se também que em uma palavra os caracteres se comportam de forma semelhante, seguindo a mesma orientação e possuindo as mesmas proporções (considerando as diferenças intrínsecas entre cada caractere).

Mesmo possuindo todas essas características, a tarefa de localizar todos e apenas os caracteres de uma imagem complexa não é simples. Isso se dá devido ao fato de uma imagem de cena real não ser um ambiente controlado, ou seja, além de não se saber os objetos que se esperam na imagem, existem várias influências que dificultam o trabalho. São algumas delas:

- **Iluminação**. A iluminação em excesso pode ser um problema, levando algumas regiões da imagem a saturarem tornando impossível a identificação

de formas dentro dela. Superfícies reflexivas também podem apresentar uma textura nessa situação, atrapalhando o algoritmo. Além disso, a falta de iluminação também caracteriza um problema, pois nessa situação o contraste da imagem estará baixo, dificultando também a identificação das formas. Regiões de sombra é outro fator ligado à iluminação que dificulta os trabalhos sobre a imagem, pois no limite entre a sombra e a luz há uma diferença no contraste que pode dividir os objetos, além do problema da redução do contraste que normalmente ocorre nessas regiões.

- **Textura**. Técnicas que utilizam detecção de bordas são afetadas negativamente por regiões texturizadas, como por exemplo árvores e asfalto. Nesse tipo de textura, um mesmo objeto apresenta diferenças de contraste em seu interior, gerando bordas fortes em alta densidade, que pode fazer essa região ser confundida com um texto.
- **Distorção da câmera.** A própria câmera utilizada para capturar a imagem pode gerar distorções que atrapalham o processamento da imagem. Parâmetros como tempo de exposição, flash, sensibilidade (ISO), lentes, foco, entre outros, afetam a imagem e podem gerar ruídos que prejudicam a sua análise. No caso de uma câmera de vídeo, que captura a imagem por varredura, outra distorção é produzida. Nesse tipo de imagem, pode-se observar que as linhas horizontais apresentam um efeito dégradé, resultando em uma transição mais suave nas regiões de borda e dificultando a sua detecção.
- **Perspectiva.** Em um ambiente tridimensional, como é o caso das fotos capturadas de um ambiente natural, grande parte dos objetos não se

encontram no plano focal da câmera e por isso apresentam uma distorção espacial ao serem convertidas para uma imagem em duas dimensões. Isso implica, por exemplo no caso de um texto possuindo uma orientação diferente, em o tamanho dos caracteres e seu espaçamento diminuírem conforme se afastam da câmera.

- **Superfícies não planas.** Como em imagens complexas não há restrições quanto ao tipo de objetos presentes, podemos encontrar texto em diversos tipos de superfície, apresentando a distorção da perspectiva apresentada acima. Exemplos desses objetos são as latas que apresentam superfície cilíndrica.

Figura 4.1 - Exemplos de interferência em imagens complexas. 1) Reflexo 2) Superfície cilíndrica 3) Superfície texturizada 4) Perspectiva

A Figura 4.1 apresenta exemplos das dificuldades encontradas em imagens complexas. A primeira mostra o efeito da iluminação excessiva em um objeto. A caneta apresenta um reflexo de luz sobre do texto, cortando a parte inferior dos caracteres “p”,

”r” e “o”. A segunda mostra um texto sobre uma superfície cilíndrica, pode-se notar que os caracteres de uma palavra já não possuem a mesma orientação, e ao se aproximar da lateral do objeto, as letras ficam cada vez mais próximas. A terceira imagem mostra a superfície texturizada de uma roupa, devido a essa textura a cor do objeto não é constante. A quarta figura mostra melhor a distorção da perspectiva causada em um texto que não está no mesmo plano focal da câmera.

4.2 Sistema proposto

Com base nas características de um texto em uma imagem e nas considerações apresentadas na seção anterior, a abordagem escolhida para o sistema proposto é a detecção de objetos por cor assumindo caracteres monocromáticos, e para isso optou-se por utilizar a técnica denominada *split & merge* (S&M) apresentada nos trabalhos de Lienhart (LIENHART e STUBER, 1996) e Jafri (JAFRI, BOUTIN e DELP, 2008).

O sistema consiste na divisão da imagem em grupos conexos gerados a partir das informações obtidas do espaço de cor utilizado pela imagem utilizando o S&M. A princípio o sistema foi desenvolvido para trabalhar com imagens em tons de cinza. Um refinamento dessa separação é feito na etapa seguinte de modo a agrupar caracteres multicoloridos e corrigir possíveis falhas de segmentação do S&M. Após isso são feitas sucessivas filtragens nos grupos encontrados de forma a eliminar grupos indesejados e manter apenas os caracteres na imagem. A Figura 4.2 apresenta as etapas deste trabalho na forma de um fluxograma.

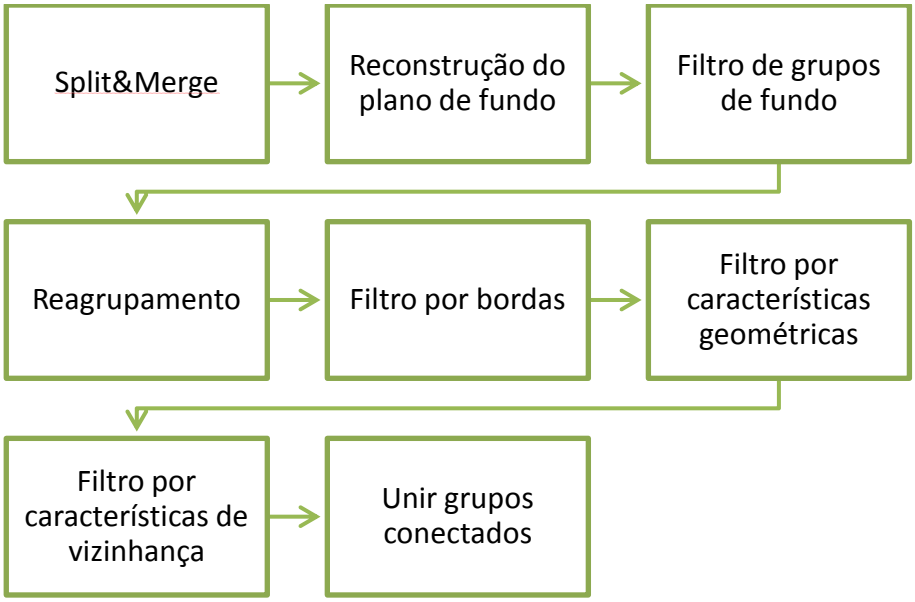

Figura 4.2 - Etapas do sistema proposto

As seções seguintes explicarão cada uma das etapas que compõem o sistema desenvolvido neste trabalho.

4.3 Split & Merge

A técnica do S&M foi estudada por Horowitz e Pavlids (1976) e consiste basicamente de duas etapas: a separação (*split*) e o agrupamento (*merge*). A primeira tem como objetivo dividir a imagem em blocos homogêneos de forma que não seja necessário trabalhar na imagem em nível de pixel, otimizando o custo computacional do algoritmo. Esta etapa é conhecida em outras aplicações como decomposição em árvore quaternária (*quadtree decomposition*)(SULLIVAN e BAKER, 1994)(WU, 1993). A segunda etapa analisa a vizinhança de cada bloco e, baseado em alguma heurística, os agrupa em componentes conexos. Cada uma dessas duas etapas será explicada em detalhes a seguir.

4.3.1 Divisão em blocos - *Split*

Dada uma imagem I, considerada como um bloco de dimensões *sx* e *sy*, tal que *sx* e *sy* sejam potências de dois, a etapa de separação consiste na divisão sucessiva da imagem em blocos de igual tamanho correspondente à quarta parte do bloco que o gerou. Para um bloco ser dividido, a aplicação de uma regra, chamada de *regra de separação*, deve falhar. Em outras palavras, enquanto um bloco não satisfizer a regra de separação, ele será dividido em quatro novos blocos, que serão testados e divididos novamente até que todos os blocos da imagem atendam à regra ou atinjam um tamanho mínimo pré-determinado (normalmente um pixel).

A Figura 4.3 mostra uma imagem binária de 8x8 pixels (1) sendo dividida pela etapa de separação do S&M. Considera-se neste exemplo a regra de separação sendo a diferença entre os pixels de maior e menor luminância do bloco a qual deve ser nula. Aplica-se essa regra primeiramente à imagem inteira, obtendo-se um resultado maior que zero e por isso, divide-se essa imagem em quatro quadrantes de dimensão 4x4 (2). Aplicando a regra a cada um desses quatro novos quadrantes, observa-se que apenas o quadrante localizado no canto inferior direito atende à regra, pois o bloco é considerado homogêneo já que a diferença entre seus pixels é igual a zero, logo esse bloco não será mais dividido. Os três quadrantes que não passaram na regra são então divididos, respectivamente, em quatro novos quadrantes, agora com dimensões 2x2 (3). Nesta nova divisão, apenas os três últimos quadrantes da segunda “coluna” falham na regra de separação, e são novamente divididos em quatro novos blocos (4), que falham à regra e também atingem o tamanho mínimo de cada bloco, finalizando então a etapa de separação da imagem.

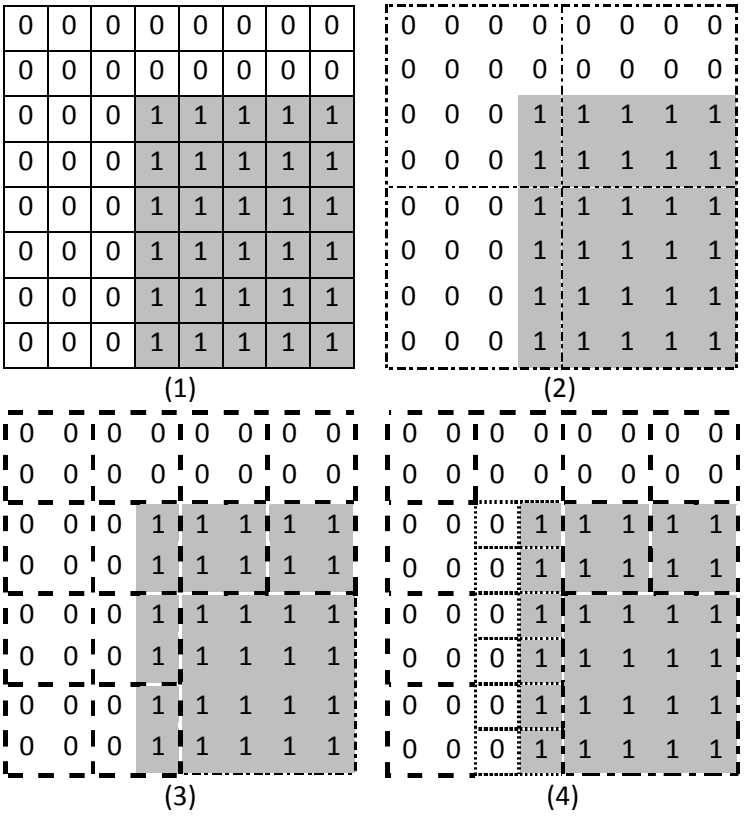

Figura 4.3 - Etapas da fase de separação- (1) imagem original (2) separação em quadrantes de 4x4 (3) 2x2 e (4) 1x1

O exemplo apresentado resultou em 22 blocos homogêneos, divididos em 1 bloco de dimensão 4x4, 9 de dimensão 2x2 e 12 de dimensão 1x1. Uma vez concluída a etapa de separação inicia-se a etapa de junção, onde blocos adjacentes e semelhantes segundo um critério devem ser unidos formando um grupo maior, gerando assim os componentes conexos da imagem agrupados pela informação do espaço de cores. Assim como na etapa de separação, na etapa de agrupamento, os blocos para serem agrupados também devem satisfazer a uma regra, a *regra de junção*, além de serem vizinhos. Conforme visto no capítulo 2, existem alguns tipos de vizinhança que podem ser considerados e essa escolha pode afetar o resultado final do algoritmo, aqui vamos considerar a vizinhança de 8 pixels. Cada bloco é testado com todos os seus vizinhos e, uma vez que atendam à regra de junção, esses blocos são agrupados em um único grupo. A etapa termina quando todos os blocos tiverem sido avaliados.

4.3.2 Agrupamento de blocos – *Merge*

A etapa de *merge* é semelhante a qualquer algoritmo de criação de componentes conexos. Cada bloco encontrado na etapa anterior tem sua vizinhança analisada e testada de acordo com uma regra de agrupamento (regra de *merge*). Caso a regra aplicada a dois grupos vizinhos seja verdadeira, então esses vizinhos pertencem a um mesmo grupo. Como não estamos trabalhando em nível de pixel, o conceito de vizinhança de 4 e de 8 serão melhor interpretados como vizinhança lateral, onde qualquer grupo que esteja conectado a um dos lados de um bloco é seu vizinho, e vizinhança diagonal, onde são considerados também os grupos nas diagonais do bloco.

Seguindo o exemplo utilizado para a etapa de divisão, o algoritmo gerou os seguintes blocos, numerados na sequência em que foram criados conforme Figura 4.4. Novamente, nesse caso binário, a regra de agrupamento será a diferença entre as luminosidades de dois grupos vizinhos deve ser nula e será considerada a vizinhança lateral e diagonal.

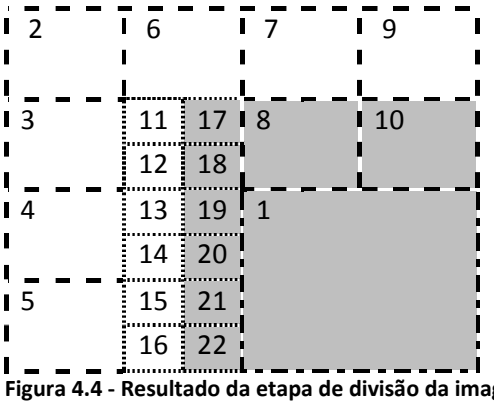

Figura 4.4 - Resultado da etapa de divisão da imagem

Inicialmente cada bloco é considerado um grupo, e eles serão analisados na ordem crescente. O bloco 1 possui como vizinho os blocos 8,10,18-22, e todos esses grupos possuem o valor 1, sendo então todos agrupados no grupo 1. Ao se analisar o bloco 2, que possui como vizinhos os grupos 3, 6 e 11, observa-se que estes também atendem à

regra de junção aplicada entre eles e o bloco 2, sendo então agrupados no grupo 2. De forma análoga, os blocos 3, 4 e 5, com seus respectivos vizinhos serão agrupados no grupo 2, gerando o agrupamento demonstrado na Figura 4.5

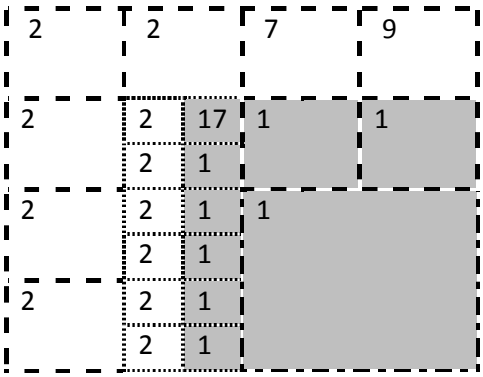

Figura 4.5 - Resultado da etapa de junção analisando até o bloco 5

Ao analisar o bloco 6, que possui como vizinhos os blocos 2, 3, 7, 8, 11 e 17, observa-se que os blocos 8 e 17 não atendem à regra de agrupamento, os blocos 2, 3 e 11 já pertencem ao mesmo grupo, sendo apenas o grupo 7 inserido ao grupo 2. Ao final da análise de todos os blocos, o resultado será composto por apenas dois grupos, correspondendo aos dois objetos da imagem (o fundo em 0 e o retângulo em 1) conforme Figura 4.6.

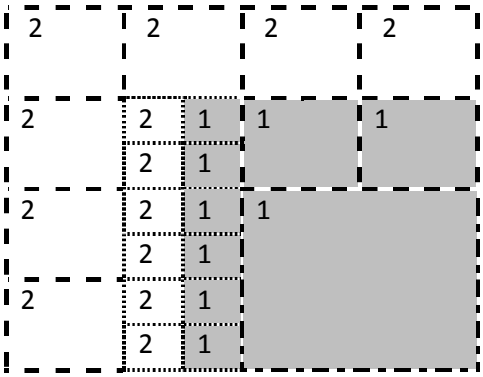

Figura 4.6 - Resultado final da etapa de junção

Quando se utiliza imagens diferentes de binárias, onde a regra de agrupamento já não é tão exata quanto ser ou não ser nulo, observa-se um problema referente à forma de se aplicar a regra. Pode-se considerar a regra aplicada aos blocos ou aos grupos nos

quais os blocos estão inseridos. Neste trabalho utilizam-se as características do grupo e não do bloco, tendo essas características atualizadas a cada inserção no grupo. Dessa forma, evita-se que objetos de cores distintas possam ser conectados em um mesmo grupo.

Para exemplificar esse problema considere a imagem da Figura 4.7-a, que possui duas barras verticais de tons de cinza distintos. A câmera utilizada para capturar essa imagem introduziu ruído, misturando as bordas de cada barra com o fundo, gerando a Figura 4.7-b (as imagens estão ampliadas para facilitar o entendimento, esse tipo de artefato ocorre mais frequentemente em objetos bem próximos). Na imagem original, as cores dos objetos eram, considerando uma linha da esquerda para a direita, contando com o fundo, 255, 66, 255, 113 e 255 (aqui são listadas luminâncias diferentes encontradas e não os pixels correspondentes a uma linha). Assumindo uma regra de agrupamento que aceite grupos com diferença de luminosidade de até 25, nenhuma dessas luminâncias seriam agrupadas. A segunda imagem, após a influência do ruído, possui na mesma linha anterior, as cores 255, 66, 80, 90, 100, 113 e 255 (respectivamente grupos 0, 1, 2, 3, 4, 5 e 0). Aqui pode ser feita a análise das duas formas de se agrupar os blocos considerando a aplicação sobre a imagem da Figura 4.7-b:

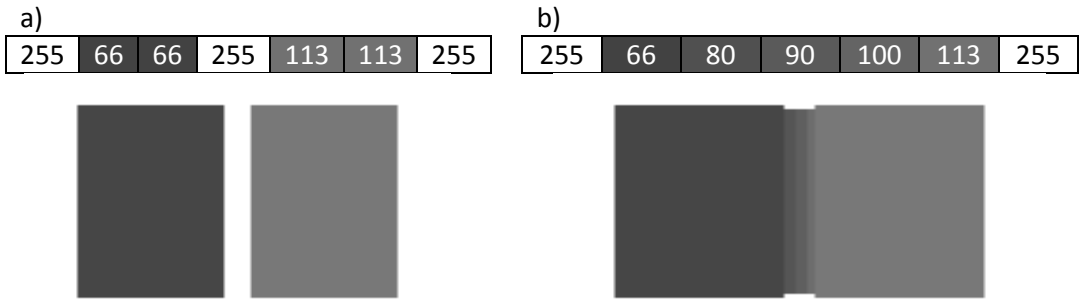

Figura 4.7 - Exemplo de interferência no agrupamento de blocos

1. Levando em consideração a cor do grupo na regra de merge:
    a. O grupo 1 possui luminosidade l=66, e é vizinho do grupo 2 com l=80, que são aceitos pela regra de merge. A nova luminosidade do grupo será a média destas cores ponderada pela área de cada grupo, e sendo o primeiro grupo maior que o segundo, será considerada a nova média do grupo 1 como 70.
    b. O grupo 2 possui vizinhança com o grupo 3 (l=90), porém a regra a ser utilizada não considera sua luminosidade de 80 e sim a do seu grupo, l=70. Ambos também são aceitos pela regra de agrupamento, e a nova luminosidade passa a ser, por exemplo, l=72.
    c. O grupo 3 é vizinho do grupo 4 com l=100, porém desta vez, ao comparar a cor do grupo (l=72), a regra de agrupamento falha pois a diferença entre as luminosidades é 28 sendo superior ao limiar definido, não inserindo o grupo 4 ao grupo 1.
    d. Seguindo a análise de forma análoga, o grupo 4 e 5 serão conectados.
2. Levando em consideração a cor do bloco:
        a. No primeiro momento, o grupo 1, 2 e 3 serão conectados de forma análoga ao item 1.
        b. Na vizinhança entre os grupos 3 e 4, temos que a diferença entre as cores dos blocos é 10, sendo aprovados pela regra e conectados ao grupo 1.
        c. De forma análoga, os grupos 4 e 5 possuem diferença de luminosidade igual a 13, também o conectando ao grupo 1.

Observa-se que, pelo método 2, os dois objetos seriam agrupados por causa do ruído criado nas bordas dos objetos, enquanto que no método 1, permaneceriam separados. A desvantagem do primeiro método é que a ordem na qual os grupos são analisados influencia no resultado.

O algoritmo retorna duas imagens, uma composta pelos grupos a qual cada pixel pertence que é o resultado de um algoritmo de componentes conexos, e uma versão da imagem original com o espaço de cores reduzido, criada substituindo-se cada grupo pela média de cores dos seus pixels. A Figura 4.8 apresenta a imagem original de um carro à esquerda, e sua versão com cores reduzidas, resultado do split & merge, à direita. Pode-se observar melhor a redução do espaço de cores da imagem ao se observar os histogramas das duas fotos, exibido na Figura 4.9.

Figura 4.8 - Imagem resultante do algoritmo S&M

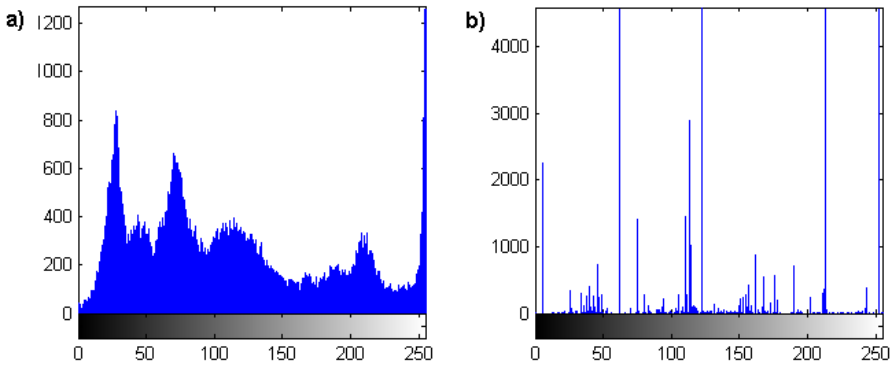

Figura 4.9 - Histograma da imagem antes (a) e após (b) a aplicação do S&M

4.3.3 Split & Merge em árvore

A etapa de separação gera um número muito elevado de grupos e analisar cada grupo com relação à sua vizinhança se torna custoso. De forma a otimizar o algoritmo, desenvolveu-se neste trabalho uma forma de representar e tratar os blocos através de uma árvore. Embora a decomposição em árvore seja difundida na comunidade científica, não se encontrou na literatura métodos descrevendo a identificação de vizinhança na representação em árvore.

A árvore é criada na etapa de divisão da imagem e pode ser representada como uma árvore quaternária, onde cada nó folha é um bloco da imagem e a altura desta folha está relacionada à dimensão deste bloco. Para facilitar o entendimento e a representação, será adotada uma árvore binária e serão consideradas apenas as alturas pares, que são as representações dos blocos. Para esse tipo de árvore, a relação entre a altura do nó e a dimensão do bloco que ele representa se dá segundo a Equação 4.1, onde *max(h)* é a altura máxima da árvore e *h(k)* é a altura do nó *k*.

$$\dim(k) = 2^{\frac{(\max(h)-h(k))}{2}} \quad \textbf{4.1}$$

4.3.3.1 Criação da árvore de blocos

A etapa de divisão da imagem em quadrantes pode ser dividida em uma divisão vertical e outra horizontal. No início, temos a imagem inteira representada pelo nó raiz da árvore. Ao aplicar a regra de separação, divide-se a imagem ao meio na vertical, gerando os nós 0 e 1 representando, respectivamente, o corte esquerdo e o corte direito. Em seguida dividem-se cada um desses blocos na horizontal, formando os quatro quadrantes denominados 0.0, 0.1, 1.0 e 1.1, onde o 0 após o ponto representa a parte superior do corte e o 1 a parte inferior. A Figura 4.10 exemplifica essa divisão

representada em uma árvore binária. Apenas as alturas pares representam quadrantes. Por ter sido adotada a representação em árvore binária no lugar de uma quaternária, a altura de um nó está relacionada às dimensões do bloco segundo a Equação 4.1. A representação em árvore do exemplo utilizado na seção anterior se encontra na Figura 4.11. Os nós folhas são representados em preto, e os números de cada nó em binário também são exibidos.

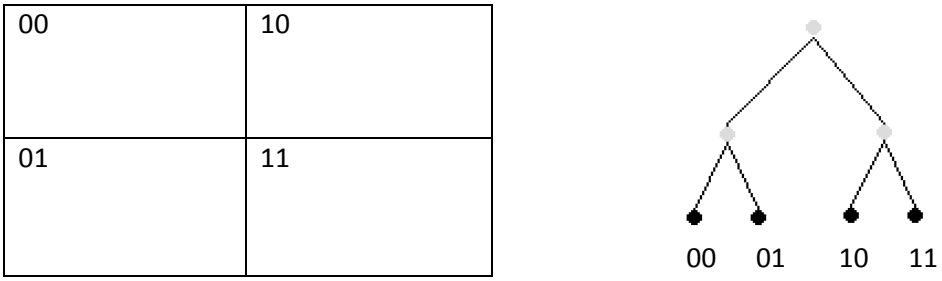

Figura 4.10 - Representação em árvore dos quadrantes

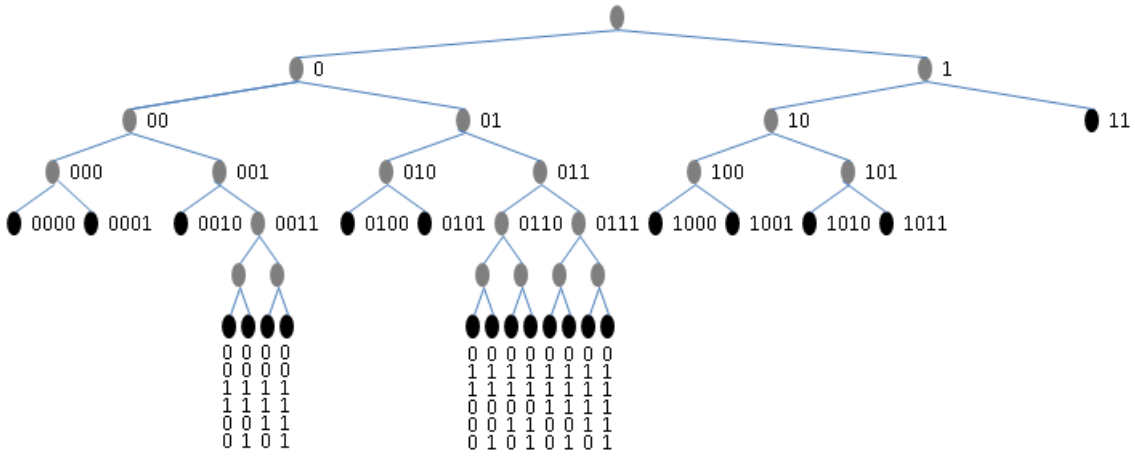

Figura 4.11 - Representação em árvore da imagem exemplo

4.3.3.2 Vizinhança espacial entre os nós da árvore

A questão agora passa a ser como identificar blocos vizinhos a partir da representação em árvore. Pode-se notar que todos os nós filhos de um mesmo pai são vizinhos, pois foram gerados a partir da mesma divisão de uma "metade" de quadrante. Tentar inferir outras regras para definir vizinhança baseada na posição do nó não é

simples e pode não haver uma solução trivial e que atenda a todos os casos. Porém, há uma solução mais simples que é utilizar a própria numeração dada a cada nó. Durante o processo de criação da árvore e a numeração dos nós, definiu-se o primeiro número como sendo a divisão da imagem na vertical, dividindo a imagem ou o quadrante em duas colunas, ou seja, duas posições distintas no eixo x. No segundo corte, é feita a divisão horizontal, separando cada nova metade em dois quadrantes da imagem, ou seja, são criadas duas novas linhas, ou posições no eixo y. Assim, podemos notar que a numeração em binário de cada nó folha é uma composição da posição x e y do quadrante. Com isso a identificação da vizinhança passa ser simplesmente a extração desta posição (x,y) do quadrante e a identificação dos seus 4 ou 8 vizinhos ((x-1,y-1), (x,y-1), (x+1,y-1), (x-1,y), (x+1,y), (x-1,y+1), (x,y+1), (x+1,y+1)). Para compor a posição do quadrante a partir do número do nó, ou compor este número a partir do quadrante, basta utilizar a relação entre os números binários expressa nas equações 4.2 e 4.3.

$$\#nó = b_{2n-1}b_{2n-2}\dots b_3b_2b_1b_0 \tag{4.2}$$

$$\#nó = x_{n-1}y_{n-1}\dots x_1y_1x_0y_0$$

$$Y_{nó} = b_{2n-2}\dots b_2b_0 \tag{4.3}$$

$$X_{nó} = b_{2n-1}\dots b_3b_1$$

Para essa abordagem funcionar deve ser feita uma restrição: um nó não pode ser vizinho de outro nó que possua uma altura maior que a sua, ou seja, um bloco não pode ser vizinho de um bloco com dimensão menor. Isso evita que um bloco possua mais que oito vizinhos, o que inviabilizaria a simplicidade da solução exposta acima. Convencionou-se que mesmo nós folha com altura inferior à altura máxima da árvore devem possuir o mesmo número de bits em sua numeração, sendo assumido como zero os bits completados. Por exemplo, o nó folha 11 do exemplo anterior, é nomeado como

110000. Por isso, considerando a dimensão do bloco analisado, deve-se ignorar os bits referentes a dimensões inferiores na análise da vizinhança.

<table>
<tr><td>000000 (0)</td><td colspan="2">001000 (8)</td><td>100000 (32)</td><td>101000 (40)</td></tr>
<tr><td rowspan="2">000100 (4)</td><td>001100 (12)</td><td>001110 (14)</td><td rowspan="2">101000 (36)</td><td rowspan="2">101100 (44)</td></tr>
<tr><td>001101 (13)</td><td>001111 (15)</td></tr>
<tr><td rowspan="2">010000 (16)</td><td>011000 (24)</td><td>011010 (26)</td><td colspan="2" rowspan="4">110000 (48)</td></tr>
<tr><td>011001 (25)</td><td>011011 (27)</td></tr>
<tr><td rowspan="2">010100 (20)</td><td>011100 (28)</td><td>011110 (30)</td></tr>
<tr><td>011101 (29)</td><td>011111 (31)</td></tr>
</table>

Figura 4.12- Figura exemplo com a numeração dos nós representando os quadrantes

A Figura 4.12 mostra a mesma figura do exemplo, porém com a numeração dos blocos de acordo com o número dos nós. Para analisar a vizinhança de um grupo, só se deve utilizar os bits correspondentes à dimensão do bloco em diante. A Tabela 4.1 mostra exemplos dos vizinhos de 4 de alguns blocos.

Analisando, por exemplo, o bloco 44 que possui dimensão igual a dois, os vizinhos são considerados com essa dimensão. Nesse caso o vizinho superior não é 101001 (41) que seria o caso da soma normal em y, e sim o 101000 (40), pois a soma só é feita a partir do bit y_1. O vizinho inferior deste bloco, seguindo análise igual a anterior, deveria ser o 111000 (56), porém, como observado na Figura 4.12, esse bloco não existe. Nesse caso, busca-se o bloco existente com dimensão superior, obtido zerando-se os bits referentes à dimensão atual (no caso, x_1 *e* y_1), obtendo assim o valor 110000 (48). Ao se avaliar o vizinho à direita do bloco 44, ocorre o estouro da variável (o resultado possui

mais bits significativos que a representação da árvore). Quando isso ocorre (tanto *overflow* quanto *underflow*) significa que o bloco vizinho não existe na imagem, pois o bloco analisado se encontra em alguma borda.

Tabela 4.1 - Vizinhança entre os nós da árvore

Dim. do nó	# nó				x	y	Vizinhos
	Dim 4	Dim 2	Dim 1	#			
1	00	11	10	**14**	011 (3)	010 (2)	001111 (**15**) *(x,y+1)* 001011 (**11**) *(x,y-1)* (***11*** *não é nó, corrigir para 01000 –* ***8*** *dimensão 2)* 001100 (**12**) *(x-1,y)* 100101 (**36**) *(x+1,y)*
1	00	10	10	**10**	011 (3)	000 (0)	001011 (**11**) *(x,y+1)* *underflow* (sem vizinho) *(x,y-1)* 001000 (**8**) *(x-1,y)* 100000 (**32**) *(x+1,y)*
2	10	11	00	**44**	110 (6)	010 (2)	111000 (**56**) *(x,y+1)* (***56*** *não é nó, corrigiria para o 110000* ***48*** *dimensão 4)* 101000 (**40**) *(x,y-1)* 100100 (**36**) *(x-1,y)* 10000100 *overflow* (sem vizinho) *(x+1,y)*

4.3.3.3 Etapa de agrupamento através da árvore

Como visto anteriormente, é possível calcular o vizinho de cada nó tendo o número do nó em questão, sua dimensão e o número máximo de blocos, utilizado para identificar vizinhos que não pertencem à imagem. Tendo essa fórmula de calcular vizinhança, vários blocos de um mesmo tamanho podem ser processados paralelamente, reduzindo o tempo gasto no algoritmo. Um cuidado deve ser tomado ao se tratar os blocos paralelamente para evitar o problema do agrupamento que considera apenas a cor do bloco, apresentada na seção 4.3.2, pois o processamento em paralelo não permitiria a atualização da média do grupo a cada inserção. Por isso,

optou-se por fazer a identificação de vizinhança de todos os blocos de uma mesma dimensão paralelamente, e agrupá-los de forma individual.

O agrupamento dos vizinhos, como explicitado anteriormente, deve ser feito começando pelos nós folha de maior altura (menores blocos) até os de menor altura (maiores blocos), pois a identificação de vizinhança ignora blocos com dimensão inferior à dimensão do bloco analisado, garantindo assim um máximo de oito vizinhos por bloco. Essa separação por dimensão dos blocos tornou possível a criação de tratamento distinto entre os diversos níveis da árvore. Por exemplo, blocos de dimensão um normalmente representam áreas ruidosas, e por isso podem ter o valor limite para a regra de agrupamento um pouco maior, de forma a facilitar que um grupo aceite esses pixels ruidosos.

De um modo geral o funcionamento da etapa de agrupamento permanece igual ao explicado anteriormente, sendo adicionado apenas a nova forma de identificação de vizinhos e a tradução dos números de cada nó para uma posição (x,y) da imagem.

4.4 Reconstrução de fundo e reagrupamento

Após a aplicação do algoritmo de S&M, é obtida uma representação da imagem em grupos conexos separados por cor. Assumiu-se que os caracteres são monocromáticos e por isso espera-se que um ou mais caracteres pertençam a um único grupo, porém, devido à complexidade da imagem e às diversas influências já apresentadas, não é incomum obter como resultado caracteres separados em mais de um grupo. Na Figura 4.13 tem-se um segmento de uma das imagens utilizada para testes do algoritmo à esquerda, bem como os grupos gerados pelo S&M à direita. Nessa imagem a regra de agrupamento utilizada foi que o módulo da diferença entre a média de dois grupos

vizinhos deve ser inferior a 30 para blocos de dimensão unitária, e inferior a 25 para os demais blocos. Com isso, estima-se que um grupo possa ter uma variação de luminosidade de até 50 (+25 e -25 acima da média do grupo), porém, como mostra a figura, uma mesma letra possui níveis de cinza variando desde 0 até 91, e suas extremidades estão mais escuras que o meio do caractere, resultando na divisão em grupos distintos conforme Figura 4.13.

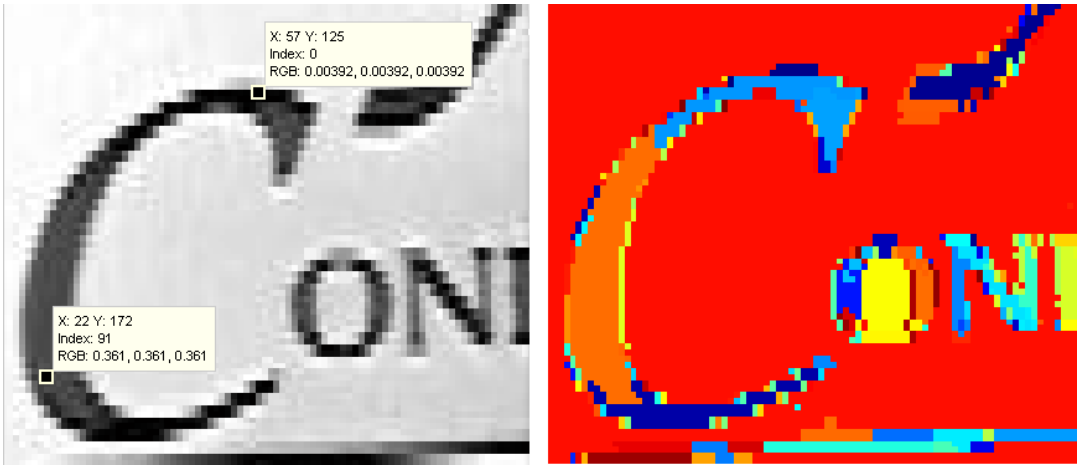

Figura 4.13 - Separação de um mesmo caractere em diversos grupos

Seja por erro na escolha dos parâmetros das regras de separação e agrupamento ou por característica própria da imagem, essa divisão do caractere compromete o desempenho da etapa de seleção, visto que grupos correspondentes a pedaços de caracteres dificilmente podem ser diferenciados de grupos que devem ser eliminados por não pertencerem a nenhum caractere, como linhas e bordas, ou outros grupos quaisquer. Na maioria das vezes, deixar a regra de agrupamento menos restritiva não é uma solução, pois afeta o resultado de outros grupos na imagem, normalmente trazendo mais problemas do que benefícios, conforme observado na Figura 4.14, onde a figura da esquerda foi obtida aplicando-se o S&M com a regra de junção citada anteriormente, e a da direita, onde a regra utilizou os limites 40 para blocos de

dimensão unitária e 35 para blocos de dimensão maior. Observa-se que neste último caso, a placa do veículo ficou deteriorada, perdendo alguns caracteres.

Figura 4.14 - Defeitos resultantes de uma regra de junção menos rígida

Diante deste problema, foi desenvolvida uma nova etapa de agrupamento que utiliza os resultados do S&M e informações sobre o fundo no qual cada grupo está inserido para avaliar se dois ou mais grupos conectados devem ser reagrupados em um único grupo. Para isso, primeiro é feita uma reconstrução do plano de fundo da imagem para que possa ser identificado em qual fundo cada grupo está inserido. Após isso é feita a análise dos grupos e seu eventual re-agrupamento. Essas duas etapas são descritas a seguir.

4.4.1 Reconstrução do plano de fundo

A reconstrução do plano de fundo busca remover objetos menores da imagem, como caracteres e outras formas menores, de modo a recriar a imagem contendo apenas grupos considerados como fundo. É aplicada sobre a imagem com o espaço de cores reduzido resultante do S&M, facilitando a composição da imagem reconstruída. Objetos correspondentes ao plano de fundo costumam ser mais homogêneos do que regiões

com caracteres, e por isso se concentram em blocos maiores na etapa de separação e consequentemente esses blocos são agrupados em grandes grupos na etapa de agrupamento do S&M. Como essas regiões de fundo normalmente englobam os caracteres e objetos menores, tendem a serem objetos com uma largura considerável. Assim, os grupos correspondentes ao plano de fundo possuem algumas características próprias que podem ser utilizados para identificá-los.

A primeira característica considerada é a área total do grupo. Este mesmo parâmetro é utilizado posteriormente para remover grupos grandes nas etapas de seleção das regiões candidatas. Assume-se uma quantidade máxima de pixels que um grupo candidato a caractere pode possuir, já levando em consideração os tamanhos máximos de caracteres esperados bem como palavras conectadas. Grupos que possuam uma quantidade de pixels (área) superior a esse parâmetro passam a ser considerados como pertencentes ao plano de fundo.

Nem todos os objetos do plano de fundo são selecionados pelo critério de área total. Objetos que devem ser considerados como fundo, mas são menores e contêm textos menores, como placas de automóvel, precisam de outro critério de aceitação. Como um objeto de plano de fundo costuma envolver diversos objetos menores, como uma sequência de caracteres, ele tende a ser mais largo que objetos comuns, então a relação entre a largura e a altura do objeto pode ser utilizada. Porém, só ela não é suficiente, pois linhas e bordas possuem relação de aspecto elevada, mas não correspondem a objetos de fundo. Assim, definiu-se o critério considerando, além da relação de aspecto, uma largura e altura mínima, considerando as dimensões da imagem. Com isso evita-se

que grupos pequenos e largos, como linhas e grupos de caracteres conectados, sejam considerados na reconstrução.

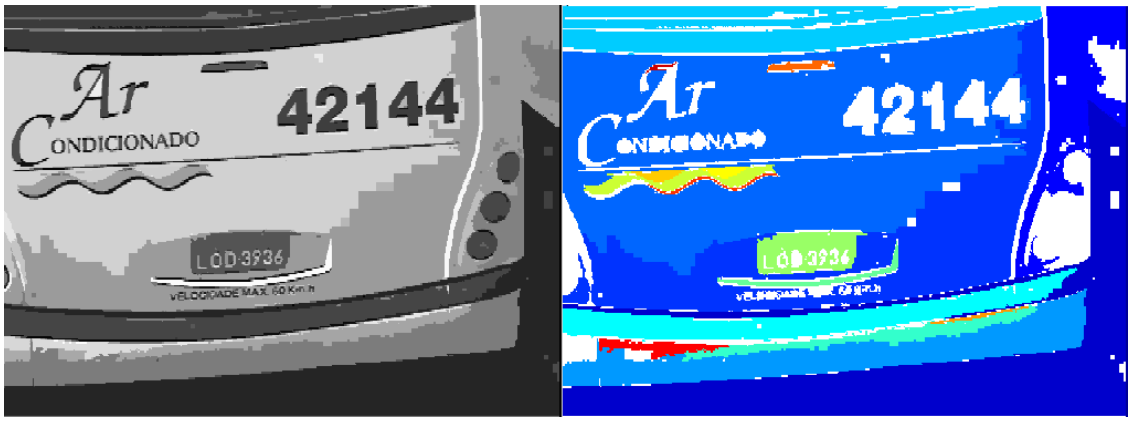

Figura 4.15 - Seleção dos grupos pertencentes ao plano de fundo. Os pixels em branco não pertencem a um grupo de fundo

Na Figura 4.15 tem-se o resultado da seleção dos grupos de fundo, tendo selecionado ao todo 19 dos 1528 grupos existentes retornados pelo S&M. Após a seleção, é iniciada a etapa de reconstrução do fundo. Essa etapa poderia ser feita analisando cada um dos grupos não selecionados, mas isso teria um alto custo computacional, além do inconveniente de haver grupos que não estão conectados a nenhum grupo de fundo, e por isso optou-se por outra abordagem.

Primeiramente é aplicado a cada grupo de fundo um operador de dilatação e identificam-se os grupos que foram conectados a esse fundo expandido. Esses grupos são então analisados e inseridos ao fundo que mais se aproxima de sua luminância. Essa primeira etapa busca fazer uma análise individual dos grupos próximos ao fundo, diferente da etapa seguinte onde os grupos não pertencentes ao fundo são agrupados, podendo trazer problemas quanto a grande variação da luminância dentro de um grupo.

A partir da imagem de rótulos dos grupos de fundo, obtém-se a imagem invertida que gera a imagem de grupos não selecionados mostrada na Figura 4.16, reduzindo dessa forma o número de grupos a serem analisados. Os testes feitos mostraram que não houve prejuízo observável adotando essa estratégia no lugar da análise individual dos grupos do S&M.

Figura 4.16 - Imagem dos rótulos dos grupos não selecionados como fundo.

Verifica-se então, para cada grupo não pertencente ao fundo desta nova imagem, quais grupos de fundo estão conectados a ele. Para isso, realiza-se uma dilatação do grupo analisado, e essa imagem dilatada é utilizada como uma máscara para se obter os grupos ao redor do grupo analisado. Como os componentes conexos foram obtidos a partir do inverso da imagem de fundo, é garantido que todo grupo esteja conectado a pelo menos um grupo de fundo. Para selecionar a qual fundo o grupo pertence, conta-se quantos pixels estão na fronteira entre cada grupo de fundo e o grupo analisado, e se

esse valor for inferior a 20% do total de pixels do contorno do grupo em questão, este grupo de fundo não é considerado (se não houver grupos com mais de 20% de interseção, esta regra é desconsiderada). Caso haja mais de um grupo pré-selecionado, é escolhido aquele cuja média de luminância esteja mais próxima do grupo em questão. Dessa forma, fragmentos de grupos de fundo que por algum motivo tenham sido segmentados do grupo maior passam a ser reagrupados ao fundo.

O grupo analisado recebe o mesmo rótulo do grupo de fundo selecionado, preenchendo então seu espaço correspondente na imagem de fundo. Uma imagem pode ser formada ao final do processamento substituindo o rótulo pela cor do fundo correspondente. Um exemplo mais didático do processo de reconstrução de fundo é apresentado a seguir.

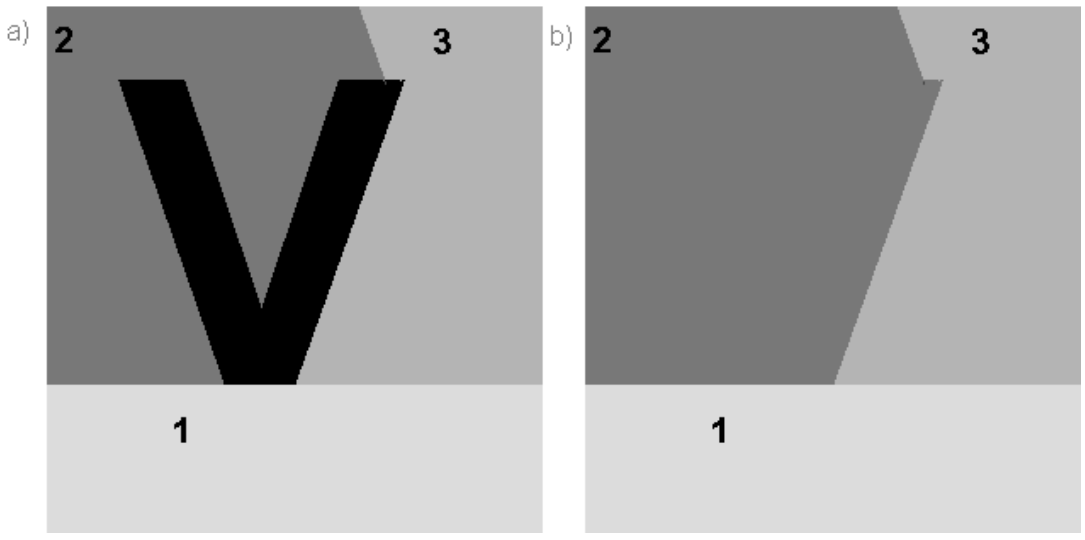

Figura 4.17 - Exemplo de decisão de seleção de grupo de fundo

Considere a imagem da Figura 4.17-a, onde a letra “V” está inserida em três fundos distintos. Ao analisar as fronteiras do grupo “V” com os grupos de fundo, o grupo 1 será eliminado pois a quantidade de pixels em sua região de fronteira com V esta abaixo de 20% do contorno do grupo “V”. Apenas os grupos 2 e 3 serão analisados e o grupo 2

será escolhido por apresentar uma cor mais próxima à cor do grupo resultando na imagem da Figura 4.17-b. A Figura 4.18 mostra o resultado final da reconstrução do plano de fundo para a imagem apresentada na Figura 4.15.

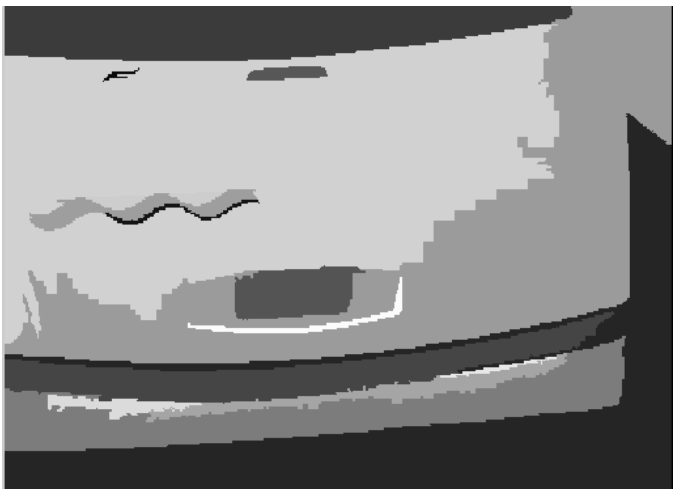

Figura 4.18 - Imagem do plano de fundo reconstruído

4.4.2 Reagrupamento

Conforme explicitado anteriormente, diversos fatores afetam a imagem e o resultado do S&M, causando a divisão de um mesmo objeto em dois ou mais grupos e isso impacta negativamente na etapa de seleção de grupos, sendo muito difícil a diferenciação entre partes de caracteres e objetos indesejados. O algoritmo de reagrupamento desenvolvido neste trabalho busca unir esses segmentos de caractere em um único grupo.

A idéia surgiu observando os resultados da técnica de *threshold* NiBlack(QI, XU, *et al.*, 2005) que utiliza características estatísticas calculadas ao redor de cada ponto para definir se o pixel está acima, abaixo ou em um nível próximo ao plano de fundo. Essa técnica tem como desvantagem a necessidade de uma escolha adequada para os parâmetros dos filtros de média e desvio padrão, além de não funcionar de forma

satisfatória em imagens pequenas ou com fundos muito variados, que atrapalham o cálculo da média e comprometem a decisão do grupo do pixel.

A técnica de NiBlack precisa utilizar o filtro de média para ter uma estimativa da cor do plano de fundo no qual o pixel está inserido. Utilizando a técnica de reconstrução do plano de fundo podemos obter o plano de fundo existente na imagem inteira e de forma mais confiável que o filtro de média, pois sua cor não é influenciada pelos objetos inseridos nele.

Assim, o primeiro passo é obter a cor de cada grupo de fundo a partir da imagem do fundo reconstruído obtida anteriormente, e a partir da imagem original, o valor do desvio padrão de cada plano de fundo. Em seguida é feita uma varredura em todos os grupos obtidos pelo S&M. Para cada grupo, identifica-se qual ou quais planos de fundo estão sobre ele, e através do contorno do grupo, identificam-se quais grupos de fundo estão conectados ao grupo analisado. Para cada grupo de fundo *n* identificado é analisada a conectividade dos grupos. De forma semelhante à técnica NiBlack, atribui-se valores para cada grupo baseado na média e desvio padrão do fundo conforme as regras apresentadas na Tabela 4.2. Como aqui são utilizados os pixels realmente pertencentes ao fundo, não é utilizada a constante que pondera o desvio padrão do fundo, apresentada na técnica NiBlack. A atribuição do valor para um grupo *k* segue as condições apresentadas na Tabela 4.2.

Tabela 4.2 - Regras de junção de grupos

Condição	Valor
$MediaGrupo(k) > \ MediaFundo(n) + DesvFundo(n)$	Grupo(k) = 1
$MediaGrupo(k) < \ MediaFundo(n) - DesvFundo(n)$	Grupo(k) = -1
$\lvert MediaGrupo(k) - MediaFundo(n)\rvert \leq \ DesvFundo(n)$	Grupo(k) = 0

Para dois grupos conectados serem unidos em um único grupo, eles devem receber o mesmo valor de acordo com as regras acima. Porém, o tratamento é um pouco mais complexo no caso do grupo analisado estar inserido em mais de um fundo. Nesse caso, dois grupos só podem ser unidos se ambos receberam valores iguais em todas as análises dos *n* fundos diferentes. Isso foi feito para garantir uma coerência maior na união dos grupos.

Ao término da varredura uma nova imagem de grupos é formada associando os novos rótulos aos grupos correspondentes. A Figura 4.19 exibe o resultado do algoritmo para a imagem utilizada como exemplo nesta seção. À esquerda tem-se o resultado do algoritmo S&M que gerou 1528 grupos e à direita a imagem de grupos após o reagrupamento, que reduziu o número de grupos para 475 e não apresentou perda de informação em nenhum caractere. A Figura 4.20 exibe o mesmo trecho da palavra "Condicionado" apresentado anteriormente (Figura 4.13), agora após o reagrupamento. Pode-se notar que não existe mais as subdivisões de cada caractere nessa imagem.

Figura 4.19 - Resultado do algoritmo de reagrupamento

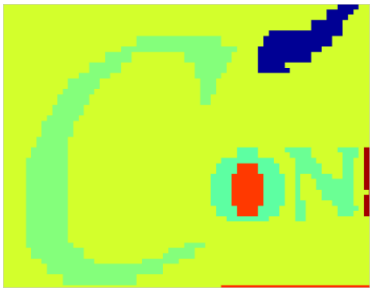

Figura 4.20 - Detalhe da imagem reagrupada

4.5 Seleção de grupos

Uma vez tendo dividido a imagem em objetos baseados na cor através do S&M e refinado esse agrupamento através do algoritmo de reagrupamento, deve-se agora iniciar o processo de filtragem dos grupos. Esse processo visa eliminar os grupos que não são de interesse, deixando na imagem apenas os caracteres.

O processo de filtragem consiste de diversas etapas executadas de forma seqüencial, de modo que o resultado de uma é utilizado como entrada para a etapa seguinte. Isso é feito para reduzir o custo computacional, visto que dentro da grande variedade de grupos detectados há aqueles mais simples de serem eliminados, e aqueles que exigem um processamento mais complexo. Se esse mesmo tratamento fosse dado a todos os

grupos impactaria no tempo de execução do algoritmo, além de não ser garantido que o processamento mais sofisticado atenda a todos os grupos.

São utilizados cinco etapas ao todo, representadas no fluxograma da Figura 4.21.

Figura 4.21 - Etapas de seleção de grupos

4.5.1 Eliminação de grupos grandes

A primeira e mais simples etapa de seleção de grupos serve para eliminar grupos excessivamente grandes. De forma a melhorar o desempenho do sistema, define-se um tamanho máximo esperado de um caractere e grupos com um tamanho superior a essa estimativa são desconsiderados.

Como parâmetro para se definir o tamanho de um grupo optou-se pela área ocupada por esses grupos, em outras palavras, o número de pixels que pertencem a esse grupo. De forma a tornar o parâmetro mais genérico, independente da resolução da imagem, utiliza-se como parâmetro de seleção o percentual da área total da imagem, ao invés de um valor absoluto. Esse parâmetro deve ser definido levando-se em consideração tanto caracteres grandes presente nas imagens, quanto grupos de caracteres que tenham sido agrupados, bem como os diferentes tipos de imagens a que o sistema se propõe a trabalhar. Tendo em vista as considerações expostas, e com base nos grupos obtidos do conjunto imagens de desenvolvimento, definiu-se o valor máximo de área para um grupo sendo 1% da área total da imagem.

Vale lembrar que esse mesmo parâmetro foi utilizado anteriormente na identificação de grupos grandes para a reconstrução do plano de fundo. Embora não seja responsável pela eliminação de muitos grupos, a contribuição desta etapa é importante pois os grupos do plano de fundo não têm uma forma bem definida, podendo não ser filtrados nas etapas seguintes, além de algumas análises em grupos grandes, como a vizinhança do grupo, serem muito custosas.

Durante os testes, verificou-se que a eficiência global do sistema era maior se essa etapa fosse aplicada entre as etapas de reconstrução do plano de fundo e a de reagrupamento. Isso ocorre pois, caso não sejam eliminados antes, alguns grupos de caracteres acabam sendo unidos aos grupos de fundo. Na Figura 4.22 tem-se um exemplo do resultado desta primeira etapa de filtragem.

Figura 4.22 - Grupos após filtragem de grupos grandes

4.5.2 Eliminação por interseção de bordas

Essa etapa retoma o conceito da abordagem por contraste: um caractere para ser legível deve contrastar com o fundo, e por isso deve ser possível detectar uma borda

entre ele e o fundo. Assim, grupos que não tenham interseção com nenhuma borda da imagem não podem pertencer a um caractere. Essa etapa elimina principalmente grupos gerados pelo S&M devido ao gradiente de cores de um objeto, por exemplo, um reflexo de luz que tornou uma área mais iluminada dividindo esse objeto em mais de um grupo.

Para a detecção de bordas utiliza-se a técnica de Canny (CANNY, 1986), forçando seu valor de corte superior em 40 (encontrado experimentalmente) e 16 para o limite inferior (calculado automaticamente com base no limite superior). Em seguida é feita a dilatação desta imagem de borda utilizando uma máscara retangular de 3x3 pixels, de forma que os pontos ao redor das bordas também sejam considerados. Uma vez obtida essa imagem binária onde as bordas são representadas pelo valor 1, aplica-se o operador lógico E entre esta imagem e a imagem de grupos. Com isso, todos os pontos de grupos que ocupam o mesmo espaço de uma borda serão obtidos e, após listar todos os valores diferentes de zero e eliminar valores repetidos, obtém-se a lista dos grupos que interceptam alguma borda. Apenas esses grupos detectados são inseridos na nova imagem de grupos, que é o resultado obtido nessa etapa. Aplicando essa etapa na imagem do exemplo anterior obtém-se a imagem de grupos da Figura 4.23, nela foram eliminados alguns grupos que não caracterizavam um objeto e sim algum erro de agrupamento do S&M. Ao todo esta etapa manteve 579 dos 637 grupos existentes na imagem.

Figura 4.23 - Resultado após a etapa de eliminação por interseção de bordas

4.5.3 Eliminação por características geométricas

Mesmo após as duas primeiras etapas, ainda é grande o número de grupos indesejados na imagem. Através da observação das imagens do conjunto de desenvolvimento, constatou-se que uma quantidade considerável de grupos é formada por linhas, correspondente principalmente às bordas de objetos maiores, como pode ser observado na Figura 4.23.

Para eliminar esses e outros grupos que não são de interesse, foram analisadas uma série de características geométricas e estatísticas dos grupos. Alguns valores limites de parâmetros foram levantados analisando os dados de grupos representando caracteres e grupos que não são caracteres, outros valores utilizaram uma linha de raciocínio lógica que será explicada posteriormente.

A primeira característica utilizada é a área do grupo. Essa característica já foi explicada anteriormente, porém agora é utilizada para eliminar grupos muito pequenos. Uma vez que já foi feito o reagrupamento espera-se não encontrar segmentos de

caracteres, e com isso, pequenos grupos podem ser eliminados sem risco para o resultado. O valor de corte desta característica deve ser escolhido com cuidado, pois existem caracteres que se aproximam muito de uma reta vertical, e por isso possuem área mínima, como é o caso das letras *i*, *j* e *l*. Assim como para a área máxima, esse parâmetro foi definido como um percentual da área total da imagem, assumindo que em imagens maiores mesmo os caracteres pequenos possuirão uma área considerável. Isso pode não ser verdade em todos os casos e alguns caracteres em imagens grandes serem legíveis apesar de ter uma área pequena, porém, esse erro foi considerado tolerável diante da redução de grupos indesejáveis que esse parâmetro proporciona.

Embora a área mínima possa eliminar grupos pequenos, para definir mais critérios de eliminação estipulou-se o tamanho mínimo que um caractere deve ter para que tenha suas características preservadas e seja legível. Esse tamanho foi idealizado considerando-se o caractere *E*, que precisa de cinco linhas para manter-se íntegro. Isso porque esta letra necessita de três linhas pretas, correspondente aos seus traços horizontais e duas linhas brancas, correspondendo ao espaço entre dois traços (Figura 4.24). Uma altura menor de caractere significaria o desaparecimento de uma das "pernas" da letra *E*, dificultando o discernimento da mesma, embora outros caracteres possam permanecer legíveis.

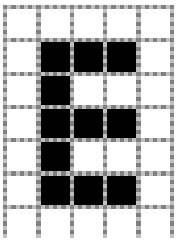

Figura 4.24 - Tamanho mínimo possível para um caractere legível foi definido como 5 pixels

Outra característica utilizada para a eliminação é a altura do grupo. Grupos que não possuam uma altura mínima para ser legível devem ser eliminados. A altura do grupo pode ser obtida através do tamanho do retângulo que envolve o grupo (*bounding Box)*. Porém, esta altura é um parâmetro pouco confiável, pois grupos inclinados apresentam uma altura diferente da altura real do grupo. Por isso, a projeção horizontal do grupo é utilizada para estimar sua altura. Define-se como projeção horizontal a contagem do número de pixel existente em cada coluna da imagem do grupo. Como se assumiu que caracteres não podem ter uma altura inferior a 5 pixels (considerando um erro de 1 pixel, assume-se altura mínima de 4 pixels), são eliminados quaisquer grupos que tenham o valor máximo da projeção horizontal igual ou inferior a 3 pixels.

A projeção horizontal é muito útil para identificar retângulos e linhas retas, que tendem a ter altura constante por todo o seu comprimento. Com isso, pode-se utilizar a variação da projeção horizontal como um parâmetro para eliminar esses grupos. Porém, existem letras que se aproximam de linhas e não devem ser eliminadas, e por isso é levado em consideração, junto com a variação da projeção, a relação de aspecto do grupo, evitando assim eliminar pequenos grupos constantes, como as letras i e l. Assim, foi escolhido como limiar de corte um desvio padrão da projeção horizontal inferior a 1.25, o que pode ser interpretado como um grupo de altura contínua ao longo de seu comprimento, com uma variação tolerável em torno de 1 pixel, característica típica das linhas e retângulos.

Outros dois parâmetros são utilizados para identificar linhas, são eles extensão e solidez. Por extensão, define-se como a razão entre a área do grupo e a área do retângulo que envolve o grupo. A área do grupo é sempre menor que a área do seu

retângulo, sendo assim essa razão sempre varia entre 0 e 1. Uma extensão próxima de 1 significa um grupo que preenche quase na totalidade o retângulo, ou seja, o próprio grupo é um retângulo (considerando que uma linha horizontal pode ser considerada uma degeneração de um retângulo). Novamente, caracteres como o i e l (e algumas vezes o j) apresentam valores próximos a 1 e por isso optou-se por não utilizar o valor máximo como parâmetro de seleção. Por outro lado, valores baixos de extensão simbolizam um grupo longo, fino e inclinado, que não está bem distribuído ao longo do seu retângulo. Esse caso caracteriza muitas das linhas encontradas como resíduos das etapas anteriores de seleção, e por isso foi utilizado para a filtragem.

De forma semelhante, a solidez é definida como a razão entre a área do grupo e a área do menor polígono convexo que envolve o grupo. O polígono convexo costuma se aproximar melhor do formato do grupo e consequentemente possui uma área menor que o retângulo, dando mais precisão ao parâmetro e sendo menos susceptível a inclinações do grupo.

Todos os grupos que estão fora das regras citadas acima são eliminados e uma nova imagem de rótulos de grupos é formada apenas com os grupos restantes. A Figura 4.25 exibe o resultado obtido nesta etapa para a mesma imagem de exemplo utilizada nas outras etapas. Neste exemplo restaram 60 grupos. Observa-se nessa imagem a grande quantidade de grupos eliminados em relação à Figura 4.23.

Figura 4.25 - Resultado da etapa de eliminação por características geométricas

4.5.4 Eliminação por características de vizinhança

Tendo eliminado uma grande parte dos grupos indesejados nas últimas etapas, restam agora poucos grupos que não são caracteres. Para eliminar esses grupos já se faz necessárias análises mais detalhadas e trabalhosas. Nessa etapa faz-se uma consideração que afeta os tipos de texto que se espera encontrar com esse trabalho: foi decidido que não será considerado um caractere isolado, ou seja, para ele ser aceito deve estar próximo de outros caracteres (grupos) formando palavras. Caso esse comportamento não seja desejado, esta etapa pode ser eliminada, afetando entretanto todo o desempenho do sistema deste ponto em diante.

Para fazer essa avaliação, analisam-se cada um dos grupos restantes da etapa anterior e aplica-se um operador morfológico de dilatação. Como elemento estruturante foi escolhido um retângulo com 3 pixels de altura e com o dobro da largura do grupo analisado. Dessa forma grupos que estejam na mesma direção horizontal e a uma largura de distância do grupo serão identificados. Caso nenhum grupo vizinho

tenha sido identificado, o grupo analisado é eliminado do processo. Além disso, para ser considerado, o grupo vizinho deve possuir uma altura compatível com a do grupo analisado, definido como entre 0.5 e 1.5 vezes sua altura. Esses valores levam em consideração as diferenças entre caracteres maiúsculos e minúsculos, e ajuda a eliminar grupos que não pertençam a palavras.

Uma vez que é identificado um grupo como vizinho, este grupo é marcado como válido e removido da análise. Dessa forma, evita-se o caso em que um grupo **A** é vizinho de um grupo **B**, porém B não é vizinho de **A**, o que ocorre quando se aplica o operador de dilatação e a largura do grupo **A** não é suficiente para tocar o grupo **B**, porém quando se analisa **B**, que possui uma largura maior, identifica-se uma vizinhança com o grupo **A**.

Ainda nesta análise de vizinhança, buscam-se também resquícios de grupos de fundo que não foram eliminados anteriormente. A identificação destes grupos é feita através da interseção dos *Bounding Box* de todos os grupos. Considerando os *n* grupos restantes, a interseção entre os grupos retorna uma matriz $\boldsymbol{M}$ de *n*x*n* onde cada elemento $\boldsymbol{M_{ij}}$ contém a área (número de pixels) em comum entre os grupos *i* e *j*. A soma de cada coluna dessa matriz representa a área total de interseção do grupo representado pela coluna *i* com todos os outros grupos. Desta soma, desconta-se o valor da própria área de *i* que foi considerada na soma, e divide o resultado pela área do grupo *i*. Com isso, obtém-se o percentual do *Bounding Box* que intercepta outros grupos. Foi definido que se esse valor for superior a 40% significa que intercepta uma quantidade considerável de grupos e é caracterizado como um fundo englobando caracteres que será eliminado. Essa eliminação só é feita para grupos que apresentem

relação de aspecto superior a 2, preservando assim caracteres fechados que geram pequenos grupos em seu interior, como as letras B, O e Q.

O resultado desta etapa, aplicado na imagem de grupos após as etapas anteriores pode ser visto na Figura 4.26. A imagem contém apenas 43 grupos.

Figura 4.26 - Resultado da etapa de eliminação por vizinhança

4.5.5 Unir grupos conectados

A próxima etapa do processo de seleção reduz o número total de grupos existentes agrupando todos os grupos que possuem contatos entre suas regiões de fronteira, ou seja, aqueles que estão diretamente conectados. Ao passar pelas etapas de eliminação anteriores, assume-se que nesta fase do algoritmo os caracteres já se encontram isolados do fundo ao qual estavam inseridos e de outros grupos indesejáveis, e com isso é possível realizar esta etapa sem que os grupos sejam conectados erroneamente, prejudicando o resultado da segmentação.

A etapa agrupa caracteres fechados, como as letras A, B, D, O e outras, com os grupos situados em seu interior, que em sua grande maioria não são eliminados nas

etapas anteriores devido à suas características serem semelhante às de um caractere. Além disso, é muito útil também na união dos grupos indesejados, que comumente se encontram conectados. A Figura 4.27 mostra a imagem de grupos resultantes da aplicação desta etapa na imagem de grupos da etapa anterior, contendo apenas 29 grupos

Figura 4.27 - Grupos resultantes da etapa de união de grupos conectados

4.6 Considerações finais

Este capítulo apresentou todo o processo de desenvolvimento do trabalho, mostrando como é feita a transformação da imagem em grupos conexos através da técnica de *split & merge*, a melhoria dessa divisão em grupos a partir do reagrupamento e reconstrução do plano de fundo, bem como a eliminação dos grupos que não são caracteres através das sucessivas etapas de filtragem. Ao longo deste capítulo viu-se a evolução das etapas do algoritmo aplicadas a uma imagem de exemplo retirada do conjunto de desenvolvimento utilizado neste trabalho. No capítulo seguinte será feita uma análise mais aprofundada do resultado deste trabalho em todas as imagens do conjunto de desenvolvimento, bem como a aplicação deste trabalho ao conjunto de

imagens da competição ICDAR, que será utilizado como base de comparação do desempenho do sistema desenvolvido aqui com outros trabalhos do meio científico. Os parâmetros utilizados nas etapas do algoritmo são descritos na Tabela 4.3

Tabela 4.3 - Parâmetros utilizados no algoritmo

Descrição do parâmetro	**valor**
Limiar da regra de separação	10
Limiar da regra de junção	30 – blocos com dim=1 25 – blocos com dim>1
Área máxima de um grupo	1% da área total
Área mínima de um grupo	0,0067% da área total
Limite superior do Canny	40
Solidez mínima do grupo	0,25
Extensão mínima do grupo	0,15
Altura mínima da projeção horizontal	4
Razão de altura de grupos vizinhos	Entre 0.5 e 1.5

5 Resultados da Segmentação Proposta

Neste capítulo serão apresentados os resultados obtidos para o sistema proposto, bem como uma análise do desempenho de cada etapa descrita na seção anterior. Além disso, será apresentado o resultado do sistema para as imagens do banco de dados da competição ICDAR para ser utilizado como base de comparação entre outros trabalhos. Para tanto, precisa-se definir quais são as métricas utilizadas para aferir o desempenho do sistema, que serão descritas na seção seguinte.

5.1 Métricas

Para ter-se uma boa medida da qualidade do resultado do sistema, o ICDAR propõe duas métricas que serão utilizadas neste capítulo: a precisão e o acerto. Assumindo que o sistema retorna retângulos que marcam as regiões de caracteres, o acerto é definido como a razão entre o número de retângulos encontrados pelo sistema e que contêm caracteres e o número total de caracteres na imagem (Equação 5.1). É uma métrica que mede o percentual de caracteres de uma imagem que o sistema consegue detectar.

Por outro lado, a precisão é definida como a razão entre o número de retângulos encontrados pelo sistema e que contêm caracteres e o número total de retângulos encontrados pelo sistema (Equação 5.2). Esta métrica representa o percentual de grupos detectados pelo sistema que são caracteres.

$$acerto = \frac{\#grupos\_de\_caracteres\_detectados}{\#total\_de\_caracteres\_na\_imagem} \quad \textbf{5.1}$$

$$precisão = \frac{\#grupos\_de\_caracteres\_detectados}{\#total\_de\_grupos\_detectados} \quad \textbf{5.2}$$

Para contabilizar o número de grupos de caracteres detectados, utiliza-se o percentual de interseção entre o grupo encontrado e o *target* segundo a Equação 5.3. Dessa forma, se o grupo encontrado for muito maior ou muito menor que o target, não será contabilizado como um acerto.

$$\%Interseção = \frac{area(grupo\_detectado \cap target)}{area(grupo\_detectado \cup target)} \quad \textbf{5.3}$$

Mesmo que um grupo contenha mais de um caractere, ele é avaliado e caso a interseção entre os retângulos dos caracteres contidos neste grupo e o retângulo do grupo corresponda a mais de 50% da área do retângulo do grupo analisado, ele é considerado como um acerto. Esse valor de corte considera a área ocupada pelos espaços entre caracteres e os espaços acima e abaixo dos caracteres no caso de palavras inclinadas. Por outro lado, esse valor de corte deve ser considerado para que grupos muito grandes e que eventualmente englobem caracteres não contem como acerto.

A Figura 5.1 mostra o exemplo de uma palavra inclinada segmentada em um único grupo (retângulo tracejado) e os *targets* marcados manualmente utilizados para a verificação do acerto (retângulos pontilhados). Nota-se que uma parte considerável do retângulo do grupo possui interseção com os *targets*. Neste caso, 81,5% da área está ocupada pelos retângulos dos caracteres.

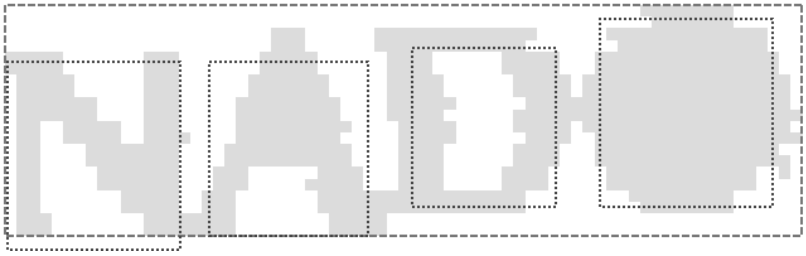

Figura 5.1 - Comparação dos retângulos de um grupo e dos targets. O retângulo tracejado corresponde ao grupo e o pontilhado aos caracteres marcados

Mesmo que um grupo contenha mais de um caractere, para o cálculo do acerto, é considerado o número de caracteres corretos e não o número de grupos, dando a noção do número de caracteres segmentados corretamente. Já para o cálculo da precisão é considerado o número de grupos, dando assim a noção da quantidade de grupos que são segmentados corretamente.

5.2 Resultados obtidos

Durante o desenvolvimento deste trabalho, foram utilizadas 14 imagens de veículos em tons de cinza, sendo 8 imagens em 640x480, 5 em 320x240 1 em 800x600 pixels. Ao término de cada etapa se calcula o acerto e a precisão, tendo assim a noção da evolução do algoritmo ao longo de suas diversas etapas. Na Figura 5.2 e Figura 5.3 são exibidas as imagens deste conjunto bem como o resultado final para cada imagem.

A aplicação nesse conjunto de imagens de desenvolvimento gerou os resultados apresentados na Tabela 5.1. São exibidos os valores médios de precisão e acerto para cada uma das seguintes etapas:

1. Eliminação de grupos grandes + reagrupamento;
2. Eliminação por interseção de bordas;
3. Características geométricas
4. Características de vizinhança;
5. União de grupos conectados

Tabela 5.1 - Resultado médio para as etapas do sistema proposto.

Etapa	**1**	**2**	**3**	**4**	**5**
Acerto	82,54%	82,54%	81,68%	79,60%	79,70%
Precisão	3,26%	3,87%	28,02%	44,65%	55,34%

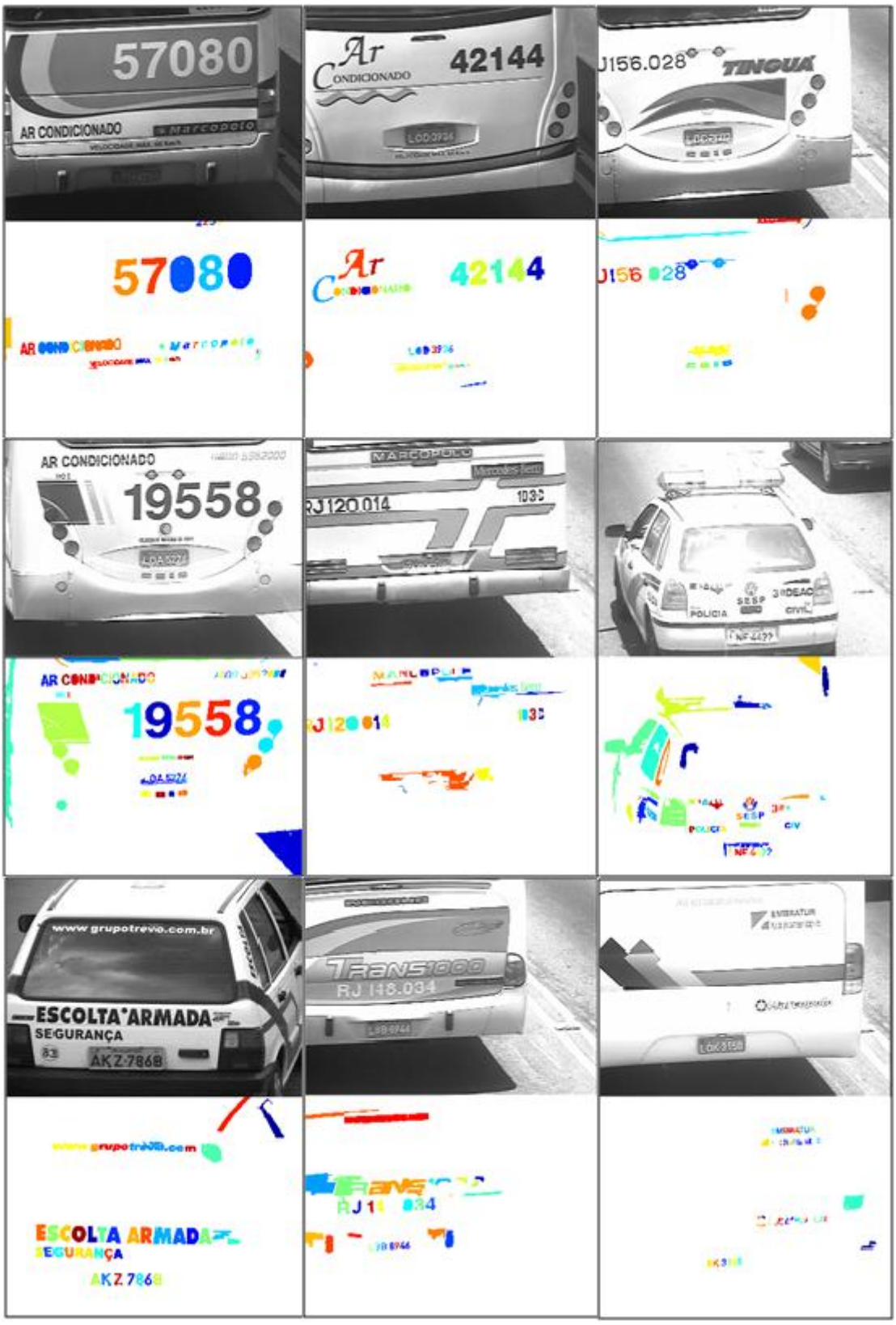

Figura 5.2- Imagens do conjunto de desenvolvimento e seus resultados

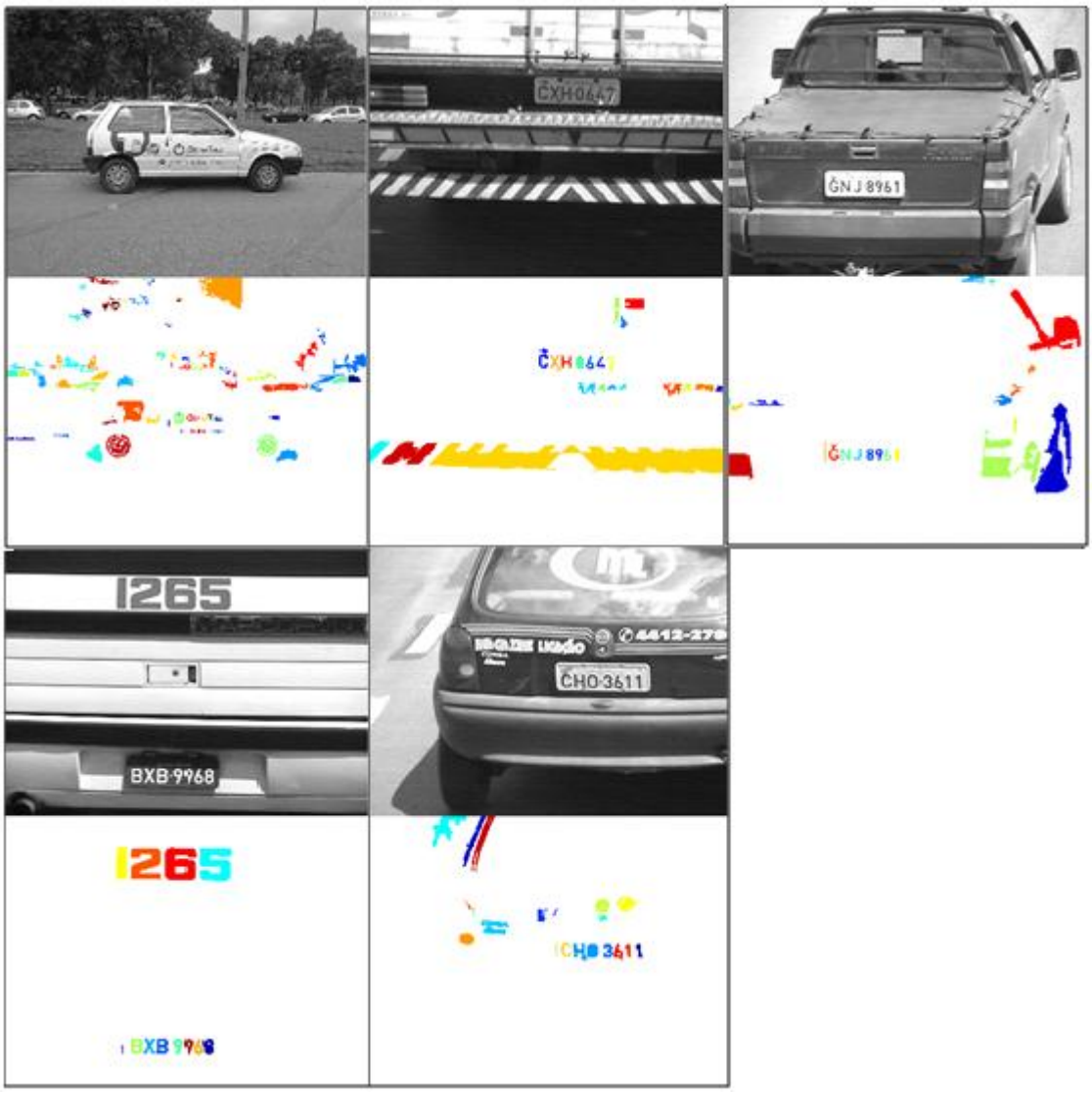

Figura 5.3- Imagens do conjunto de desenvolvimento e seus resultados

Nota-se que as duas primeiras etapas não geram impacto na segmentação dos caracteres, mas apresentam pouco impacto na precisão do sistema. Isso ocorre porque grupos grandes e aqueles que não interceptam bordas são minoria na imagem. Já a terceira etapa, que avalia as características geométricas e leva em consideração parâmetros calculados para os caracteres, apresenta uma melhora significativa na precisão, eliminando vários grupos indesejáveis, tendo um pequeno impacto de 1 ponto percentual no resultado do acerto. A quarta etapa, que só é possível devido à restrição imposta ao sistema de aceitar apenas caracteres pertencentes a palavras, apresenta

mais uma queda de 2 pontos percentuais no acerto, porém a precisão do sistema é melhorada para 44,65%. Comparado o ganho na precisão com a redução do acerto, é possível concluir que o impacto dessa restrição ao sistema é mínimo devido à baixa ocorrência de caracteres isolados, e traz de fato mais vantagens que desvantagens. Por fim, a quinta etapa, que não elimina grupos e sim reduz a quantidade total agrupando todos os grupos que estão diretamente conectados, não afeta o acerto, provando que a suposição de que nesta fase todos os caracteres já se encontram isolados do fundo e de outros grupos é verdadeira.

Conforme dito no capítulo anterior, a etapa de eliminação de grupos grandes é aplicada entre as etapas de reconstrução de fundo e a de reagrupamento, visando um melhor desempenho e evitando que os grupos de fundo sejam conectados a algum grupo de forma a atrapalhar o resultado. Ao utilizar a etapa de eliminação de grupos grandes somente após a etapa de reagrupamento, nota-se uma queda no desempenho em relação ao resultado apresentado na Tabela 5.1. A Tabela 5.2 apresenta os resultados obtidos no sistema com essa inversão da ordem das etapas conforme descrito.

Tabela 5.2 - Resultados obtidos com a eliminação de grupos grandes após a etapa de reagrupamento

Etapa	**1**	**2**	**3**	**4**
Acerto	74,02%	74,02%	69,93%	71,98%
Precisão	4,19%	4,84%	47,78%	57,75%

A redução da taxa de acerto do sistema ocorre principalmente pelo fato de a etapa de reagrupamento unir vários grupos de caracteres, de forma que em alguns casos esses grupos assumem características de grupos de fundo, que são eliminados na etapa de

eliminação de grupos grandes, conforme visto na Figura 5.4 que mostra o resultado de uma imagem quando a eliminação de grupos grandes é feita antes do reagrupamento (a), e depois do reagrupamento (b). Nota-se que os conjuntos de caracteres “COND” e “NADO” foram eliminados neste último caso, pois uma vez agrupados, esses caracteres adquirem a relação de aspecto de um grupo considerado de fundo, sendo assim removidos dos grupos desejados.

a)

b)

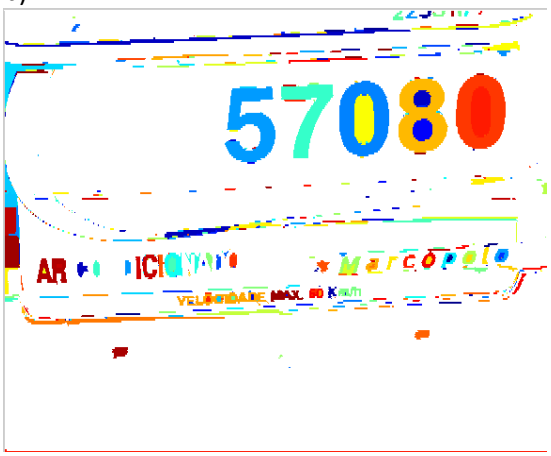

Figura 5.4 - Diferenças entre a utilização da etapa de etapa de eliminação de grupos grandas: a) antes do reagrupamento, b) após o reagrupamento.

A seguir foi feita uma avaliação da eficiência das etapas de eliminação de grupos. Executou-se o algoritmo retirando uma das etapas e se comparou os resultados obtidos com o resultado global da Tabela 5.1. Dessa forma pode-se estudar o impacto de cada uma das etapas no resultado final do sistema. As seis linhas da Tabela 5.3 apresentam o resultado desses testes e cada coluna apresenta o resultado obtido em cada uma das etapas existentes, de forma semelhante à das tabelas anteriores. Os testes são:

1. Resultado global – Teste com todas as etapas de eliminação de grupos apresentado na Tabela 5.1

2. Retirando a eliminação de grupos grandes – A etapa de eliminação de grupos, localizada entre a reconstrução de fundo e o reagrupamento é removida neste teste. Nesse caso o resultado da coluna 1 representa puramente o resultado da etapa de reagrupamento.
3. Retirando a eliminação por bordas – Neste teste, o resultado da coluna 2 não apresenta resultados pois a etapa foi removida.
4. Retirando a eliminação por características geométricas - Neste teste, o resultado da coluna 3 não apresenta resultados pois a etapa foi removida.
5. Retirando a eliminação por características de vizinhança - Neste teste, o resultado da coluna 4 não apresenta resultados pois a etapa foi removida.
6. Retirando a etapa de unir grupos conectados – Nesse caso o resultado é puramente o global apresentado na linha 1, retirando-se a coluna 5, pois a remoção dessa etapa, por ser a última, não afeta outras etapas.

Tabela 5.3 – Testes da importância de cada etapa de eliminação de grupos

	Etapa	1	2	3	4	5
1	**Acerto**	82,54%	82,54%	81,68%	79,60%	79,70%
	Precisão	3,26%	3,87%	28,02%	44,65%	55,34%
2	**Acerto**	87,11%	87,11%	84,46%	82,15%	81,97%
	Precisão	4,23%	4,84%	29,94%	45,90%	55,31%
3	**Acerto**	82,54%		81,86%	79,60%	79,70%
	Precisão	3,26%		25,59%	44,31%	54,78%
4	**Acerto**	82,54%	82,54%		79,25%	78,25%
	Precisão	3,26%	3,87%		12,24%	19,99%
5	**Acerto**	82,54%	82,54%	81,68%		80,88%
	Precisão	3,26%	3,87%	28,02%		35,77%
6	**Acerto**	82,54%	82,54%	81,68%	79,60%	
	Precisão	3,26%	3,87%	28,02%	44,65%	

Os testes realizados levaram a uma conclusão inesperada: a etapa de eliminação de grupos grandes, independente de estar antes ou após o reagrupamento, tem um impacto negativo significativo no desempenho do sistema. Ao se remover esta etapa, obteve-se no final um aumento no desempenho médio de mais de dois pontos percentuais, sendo que a precisão manteve-se praticamente inalterada.

Porém, investigando-se mais a fundo esse ganho, foi descoberto que grande parte deste impacto no resultado se deve à última imagem, apresentada na Figura 5.3, onde as palavras “Magazine Ligação” e o número de telefone têm os caracteres próximos, formando grupos grandes que seriam eliminados por essa etapa. Na versão original do sistema (Figura 5.5-b) essa imagem apresenta um acerto de 31,03% enquanto que na versão do teste 2 (Figura 5.5-c) apresenta 86,21%.

Figura 5.5 - Diferenças no resultado do sistema. a) imagem original, b) resultado do sistema com eliminação de grupos grandes, c) resultado sem eliminação de grupos grandes.

Outra imagem que apresentou ganho pode ser vista na Figura 5.6, porém neste caso verifica-se visualmente que não há diferença entre os grupos segmentados corretamente. Aqui a diferença ocorreu por uma pequena diferença no grupo da palavra “velocidade” que na Figura 5.6-b não foi aceita pela verificação de acerto por estar um pouco maior que a palavra, mas foi aceito como acerto no teste 2, apresentado na

Figura 5.6-c. As demais imagens do conjunto de desenvolvimento apresentaram perdas nos resultados quando se retira a etapa de eliminação de grupos grandes.

Figura 5.6 – Um falso exemplo de diferença de resultados

Os outros testes realizados mostram que as etapas 2 a 5 são necessárias, cada uma sendo responsável por uma parcela do resultado final. A etapa mais importante, ou seja, a que mais impacta no resultado final do sistema, é a eliminação de grupos baseado em suas características geométricas, seguida pela eliminação por características de vizinhança. A etapa de eliminação por bordas é a menos significativa, mas por não impactar no acerto do sistema, justifica sua utilização.

5.3 Tempo de execução

Utilizando a ferramenta *profile* do MATLAB, levantou-se o tempo de execução do algoritmo em cada uma das etapas, apresentado na Tabela 5.4 - Tempo de execução de cada função do sistema onde tempo total representa o tempo que a função demorou incluindo todas as funções existentes dentro da função analisada, e tempo próprio representa apenas o tempo do próprio algoritmo, excluindo as chamadas de função. Os resultados apresentados são os obtidos em todas as 14 imagens, por isso foi calculado também a média de tempo por imagem. A etapa final de união de grupos conectados não foi contabilizada por levar um tempo total inferior a 0,1 segundos.

Tabela 5.4 - Tempo de execução de cada função do sistema

Função	Tempo total (s)	Tempo próprio (s)	Tempo total médio (s)	Tempo próprio médio (s)
Split&Merge por árvore	293,16	241,92	20,94	17,28
Reconstrução do plano de fundo	15,95	3,44	1,14	0,25
Elimina grupos de fundo	5,06	0,11	0,36	0,01
Junta grupos	29,89	6,94	2,14	0,50
Elimina por bordas	3,78	0,22	0,27	0,02
Elimina por características geométricas	41,40	5,10	2,96	0,36
Elimina por vizinhança	9,68	2,98	0,69	0,21

Vale ressaltar que, por ser uma linguagem interpretada, o MATLAB costuma ser mais lento do que sistemas desenvolvidos em linguagens compiladas. Dessa forma, os tempos obtidos neste trabalho não devem ser utilizados como base de comparação com outros sistemas desenvolvidos em outros ambientes.

5.4 Conjunto da competição ICDAR

Para validar o desempenho do sistema com um grupo de imagens complexas que não foram utilizadas em seu desenvolvimento optou-se por utilizar o conjunto de exemplo da competição ICDAR(LUCAS, PANARETOS, *et al.*, 2003). É um conjunto de 20 imagens obtidas através de câmeras fotográficas e com resoluções variando de 640x480 a 1600x1200. As imagens são bastante desafiadoras, apresentando superfícies não planas, letras sobre vidro com reflexo, imagens com regiões saturadas, texturizadas, entre outros.

As marcações utilizadas na competição (*targets*) também são bastante exigentes, utilizando todos os caracteres existentes na imagem, legíveis ou não, o que leva a certa discordância sobre o que deveria ser utilizado como *target* para fins de teste. A Figura 5.7 apresenta algumas das imagens do conjunto. Nota-se que na Figura 5.7-a, há alguns números brancos sobre o papel branco na parede, o que é difícil até mesmo para um avaliador humano identificar quais são os números, embora consiga inferir a presença de caracteres naquela região.

Figura 5.7 - Algumas imagens do conjunto de exemplo do ICDAR 2003

Conforme dito no início deste capítulo, há diferenças na filosofia dos sistemas inscritos no ICDAR e o desenvolvido neste trabalho. Basicamente um trabalha com palavras inteiras, enquanto o outro não se importa em encontrar caracteres simples ou conjuntos deles. Dada essa diferença, torna-se difícil utilizar a marcação do conjunto-

verdade criada para a competição. Devido a essa dificuldade, não será feita aqui a comparação com os outros dois conjuntos disponibilizados pela competição (conjunto de treino e conjunto de testes, cada qual com cerca de 250 imagens). Assim, optou-se por fazer a inspeção visual da imagem final obtida pelo sistema, realizando uma estimativa de agrupamento para considerar as palavras, mas que também servirá para grupos indesejáveis, e com isso calculou-se os valores de precisão e acerto para cada imagem considerando-se as palavras. Levantou-se também o acerto dessas imagens considerando a métrica deste trabalho, que considera os caracteres individualmente. Os resultados obtidos para cada uma das 20 imagens podem ser observados na Tabela 5.5 e o resultado global (total de acertos) e médio se encontra na Tabela 5.6.

Tabela 5.5 – Métricas medidas nas imagens do conjunto de exemplo do ICDAR

	Caracteres		**Palavras**	
	Acerto	**Precisão**	**Acerto**	**Precisão**
1	83,33%	88,89%	50,00%	50,00%
2	100,00%	27,66%	100,00%	15,38%
3	59,09%	86,67%	50,00%	66,67%
4	80,00%	7,14%	0,00%	0,00%
5	47,62%	45,45%	44,44%	9,09%
6	89,23%	76,67%	78,05%	72,73%
7	96,88%	59,62%	85,71%	46,15%
8	100,00%	53,19%	100,00%	33,33%
9	17,65%	34,21%	21,62%	38,10%
10	0,00%	0,00%	0,00%	0,00%
11	53,57%	47,62%	40,00%	20,00%
12	66,67%	32,50%	33,33%	3,57%
13	0,00%	0,00%	0,00%	0,00%
14	20,00%	33,33%	0,00%	0,00%
15	100,00%	46,88%	100,00%	20,00%
16	69,57%	59,26%	33,33%	8,33%
17	100,00%	93,94%	100,00%	85,71%
18	68,97%	62,50%	66,67%	30,77%
19	75,00%	16,05%	66,67%	8,00%
20	63,64%	80,00%	33,33%	25,00%

As imagens do conjunto de exemplo e o resultado final do sistema para essas imagens podem ser vistos no Anexo I. Desconsiderando as imagens que resultaram em 0% de acerto das palavras, o sistema apresenta um desempenho conforme a Tabela 5.7.

Tabela 5.6 – Resultado geral para o conjunto do ICDAR

	Caracteres		Palavras	
	Acerto	**Precisão**	**Acerto**	**Precisão**
Resultado médio	**61,35%**	**50,56%**	**52,05%**	**23,36%**
Resultado global	**64,56%**	**47,58%**	**50,16%**	**26,64%**

Tabela 5.7 – Resultado para o conjunto do ICDAR sem as imagens com 0% de acerto

	Caracteres		Palavras	
	Acerto	**Precisão**	**Acerto**	**Precisão**
Resultado médio	**66,58%**	**54,29%**	**58,55%**	**27,99%**
Resultado global	**74,45%**	**56,94%**	**62,70%**	**33,30%**

Considerando as métricas para os caracteres individuais, o resultado obtido não está muito distante do observado no conjunto de desenvolvimento. Algumas imagens apresentaram resultados muito baixos, e serão avaliadas na seção seguinte, onde serão apresentadas algumas sugestões de melhorias não implementadas que poderiam melhorar o desempenho para essas imagens.

5.5 Avaliação das imagens e sugestões

Nos resultados obtidos para as imagens individuais do conjunto do ICDAR, algumas imagens apresentaram uma taxa de acerto de 0%. Isso não necessariamente representa que o método proposto é incapaz de identificar caracteres nessas imagens. Por vezes, a causa dessa baixa eficiência é devido aos parâmetros utilizados no sistema que não estão de acordo com as características da imagem, por ser muito difícil definir um

parâmetro que atenda em todas as condições. Em outros casos existem melhorias que podem ser implementadas de forma a aumentar o desempenho, especialmente nas imagens apresentadas. Poucas serão as vezes em que o baixo desempenho se dá puramente pela incapacidade das técnicas utilizadas.

Serão analisadas primeiramente as imagens que obtiveram 0% de acerto. Depois algumas imagens que também apresentaram um baixo índice de desempenho.

Figura 5.8 - Imagem 4 do conjunto ICDAR

A imagem da Figura 5.8 apresenta a quarta imagem do conjunto, que contém apenas uma palavra, “PEPSI”. Pelo acerto elevado obtido considerando os caracteres pode-se dizer que o sistema perdeu algum caractere da palavra, não conseguindo segmentar a palavra como um todo. De fato, a letra “S” foi perdida (Figura 5.9-a), devido ao parâmetro de área máxima subdimensionado para essa imagem. Ao se alterar esse parâmetro para 4% da área total consegue-se segmentar a palavra completa, passando a acertar 100% dos targets da imagem (Figura 5.9-b).

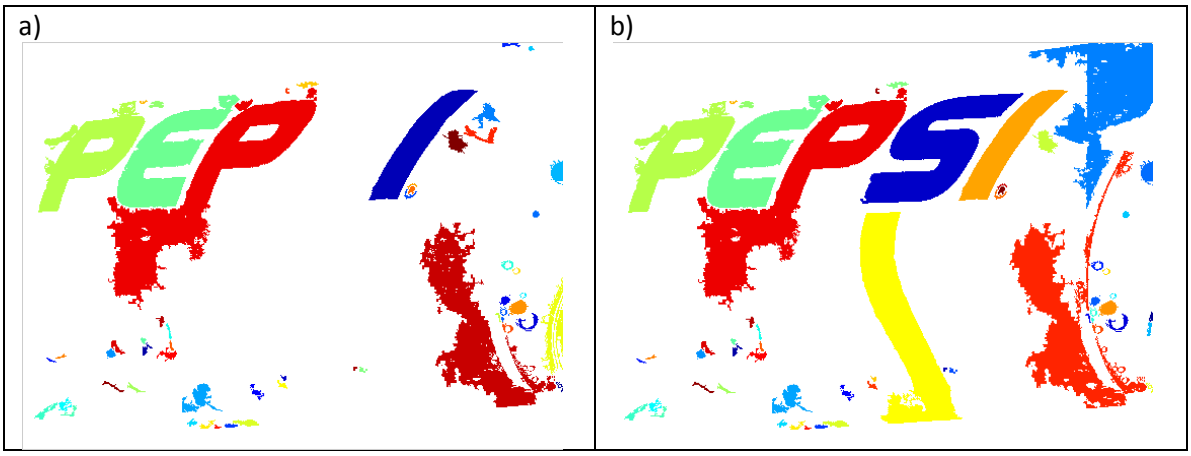

Figura 5.9 - Resultado para a imagem 4. a) com parâmetro antigo, b) com parâmetro de área máxima ajustado

A segunda imagem que ficou com acerto nulo é a imagem de número 10. Nenhum caractere foi segmentado e isso pode ser explicado observando a imagem em cores e sua conversão para tons de cinza na Figura 5.10. Ao se converter para tons de cinza, a imagem perde o contraste dos caracteres com o fundo, ambos assumindo uma cor cinza muito próxima, que não consegue ser diferenciada no S&M. Uma sugestão que poderia corrigir esse problema seria adaptar o S&M para trabalhar com imagens em cores. Dessa forma, as letras em vermelho que estão bem destacadas na imagem colorida seriam facilmente identificadas pelo algoritmo.

Figura 5.10 - Imagem de número 10 do conjunto ICDAR em cores e em tons de cinza

A terceira imagem, de número 13 exibida na Figura 5.11, é a única imagem deste conjunto que dificilmente poderia ser segmentada pelo sistema proposto. Isso ocorre pois as letras nesta imagem são formadas por retas horizontais paralelas e espaçadas. Apenas uma visão mais global da imagem, que entende o contexto dos objetos (como no caso do ser humano), seria capaz de agrupar este conjunto de retas como letras. Das abordagens apresentadas neste trabalho, é possível que a única capaz de segmentar essa imagem seria a abordagem por textura, por buscar as informações ao redor de cada pixel e classificá-lo como pertencente à região de imagem ou não.

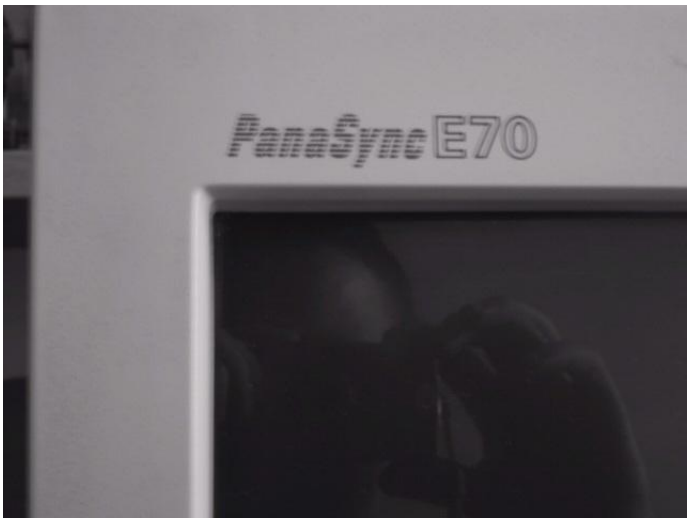

Figura 5.11 – Imagem não segmentada pelo sistema.

A quarta imagem com acerto de palavras em 0%, representada na Figura 5.12, é mais um exemplo de imagem contendo apenas uma palavra e que talvez obtivesse um melhor desempenho utilizando imagens em cores ao invés de tons de cinza. Essa imagem conta com um forte foco de luz (reflexo do *flash*), saturando uma parte da imagem acima dos caracteres. Na imagem em tons de cinza essa saturação se aproxima da cor dos caracteres, unindo-os em um único grupo. Se fosse aplicada a imagem em cores, a cor vermelha da letra poderia servir para diferenciá-la da região branca saturada pelo flash, possibilitando a sua segmentação. A Figura 5.12 mostra essa imagem em cores sem nenhum tratamento e a Figura 5.13, a imagem em tons de cinza seguida pela imagem retornada pelo S&M com seu espaço de cores reduzido. Note que o S&M agrupou as letras “CONDIT” e a zona saturada em um único grupo.

Figura 5.12 - Imagem 14 do conjunto do ICDAR

Figura 5.13 - Imagem 14 em tons de cinza (esq.) e com redução do espaço de cores pelo S&M (dir.)

As sugestões apresentadas aqui também trariam ganhos nas outras imagens do conjunto do ICDAR, melhorando de forma global o desempenho do sistema.

6 Conclusões e Trabalhos Futuros

Neste trabalho, foi desenvolvido um sistema de segmentação de caracteres baseado na homogeneidade dos objetos. Primeiro, assumiu-se que os caracteres são monocromáticos e com isso aplicou-se a técnica de *split&merge* para segmentar a imagem em objetos de tonalidades semelhantes. Ao se constatar que essa pressuposição era insuficiente, pois por diversos motivos caracteres apresentavam variações, algumas vezes bruscas, em sua tonalidade, foi desenvolvido um método de análise e reagrupamento utilizando a informação do plano de fundo no qual o caractere está inserido, melhorando dessa forma a segmentação dos caracteres. Por último foram desenvolvidas cinco etapas para a remoção dos grupos que não são caracteres.

Os resultados obtidos foram satisfatórios, sendo comparáveis aos resultados obtidos na competição ICDAR 2003. Se o resultado obtido no conjunto de exemplo se mantivesse no conjunto de avaliação, o sistema proposto ocuparia a terceira posição da competição.

As etapas de seleção de grupos ainda merecem mais atenção pois a precisão do sistema está abaixo do desejado, reduzindo o desempenho global deste trabalho.

De um modo geral, este trabalho apresentou toda a dificuldade envolvida no desenvolvimento de um leitor robusto e de se trabalhar com imagens complexas.

6.1 Resultados obtidos

Os resultados obtidos pelo sistema, principalmente o referente ao acerto do sistema, foram satisfatórios, apesar de não reconhecer todos os caracteres da imagem. Quanto à precisão, houve uma dificuldade em definir regras capazes de diferenciar

completamente caracteres de outros grupos. Em alguns casos, os grupos indesejáveis se assemelham a caracteres tornando muito difícil sua eliminação. Alguns grupos longos também restaram na imagem final, e embora tenha se cogitado a idéia de eliminar grupos com altura ou largura maior que um percentual das dimensões da imagem, essa idéia foi descartada pois poderia eliminar linhas de caracteres que ocupassem grande parte da largura da imagem, reduzindo a robustez do sistema.

6.2 Limitações

Algumas limitações encontradas no sistema estão ligadas diretamente às influências da imagem. Como visto nos testes das imagens do ICDAR, regiões com focos de luz que saturam uma parte da imagem, dificilmente são reconhecidas pelo sistema, embora visualmente ainda seja possível ler os caracteres presentes nessa região.

Regiões com baixo contraste, como na Figura 5.10 ou na região da placa do ônibus da primeira imagem da Figura 5.2, não são captadas pelo sistema. Essa limitação se deve aos parâmetros utilizados para as regras de junção e separação, porém utilizar valores menores impactaria negativamente no resto da imagem.

6.3 Trabalhos futuros

Durante o desenvolvimento deste trabalho, foram identificados alguns pontos de melhoria que ficam como sugestão para trabalhos futuros. As melhorias almejam principalmente a melhoria no acerto e na precisão do sistema. Porém, como não foi o foco deste trabalho desenvolver o sistema de forma a ser executado no menor tempo possível, é sugerido para o futuro uma análise e otimização dos algoritmos gerados, para diminuir o tempo de execução do sistema e de aperfeiçoar o uso de memória.

A primeira sugestão é a implementação da idéia apresentada na seção Avaliação das imagens e sugestões de adaptar o algoritmo de *split&merge* para trabalhar com imagens coloridas. A idéia é utilizar as informações de cores (no espaço de cor HSV) nas regras de separação e junção, fazendo com que essa informação extra ajude a definir melhor os grupos segmentados.

Outra idéia de melhoria no algoritmo de S&M que surgiu durante o trabalho e que não pode ser implementada é a de utilizar valores variáveis nas regras de separação e junção, deixando esses limiares se adaptarem às condições da região da imagem analisada. Essa idéia é aplicada em diversos algoritmos de binarização adaptativa e poderia trazer ganhos na segmentação de regiões muito escuras ou muito claras, como regiões com fortes sombras ou com reflexo e saturação.

Há bastante espaço para melhorias no sistema desenvolvido neste trabalho em relação às etapas de seleção de grupos. Muitos grupos indesejáveis ainda se mantêm na imagem final e o estudo de novas regras e técnicas para a eliminação destes grupos se faz necessário.

Foi estudado o uso de técnicas de aprendizado de máquina, como redes neurais ou SVM (*support vector machine*), utilizando as características utilizadas pelas etapas de seleção utilizadas neste trabalho, porém o resultado não foi satisfatório e o estudo não continuou. Uma última sugestão para um trabalho futuro seria desenvolver melhor uma rede neural capaz de diferenciar grupos de caracteres de outros grupos, e poderia fazer o papel de todas as etapas de seleção, ou complementá-las.

Referências Bibliográficas

BEUCHER, S. Numerical residues. **Image and Vision Computing**, v. 25, n. 4, p. 405-415, 2007.

CANNY, J. A computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. PAMI-8, n. 6, p. 679-698, 1986.

CASEY, R. G.; LECOLINET, E. A Survey of Methods and Strategies in Character Segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 18, n. 7, Julho 1996. 690-706.

CHEN, D.; BOURLARD, H.; THIRAN, J.-P. Text identification in complex background using SVM. **Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.**, v. 0, n. C, p. 621-626, 2001.

CHEN, D.; ODOBEZ, J.-M.; THIRAN, J.-P. A localization/verification scheme for finding text in images and video frames based on contrast independent features and machine learning methods. **Signal Processing: Image Communication**, v. 19, n. 3, p. 205-217, 2004.

CHEN, Y.-L.; WU, B.-F. A multi-plane approach for text segmentation of complex document images. **Pattern Recognition**, v. 42, n. 7, p. 1419-1444, 2009.

CLARK, P.; MIRMEHDI, M. Finding text regions using localised measures. **Proceedings of the 11th British Machine Vision Conference**, p. 675-684, 2000.

COMANICIU, D.; MEER, P. Robust analysis of feature spaces: Color image segmentation. **Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 1997. 750-755.

FABRIZIO, J.; MARCOTEGUI, B.; CORD, M. Text segmentation in natural scenes using Toggle-Mapping. **IEEE International Conference on Image Processing**, 2009. 2373-2376.

GONZALES, R. C.; WOODS, R. E. **Digital Image Processing**. 2a. Edição. ed. New Jersey: Prentice Hall, 2002. ISBN ISBN: 0-201-18075-8.

GUINGO, B. C.; STIEBLER, G.; THOMÉ, A. C. CONCEPÇÃO E DESENVOLVIMENTO DE UM SISTEMA DE RECONHECIMENTO AUTOMÁTI-CO DE PLACAS DE VEÍCULOS AUTOMOTORES. **XV Congresso Brasileiro de Automática**, 2004.

HAYKIN, S. **Neural Networks - A Comprehensive Foundation**. 2a. Edição. ed. [S.l.]: Pearson, 1999.

HOROWITZ, S. L.; PAVLIDIS, T. Picture segmentation by a tree traversal algorithm. **Journal of the ACM (JACM)**, v. 23, n. 2, p. 368-388, 1976.

JAFRI, S. A. R.; BOUTIN, M.; DELP, E. J. AUTOMATIC TEXT AREA SEGMENTATION IN NATURAL IMAGES. **IEEE International Conference on Image Processing**, 2008. 416.

JUNG, K.; KIM, K. I.; JAIN, A. K. Text Information Extraction in Images and Videos: A Survey. **Pattern Recognition**, v. 37, n. 5, p. 977-997, 2004.

LEBOURGEIS, F. Robust multifont OCR system from gray level images. **Proceedings of the Fourth International Conference on Document Analysis and Recognition**, 1997.

LEE, S.-W.; LEE, D.-J.; PARK, H.-S. A new methodology for gray-scale character segmentation and recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 18, n. 10, p. 1045-1050, 1996.

LIENHART, R.; STUBER, F. Automatic text recognition in digital videos. **Proceedings of SPIE**, 1996. 180-188.

LIU, X.; SAMARABANDU, J. Multiscale edge-based text extraction from complex images. **IEEE International Conference on Multimedia and Expo**, 2006. 1721-1724.

LUCAS, S. M. **ICDAR 2005 text locating competition results**. Proceedings of Eighth International Conference on Document Analysis and Recognition. [S.l.]: [s.n.]. 2005. p. 80-84.

LUCAS, S. M. et al. **ICDAR 2003 robust reading competitions**. Proceedings on the Seventh International Conference on Document Analysis and Recognition. [S.l.]: [s.n.]. 2003. p. 682-687.

LUCAS, S. M. et al. **ICDAR 2003 robust reading competitions:** entries, results, and future directions. International Journal of Document Analysis and Recognition (IJDAR). [S.l.]: [s.n.]. 2005. p. 105-122.

LYU, M. R.; SONG, J.; CAI, M. A comprehensive method for multilingual video text detection, localization, and extraction. **IEEE Transactions on Circuits and Systems for Video Technology**, v. 15, n. 2, p. 243-255, 2005.

MARTINSKY, O. **Algorithmic and mathematical principles of automatic number plate recognition systems**. [S.l.]: BRNO UNIVERSITY OF TECHNOLOGY, 2007.

MORI, S.; SUEN, C. Y.; YAMAMOTO, K. Historical review of OCR research and development. **Proceedings of the IEEE** , v. 80, n. 7, p. 1029-1058, 1992. ISSN ISSN: 00189219.

NOURBAKHSH, F.; PATI, P. B.; RAMAKRISHNAN, A. G. Text localization and extraction from complex gray images. **Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)**, 2006. 776-785.

OTSU, N. A threshold selection method from gray-level histograms. **IEEE transactions on Systems Man and Cybernetics**, 1979. 62-66.

QI, K. Z. et al. Using Adaboost to Detect and Segment Characters from Natural Scenes. **Workshop on Camera-based Document Analysis and Recognition**, Agosto 2005.

RETORNAZ, T.; MARCOTEGUI, B. Scene text localization based on the ultimate opening. **International Symposium on Mathematical Morphology**, 2007. 177-188.

SAMARABANDU, J.; LIU, X. An edge-based text region extraction algorithm for indoor mobile robot navigation. **IEEE International Conference on Mechatronics and Automation**, 3, n. 4, 2005. 701-706.

SATO, T. et al. Video OCR: indexing digital news libraries by recognition of superimposed captions. **Multimedia Systems**, v. 7, n. 5, p. 385-395, 1999.

SHAPIRO, V. et al. Adaptive license plate image extraction. **Proceedings of the 5th international conference on Computer systems and technologies - CompSysTech**, 2004.

SULLIVAN, G. J.; BAKER, R. L. Efficient Quadtree Coding of Images and Video. **IEEE Transactions on Image Processing**, 3, Maio 1994. 327-331.

TSUJIMOTO, S.; ASADA, H. Major components of a complete text reading system. **Proceedings of the IEEE**, p. 1133-1149, 1992.

WIKIPEDIA. Optical Character Recognition. **Wikipedia**. Disponivel em: <http://en.wikipedia.org/wiki/Optical_character_recognition>. Acesso em: 07 fev. 2011.

WU, V.; MANMATHA, R.; RISEMAN, E. M. Finding Text in Images. **Proceedings of the second ACM international conference on Digital libraries - DL '97**, p. 3-12, 1997.

WU, X. Adaptive Split-and-Merge Segmentation Based on Piecewise Least-Square Approximation. **IEEE Transaction on Pattern Analysis and Machine Intelligence**, 15, n. 8, Agosto 1993. 808-815.

YUILLE, A. L.; CHEN, X. Detecting and reading text in natural scenes. **Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 2004. 366-373.

ZHANG, D.-Q.; CHANG, S.-F. Learning to detect scene text using a higher-order MRF with belief propagation. **Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on**, 2004.

ZHANG, Y.; ZHANG, C. A new algorithm for character segmentation of license plate. **Proceedings of IEEE IV2003 Intelligent Vehicles Symposium**, 2003. 106-109.

Anexo I – Conjunto de exemplo do ICDAR e seus resultados

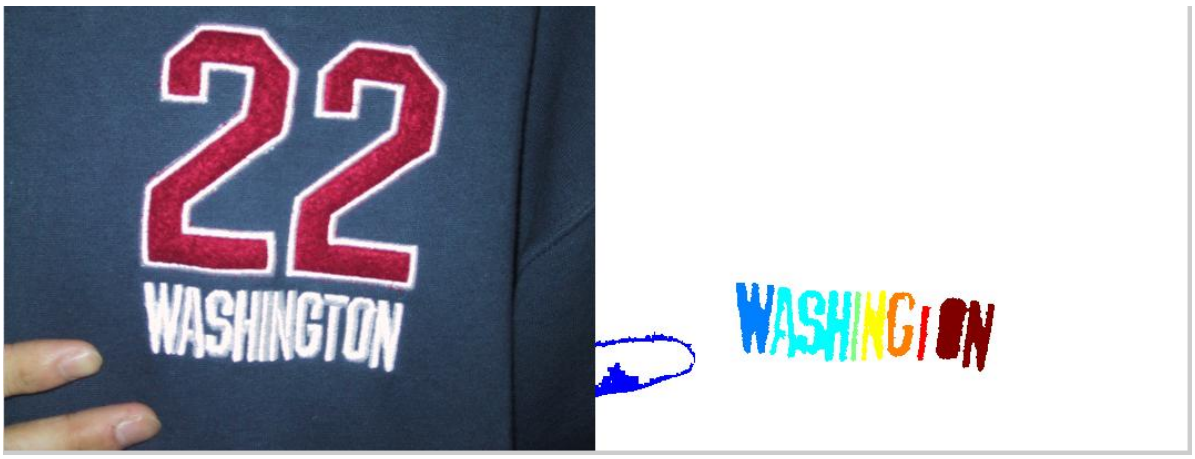

Figura I.1 - Imagem e resultado da imagem 1 do ICDAR

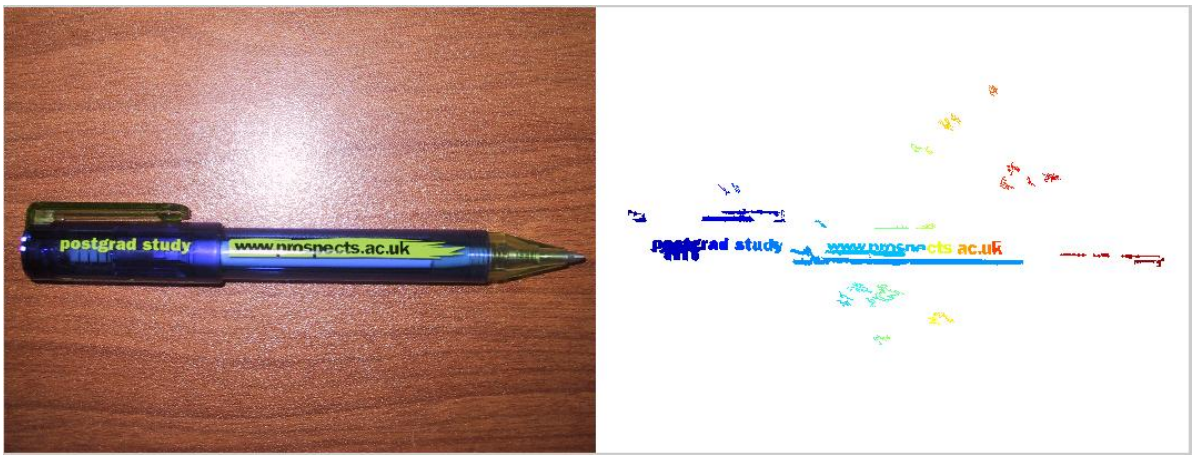

Figura I.2 - Imagem e resultado da imagem 2 do ICDAR

Figura I.3 - Imagem e resultado da imagem 3 do ICDAR

Figura I.4 - Imagem e resultado da imagem 4 do ICDAR

Figura I.5 - Imagem e resultado da imagem 5 do ICDAR

Figura I.6 - Imagem e resultado da imagem 6 do ICDAR

Figura I.7 - Imagem e resultado da imagem 7 do ICDAR

Figura I.8 - Imagem e resultado da imagem 8 do ICDAR

Figura I.9 - Imagem e resultado da imagem 9 do ICDAR

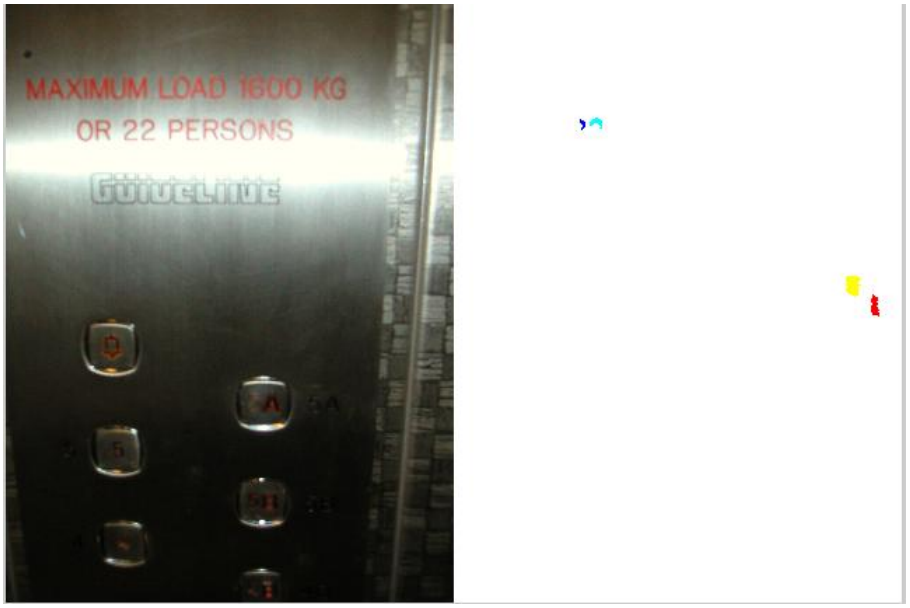

Figura I.10 - Imagem e resultado da imagem 10 do ICDAR

Figura I.11 - Imagem e resultado da imagem 11 do ICDAR

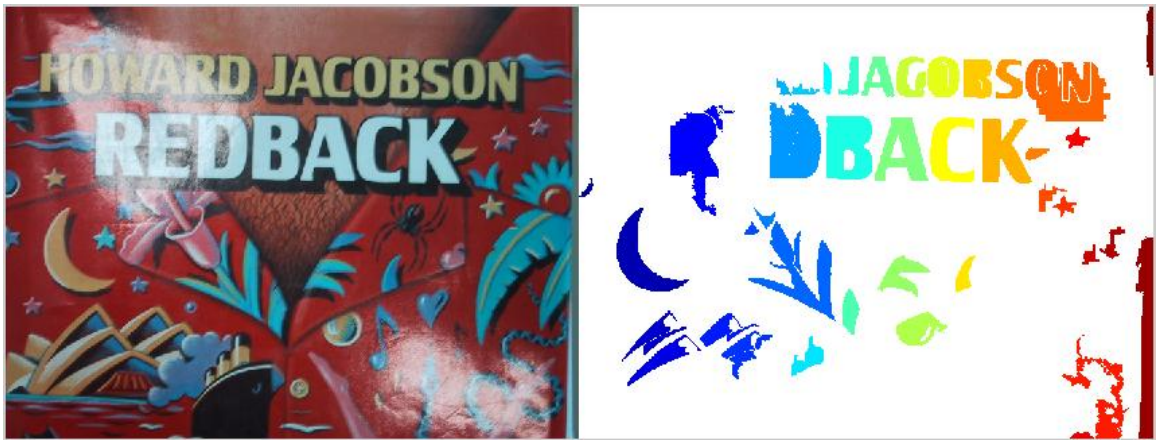

Figura I.12 - Imagem e resultado da imagem 12 do ICDAR

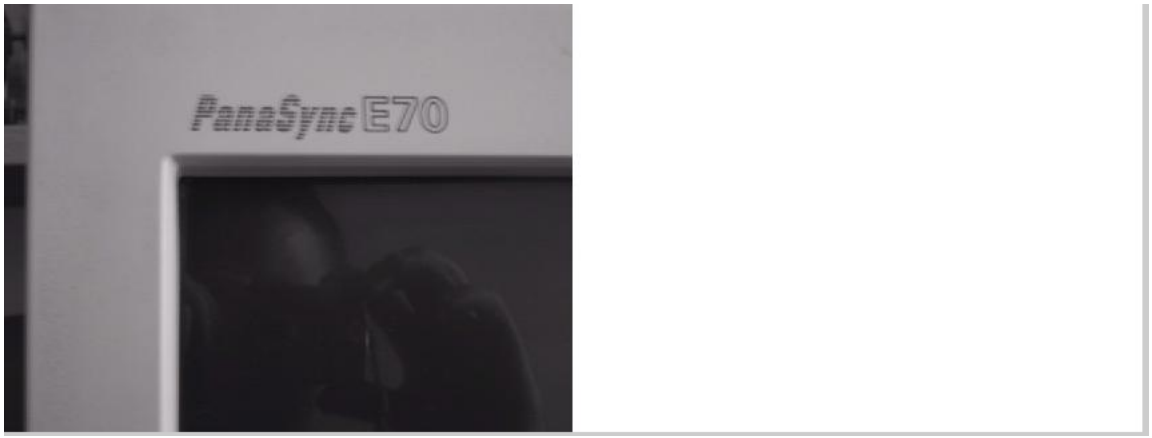

Figura I.13 - Imagem e resultado da imagem 13 do ICDAR

Figura I.14 - Imagem e resultado da imagem 14 do ICDAR

Figura I.15 - Imagem e resultado da imagem 15 do ICDAR

Figura I.16 - Imagem e resultado da imagem 16 do ICDAR

Figura I.17 - Imagem e resultado da imagem 17 do ICDAR

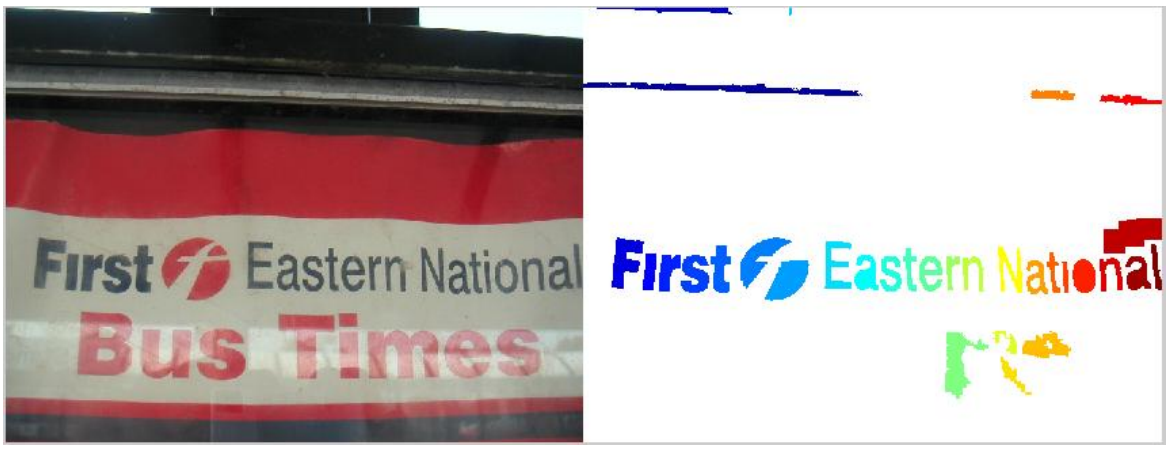

Figura I.18 - Imagem e resultado da imagem 18 do ICDAR

Figura I.19 - Imagem e resultado da imagem 19 do ICDAR

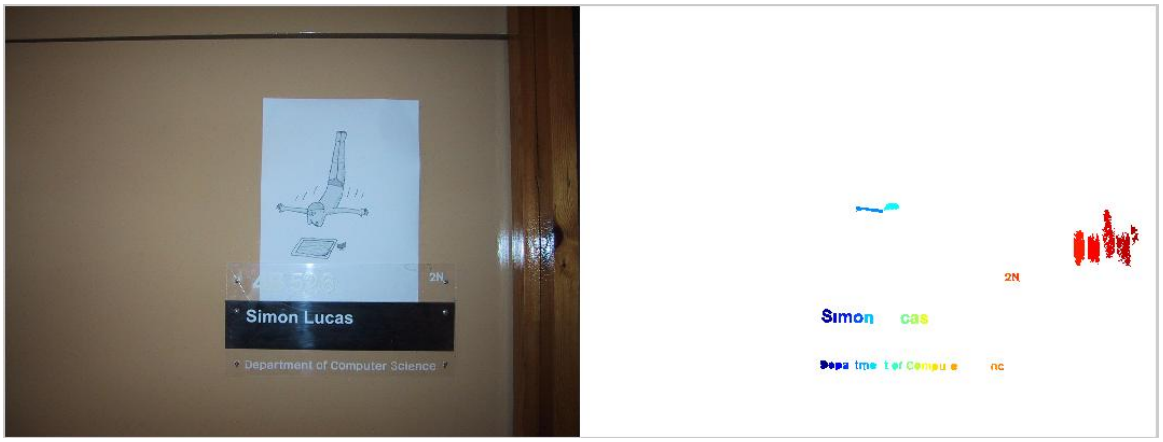

Figura I.20 - Imagem e resultado da imagem 20 do ICDAR