

UNIVERSIDADE DE CAXIAS DO SUL
Centro de Computação e Tecnologia da Informação
Curso de Bacharelado em Ciência da Computação

Cássio Pinheiro Almeron

ESTUDO EM PROCESSAMENTO DE IMAGENS DIGITAIS
APLICADO AO RECONHECIMENTO DE PLACAS DE VEÍCULOS
AUTOMOTORES BRASILEIROS

Caxias do Sul

Dezembro de 2010

Cássio Pinheiro Almeron

**ESTUDO EM PROCESSAMENTO DE IMAGENS DIGITAIS
APLICADO AO RECONHECIMENTO DE PLACAS DE VEÍCULOS
AUTOMOTORES BRASILEIROS**

Trabalho de Conclusão de
Curso para obtenção do
Grau de Bacharel em
Ciência da Computação da
Universidade de Caxias do
Sul.

**Carlos Eduardo Nery
Orientador**

Caxias do Sul

Dezembro de 2010

**Dedico este trabalho em especial ao meu irmão
Giusepe Pinheiro Almeron (*in memoriam*).**

RESUMO

Este trabalho apresenta a pesquisa e o desenvolvimento de um método para identificação dos caracteres de placas de licenciamento de veículos automotores através de imagens digitais. Esta identificação é realizada através de processamento digital de imagens. São abordados conceitos e algumas técnicas para análise e modificação de uma imagem, assim como os princípios da Morfologia Matemática. Também é descrito em detalhes todas as etapas do método proposto, sendo estas: a localização da placa do veículo; após a segmentação dos caracteres; e por último a identificação destes caracteres através de uma base de caracteres. Além de uma etapa extra que é o treinamento desta base de caracteres, a fim de tornar o processo de identificação mais assertivo. Por fim, este trabalho aborda resultados estatísticos obtido em testes.

Palavras-chaves: Processamento de Imagens, Morfologia Matemática, Reconhecimento de Padrões, Identificação de Placas

ABSTRACT

This study presents research and development of a method that identify characters in a car's license plate by digital images. This identification is made with digital image processing. Concepts and some techniques are studied to analysis and treatments with principles of Mathematical Morphology for image manipulation. The steps of proposed method are also described, which are: the location of the car's license plate after the characters' segmentation, and finally the identification of these characters through a character's database. In addition to an extra step, that is the training of basic character in order to make the identification process more reliable. Finally, the statistical results obtained in testing is presented.

Key words: Image processing, mathematical morphology, Pattern Recognition, Plate license identification

LISTA DE FIGURAS

Figura 1: Imagem Matricial	16
Figura 2: Imagem Vetorial	16
Figura 3: Exemplo de uma imagem sintetizada.....	16
Figura 4 convenção dos eixos em uma imagem matricial.....	17
Figura 5: Ilustração da mistura de cores aditivas e subtrativas	18
Figura 7: Escala de cinza no cubo RGB	19
Figura 10: Imagem com pouco contraste linear	22
Figura 11: Imagem anterior equalizada pela transformação da fórmula	22
Figura 12: Exemplos de limialização	23
Figura 13: Perfil de Linha.....	25
Figura 14: (a) Os conjuntos (b) União (c) Intersecção (d) Complemento (e) Diferença Fonte: <i>Gonzalez, 2002</i>	28
Figura 16: Exemplo de dilatação	30
Figura 18: Demonstração de abertura.....	31
Figura 19: Demonstração de fechamento	32
Figura 20: Demonstração do processo de extração de contornos.....	33
Figura 21: Demonstração do Processo de Preenchimento de Regiões.....	34
Figura. 22: Demonstração do processo de extração de componentes conectados.....	35
Figura 23: Fluxograma do método de reconhecimento de uma placa de veículo proposto	38
Figura 25: Perfil de linha com os pontos de máximo e mínimo destacados	42
Figura 26: Linha recortada a partir do perfil de linha com maior número de pontos máximos e mínimos.....	42
Figura 27: Placa localizada e segmentada	42
Figura 28: Placa binarizada	44
Figura 29: Caracteres da placa identificados e segmentados	45
Figura 30: Representação gráfica de uma matriz na base de caracteres, onde em preto é igual a 1 e em branco a 0	46
Figura 31: Resultado final do processamento do método proposto.....	47
Figura 32: Percentual de Certeza.....	48
Figura 33: Imagem que não teve acerto na identificação dos caracteres.....	48
Figura 34: Treinamento da base de caracteres.....	49
Figura 35: Imagem reconhecida corretamente após treinamento do software	50
Figura 35: Sucesso ao segmentar a placa	53
Figura 36: falha na localização da linha da placa.....	54
Figura 37: sucesso ao encontrar a linha e falha ao segmentar a placa.....	54
Figura 39: Falha ao localizar a linha devido ao excesso de ruído na imagem	55
Figura 40: Sucesso ao segmentar os caracteres	57
Figura 41: Sucesso ao segmentar os caracteres em imagem escura	58
Figura 42: Falha ao segmentar os caracteres devido ao pouco contraste entre a placa e o fundo.....	58
Figura 43: Falha ao segmentar os caracteres devido ao detalhe do parafuso	59
Figura 44: Falha ao segmentar os caracteres devido a má qualidade da placa.....	60
Figura 45: Sucesso ao reconhecer os caracteres	61
Figura 46: Falha ao reconhecer os caracteres com grau de certeza elevado	61

Figura 47: Falha ao reconhecer os caracteres	62
Figura 48: Percentual de certeza mais baixo entre as amostras.....	62
Figura 49: Percentual de certeza mais elevado entre as amostras	63

LISTA DE TABELAS

Tabela 1: Operações Lógicas Básicas.....	29
Tabela 2: Resultados Estatísticos da Localização e Segmentação da Placa	53
Tabela 3: Resultados Estatísticos da Segmentação dos Caracteres	56
Tabela 4: Resultados Estatísticos do Reconhecimento dos caracteres	60
Tabela 5: Resultados Completos das Amostras	68

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado em Português	Significado em Inglês
MM	Morfologia Matemática	
MMB	Morfologia Matemática Binária	
RGB	Vermelho, Verde, Azul	Red, Green, Blue
CMY	Ciano, Magenta, Amarelo	Cyan, Magenta, Yellow
CMYK	Ciano, Magenta, Amarelo, Preto	Cyan, Magenta, Yellow, Black
HSI	Tom, Saturação, Intensidade	Hue, Saturation, Intensity

SUMÁRIO

1	Introdução	12
1.1	Processamento Digital de Imagens.....	12
1.2	Motivação	13
1.3	Organização dos Capítulos	14
2	imagem digital.....	15
2.1	A Imagem Digital	15
2.2	Sistemas de cores primárias	17
2.3	Padrão RGB.....	18
2.4	ImageNS em Tons de Cinza	19
2.5	Histograma	20
2.6	Contraste Linear	21
2.7	Limiarização	22
2.7.1	Ponto Limiar Ótimo	23
2.8	Perfil de linha	24
3	Morfologia Matemática.....	26
3.1	Teoria dos Conjuntos	26
3.1.1	União.....	26
3.1.2	Intersecção	27
3.1.3	Complemento.....	27
3.1.4	Diferença.....	27
3.2	Morfologia Matemática Binária	28
3.3	Dilatação.....	29
3.4	Erosão.....	30
3.5	Abertura.....	31
3.6	Fechamento	31
3.7	Alguns Algoritmos Morfológicos.....	32
3.7.1	Extração de contornos.....	32
3.7.2	Preenchimento de Regiões.....	33
3.7.3	Extração de Componentes Conectados.....	34
3.7.4	Demais algoritmos	35
3.8	Morfologia Matemática em tons de cinza	36
4	PROPLACAS.....	37
4.1	Recursos Tecnológicos	37
4.2	Método Proposto	38
4.3	Pré-processamento.....	38
4.3.1	Verificação do Tamanho da Imagem.....	39
4.3.2	Conversão da imagem para tons de cinza	39
4.4	Localização e segmentação da Placa.....	40
4.5	Segmentação dos Caracteres	43
4.5.1	Verificação do contraste linear.....	43
4.5.2	Limiarização	44
4.5.3	Segmentação dos caracteres.....	44
4.6	Reconhecimento dos caracteres.....	45
4.6.1	Base de caracteres	45
4.6.2	Reconhecimento de caracteres.....	46

4.6.3	Treinamento da base de caracteres.....	48
4.7	Arquitetura do Software	50
4.7.1	TCC.dll.....	50
4.7.2	Proplacas.dll.....	51
4.7.3	Proplacas.exe	51
5	Testes e Resultados	52
5.1	Aquisição das imagens	52
5.2	Localização da placa	52
5.3	Segmentação e identificação dos caracteres das placas	55
5.3.1	Segmentação	56
5.3.2	Reconhecimento.....	60
6	Conclusões gerais.....	64
7	Referências.....	66

1 INTRODUÇÃO

1.1 PROCESSAMENTO DIGITAL DE IMAGENS

A área de Processamento Digital de Imagens é um objeto de estudo com um crescente interesse por viabilizar um grande número de aplicações, segundo Marques Filho (1999), em duas categorias distintas: (1) o aprimoramento de informações gráficas para possibilitar a interpretação humana; (2) e a análise e processamento automático por computadores.

Uma das primeiras aplicações de processamento de imagens ocorreu no início do século XX por volta dos anos 20, onde se buscava formas de melhorar a qualidade de impressão das imagens digitalizadas que eram transmitidas através de um cabo submarino entre Londres e Nova Iorque (Gonzalez, 2002).

O grande impulso da área de Processamento de Imagens ocorreu três décadas mais tarde, com o advento dos primeiros computadores digitais de grande porte e o início do programa espacial norte-americano. O uso de técnicas computacionais para o aprimoramento de imagens teve início em 1964, quando imagens da lua eram transmitidas por uma sonda espacial, processadas por um computador para corrigir diversos tipos de distorções inerentes à câmera acoplada à sonda. Essas técnicas serviram de base para o aprimoramento de outros métodos de restauração de imagens para outros programas espaciais posteriores (Marques Filho, 1999).

Desde 1964 até os dias atuais, a área de processamento de imagens vem tendo um crescimento expressivo em todos os ramos da atividade humana. Por exemplo, na Medicina, o uso de imagens nos diagnósticos médicos tornou-se rotineiro, e os avanços do processamento de imagens vêm permitindo o desenvolvimento de equipamentos mais modernos, com maior facilidade de interpretação. Na Biologia, possibilita a contagem automática de células presentes em uma imagem microscópica, que seria uma tarefa demorada e repetitiva sem este recurso e com alto grau de precisão. Outro exemplo é o uso de robôs dotados de visão artificial em tarefas como o controle da qualidade nas linhas de produção industriais. O processamento automático de imagens captadas por satélites tem auxiliado os trabalhos nas áreas de Geografia, Geoprocessamento, Meteorologia e Sensoriamento Remoto, assim como técnicas de

restauração de imagens em fotos borradas de artefatos antigos, na Arqueologia. Entre outras há inúmeras aplicações distintas na Astronomia, Segurança, Publicidade e outras áreas (Marques Filho, 1999).

1.2 MOTIVAÇÃO

Inicialmente, uma solução de identificação dos caracteres de uma placa de um veículo automotivo pode parecer bastante simples, pois o objeto de identificação são os caracteres da placa. Até pode-se pensar que uma solução de OCR convencional já poderia atingir o objetivo. No entanto a realidade mostra que apenas uma solução de OCR não apresenta resultados satisfatórios porque estas soluções trabalham apenas em situações muito controladas, como um documento de texto (Cardoso, 2006). Enquanto uma imagem com uma placa de veículo possui características especiais e situações diversas como, por exemplo: neblina, chuva, luminosidade, sombra, distância, deslocamento e até mesmo o estado de conservação da placa, que tornam o processo de reconhecimento muito mais complexo (Carvalho, 2006). Em muitos casos não se encontram em um ambiente muito controlado, pois possui em vários casos cenários de fundo e riquezas em detalhes, que geram ruídos.

Tendo essas informações como base, neste trabalho é proposto o estudo de uma solução para a identificação automática dos caracteres de uma placa de veículo automotor, que é a principal informação a ser considerada, por um sistema de informação, através de imagens obtidas do mundo real e digitalizadas, utilizando-se de técnicas de processamento de imagens.

Uma solução para reconhecimento automático de placas de veículo pode ser aplicada na automatização do processo de identificação de veículos. Como para o monitoramento de trânsito, automação no processo de identificação automática de veículos furtados, aplicação de multas, controle de acessos de veículos a estacionamentos, empresas com alto fluxo de carros, caminhões e veículos de outras categorias, e inclusive ser subsídio para geração de relatórios estatísticos de tráfego (Carvalho, 2006).

Apesar da identificação manual dos caracteres ser uma atividade relativamente simples, está limitada à atividade humana. A digitação em uma grande escala de fluxo de veículos pode se tornar um processo lento e passível de falhas, ou até mesmo,

inviável (Cardoso, 2006).

1.3 ORGANIZAÇÃO DOS CAPÍTULOS

O Capítulo 1 apresenta a introdução desta dissertação, abrange um pouco da história da área de Processamento de Imagens, considerações preliminares e os objetivos do trabalho.

O Capítulo 2 aborda a fundamentação teórica com conceitos sobre imagem digital e algumas técnicas de tratamento de imagem as quais se utilizou para o desenvolvimento do método proposto.

No Capítulo 3 é apresentada a fundamentação teórica introdutória quanto à Morfologia Matemática, aborda sua base, suas operações e apresenta alguns algoritmos morfológicos.

O Capítulo 4 traz o método proposto para este trabalho, e descreve em detalhes todas as etapas do mesmo, a partir da localização da placa de um veículo automotor, a segmentação dos caracteres e o reconhecimento dos mesmos, bem como o treinamento da base de caracteres e por fim um pouco da arquitetura do software.

Sucessivamente no Capítulo 5, é realizada uma análise dos resultados obtidos a partir de uma amostragem, separados pela localização da placa, segmentação e o reconhecimento dos caracteres, assim como a apresentação de alguns casos bem sucedidos tanto quanto casos de falhas.

Por fim, o Capítulo 6 abrange a conclusão final desta dissertação, traz as observações finais, considerações sobre os assuntos abordados, limitações, e restrições do trabalho e possibilidades para continuidade.

2 IMAGEM DIGITAL

Este capítulo descreve de forma sumária o que é uma imagem digital e as técnicas utilizadas no desenvolvimento do software proposto para este trabalho de conclusão de curso. Inicia descrevendo como é estruturada uma imagem digital. Descreve-se o que é um sistema de cores primárias e o padrão RGB, sistema utilizado nas imagens ao longo deste trabalho. Em seguida aborda-se o padrão de imagens em tons de cinza e a conversão a partir de imagens coloridas. Posteriormente são apresentadas as técnicas de transformação de imagem e de obtenção de informações, as quais são fundamentais para o desenvolvimento do software final, tais como o Histograma, o Contraste Linear e a Limiarização de imagens.

2.1 A IMAGEM DIGITAL

As imagens são uma representação gráfica de pessoa ou objeto formada por um conjunto de pontos no espaço, os quais convergem ou divergem raios de luz originado de um objeto luminoso ou iluminado que passam através de um sistema óptico. (Manfio, 2004)

Uma imagem digital também é formada através do mesmo processo. Porém, o conjunto de pontos que a formam possui uma denominação específica. Tais pontos são denominados de *pixel* (*Picture Element*). (Gonzalez, 2002)

A imagem digital possui características como resolução e número de cores, podendo ser representada ou processada em forma matricial ou vetorial. A imagem matricial é formada por uma série ordenada de pixels dispostos continuamente em linhas e colunas. A Figura 1 apresenta na imagem (A) uma reta no tamanho original, e na imagem (B) a ampliação da mesma reta, e demonstra que quando ampliada, a reta perde a definição. Já a imagem vetorial é composta por uma série de instruções matemáticas e a imagem apresentada é gerada a partir dos cálculos dessas instruções, neste formato a imagem pode ser ampliada sem perder a definição, pois serão calculados novamente as instruções para o novo tamanho, conforme demonstrado na Figura 2 (Manfio, 2004).

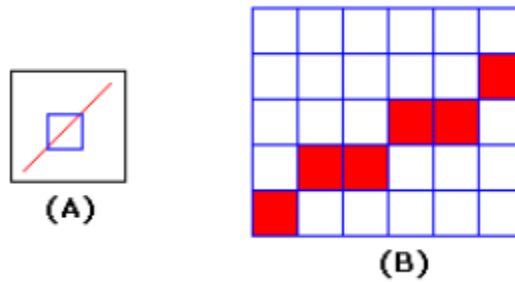


Figura 1: Imagem Matricial

Fonte: *Manfio, 2004*

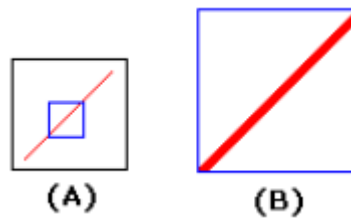


Figura 2: Imagem Vetorial

Fonte: *Manfio, 2004*

A imagem pode resultar de uma captura do mundo real através de dispositivos de entrada como scanners ou câmeras digitais, ou podem ser sintetizadas em um computador através de softwares de desenho ou captura de tela. Após a digitalização as imagens podem ser manipuladas através de softwares de edição de imagens, os quais não produzem estrutura semântica. (Willrich, 2001).



Figura 3: Exemplo de uma imagem sintetizada

Fonte: <http://galeria.colorir.com/veiculos/carros/carro-de-cidade-pintado-por-carrinho-92547.html>

Uma imagem matricial é representada por bitmaps, ou seja, uma matriz bidimensional composta por elementos denominados de pixels. O pixel é o menor elemento da resolução de uma imagem, e armazena um valor que representa o brilho

correspondente em tom de cinza. O número de bits disponíveis para armazenar este valor de brilho é denominado amplitude ou profundidade (bits por pixel). (Bittencourt, 2003)

O valor do brilho estará no intervalo de $[0, 2^b - 1]$, onde b é o número de bits por pixel utilizado para o armazenamento deste valor. Para $b=8$ a imagem poderá armazenar até 256 níveis de cinza, onde 0 é o preto e 255 é o branco. (Falcão, 2003)

A Figura 4 ilustra a convenção dos eixos utilizados nas referências bibliográficas, e também ao longo deste trabalho. Onde X representa a coluna e Y representa a linha, e define-se de $f(x,y)$ sendo o valor do brilho em tom de cinza do ponto da imagem. A origem da imagem é o ponto $f(0,0)$ e o último ponto é $f(\text{LARGURA} - 1, \text{ALTURA} - 1)$.

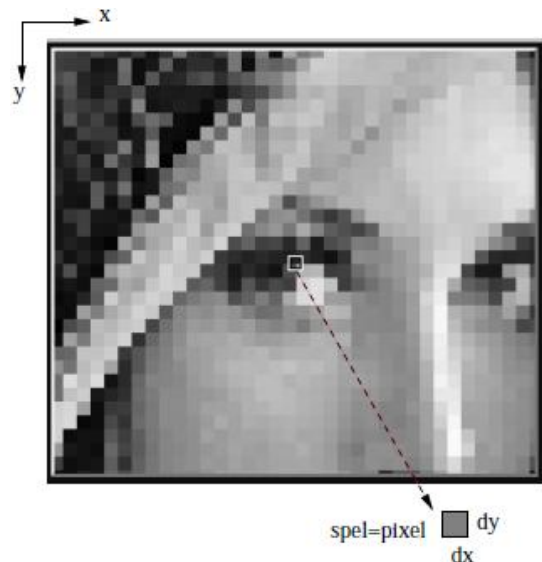


Figura 4 convenção dos eixos em uma imagem matricial

Fonte: Falcão, 2003

2.2 SISTEMAS DE CORES PRIMÁRIAS

Um sistema de cores primárias são métodos para a representação de propriedades e comportamentos de combinações em contextos particulares. Neste trabalho será apresentado apenas o sistema de cores RGB, porque todas as imagens utilizadas encontram-se neste sistema. Mas existem outros sistemas como CMY e CMYK, HSI. (Osawa, 2004)

Um sistema primário utiliza-se de algumas cores básicas, e a partir da

combinação dessas cores podem ser produzidas diversas outras cores. Podendo ser aditiva ou subtrativa, ou seja, podendo ser adicionadas ou subtraídas cores para a composição da combinação. (Osawa, 2004)

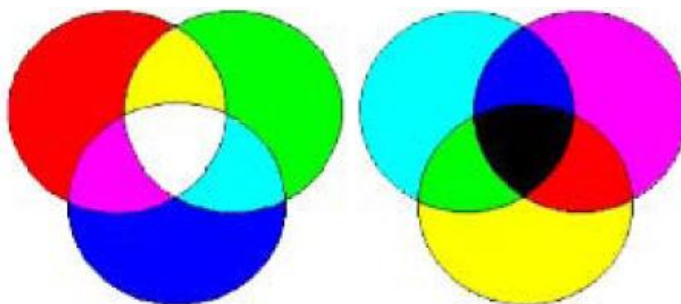


Figura 5: Ilustração da mistura de cores aditivas e subtrativas

Fonte: *Osawa, 2004*

O sistema de cores RGB que é um sistema aditivo é utilizado para a representação de luz, onde o branco é a presença máxima das três cores básicas e o preto a ausência total. Enquanto o CMY que é subtrativo é utilizado para combinações de tintas e pigmentos, é o padrão utilizado nas impressoras coloridas.

2.3 PADRÃO RGB

No modelo RGB (Red, Green, Blue) cada cor é representada a partir de uma combinação das três cores básicas de forma aditiva: Vermelho, Verde e Azul. É baseado em um sistema de coordenadas cartesianas tridimensional, como mostrado na Figura 6, onde todas as cores possíveis são pontos dentro do cubo. Por uma convenção, o domínio que os valores que cada canal pode assumir se encontra entre 0 e 1. (Gonzalez, 1992)

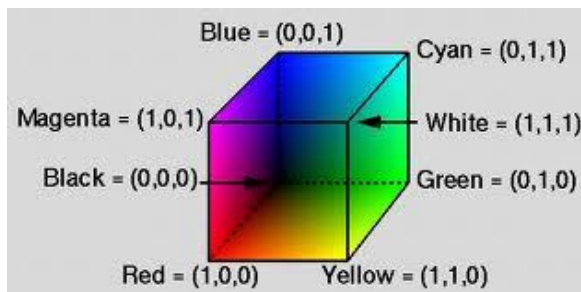


Figura 6: Cubo RGB

Considerando imagens RGB com a profundidade de cada pixel em 24 bits, estes bits serão divididos em três grupos de 8 bits, e cada grupo de bits é destinado para

armazenar o nível de cada um dos três canais. Dessa forma o número total de cores possíveis com 24 bits é $(2^8)^3 = 16.777.216$ cores. (Gonzalez, 2002)

2.4 IMAGENS EM TONS DE CINZA

Imagens em tons de cinza possuem apenas um canal de cor. As imagens utilizadas apresentam cada pixel com uma profundidade de 8 bits, onde o valor do pixel varia entre 0 e 255.

A conversão de imagens coloridas para imagens em tons de cinza é realizada a partir da seguinte fórmula (Gonzalez, 2002):

$$C = (R + G + B) / 3$$

Onde:

- C: Cinza;
- R: Nível de Vermelho;
- G: Nível de Verde;
- B: Nível de Azul;

Está fórmula representa a média dos níveis das três cores básicas. A representação desta função no cubo RGB ficaria como na Figura 7.

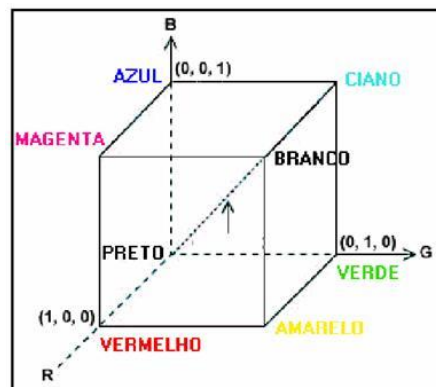


Figura 7: Escala de cinza no cubo RGB

Fonte: <http://www6.ufrgs.br/engcart/PDASR/formcor.html>

Nesta imagem a reta que inicia em preto (0,0,0) e termina em branco (1,1,1) é a reta da escala de cinza.

Aplicando essa fórmula em todos os pixels obtém-se a conversão de toda a

imagem para tons de cinza, como disposto na Figura 8 (a) (a imagem original e 8 (b) (a imagem em tons de cinza):



Figura 8: Conversão de imagem colorida em tons de cinza

Fonte: <http://www.quebarato.com.br>

2.5 HISTOGRAMA

O histograma trata-se de um conjunto de números que indica a quantidade de pixels em uma imagem que apresenta um determinado nível de cinza. Estes valores geralmente são representados em um gráfico em barras demonstrando a frequência quantitativa ou percentual dos pixels nos níveis de cinza. (Marques Filho, 1999)

Com o histograma é possível obter algumas informações da imagem quanto à qualidade, o contraste e o brilho médio.

Os elementos do histograma são calculados com a seguinte fórmula:

$$\text{Pr}(r_k) = N_k / N$$

Onde

$$0 \leq r_k \leq 1;$$

$$k = 0, 1, \dots, L - 1, \text{ onde } L \text{ é o número de níveis de cinza;}$$

n = número total de pixels na imagem;

$\text{Pr}(r_k)$ = total em percentual de pixels no K -ésimo nível de cinza;

N_k = número de pixels cujo nível de cinza corresponde a k ;

A escala de cinza utilizada está entre 0, que representa preto, e 255, que representa branco. Esta escala é a mesma utilizada nas imagens utilizadas nas ilustrações e no processamento deste trabalho.

A Figura 9 apresenta uma imagem em tons de cinza e seu respectivo histograma.

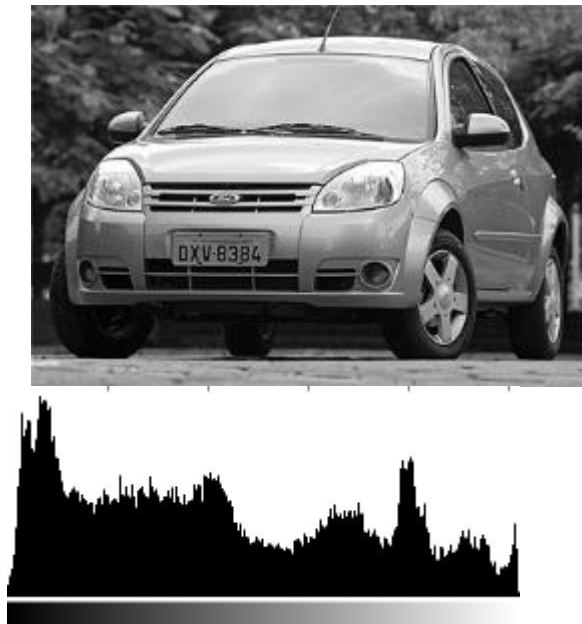


Figura 9: Imagem em tons de cinza e seu respectivo histograma de frequência

(<http://carros.uol.com.br>)

O conceito de histograma também se aplica a imagens coloridas. Para este caso deve-se decompor a imagem nas cores básicas (Vermelho, Verde, Azul) e calcular o histograma de cada componente (Marques Filho, 1999).

2.6 CONTRASTE LINEAR

A alteração do contraste linear em uma imagem é a tentativa de modificar o seu histograma com objetivo de tornar-la mais nítida. Normalmente essa técnica é aplicada a imagens que estejam escuras, provenientes de um ambiente com pouca iluminação ou falhas em suas capturas, ou seja, com pouco contraste. (Souza, 2000) E imagens escuras possuem uma grande concentração de pixels no início do histograma. (Manfio, 2004)

Com o contraste linear, a concentração de pixels em um pequeno intervalo do histograma será distribuída entre todos os níveis de cinza. Ou uma faixa pré-definida. Este processo é chamado de equalização. Para isto deve-se percorrer todos os pixels da imagem aplicando a seguinte fórmula (Souza, 2000)

$$g(x,y) = ((B - A) / (b - a)) * (f(x,y) - a) + A$$

Onde:

[A, B]: faixa de tons de cinza expandida;

[a, b]: faixa de tons de cinza original;

$f(x, y)$: valor original do tom de cinza do pixel;

$g(x, y)$: novo valor do tom de cinza do pixel;

Na Figura 10 pode se perceber uma imagem escura e com pouco contraste e ao lado o seu histograma, com a concentração dos níveis de cinza entre 0 e 60.

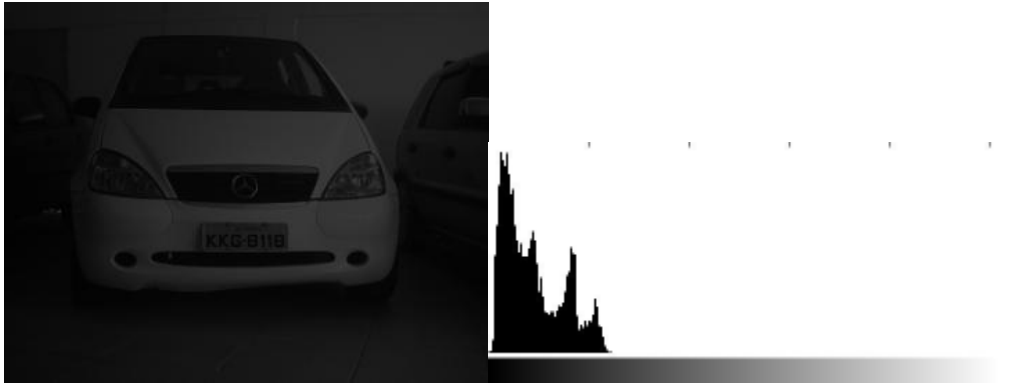


Figura 10: Imagem com pouco contraste linear

Fonte <http://www.veiculos-rs.com.br>

Na Figura 11 a equalização da mesma imagem e seu histograma com a distribuição dos níveis de cinza em todo o intervalo disponível, 0 e 255. Pode se perceber uma maior nitidez nesta imagem.

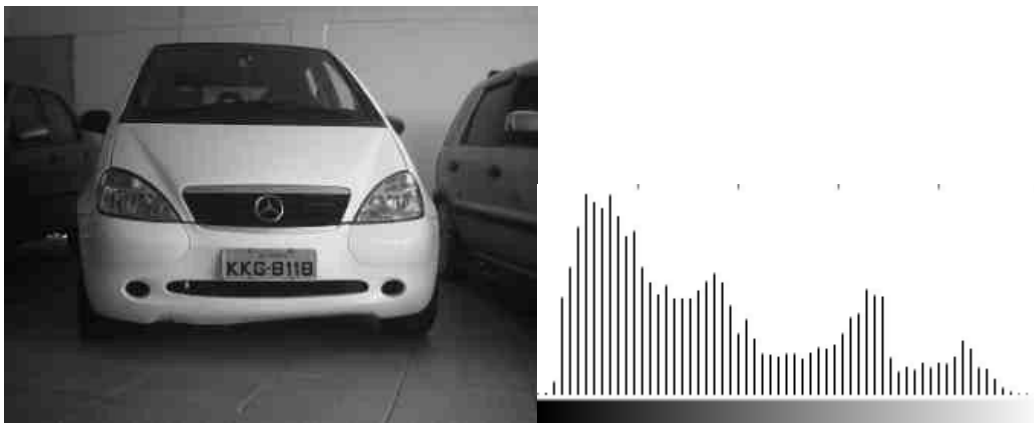


Figura 11: Imagem anterior equalizada pela transformação da fórmula

2.7 LIMIAZIZAÇÃO

A Limiarização consiste em separar regiões de uma imagem quando esta apresentar as classes fundo e objeto bem definidas. Este processo produz uma imagem binária de saída, por isso também é conhecido como binarização. (Marques Filho, 1999)

A forma mais simples de limiarizar uma imagem consiste na bipartição do histograma, convertendo os valores de tom de cinza que sejam maior ou igual a um valor limiar (T) em branco e os demais em preto. (Marques Filho, 1999)

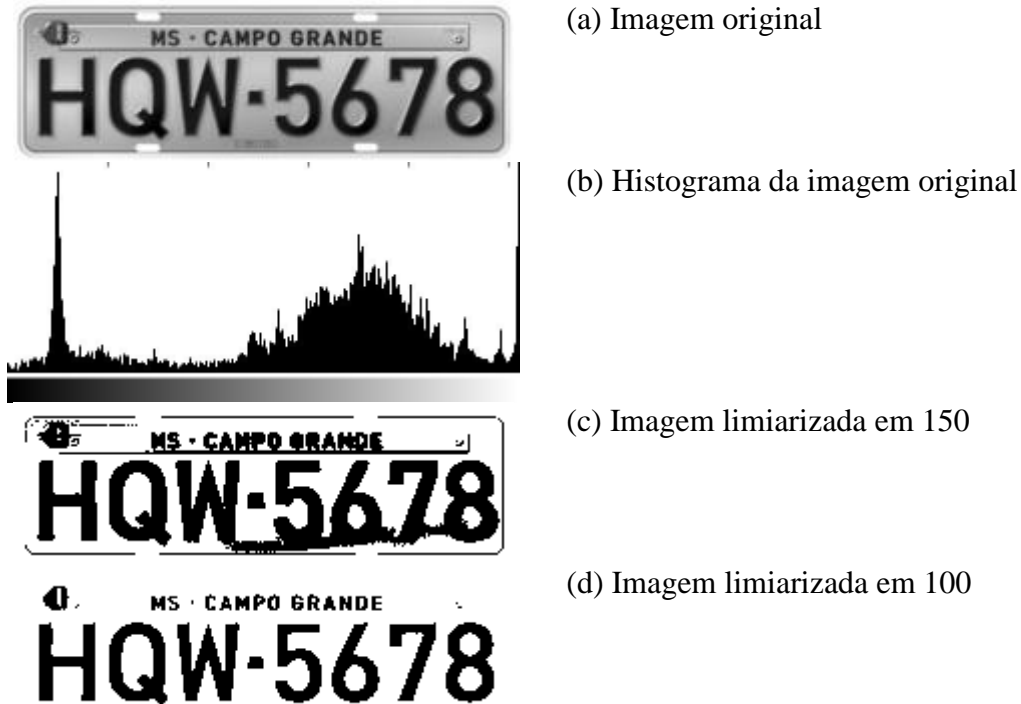


Figura 12: Exemplos de limialização

Fonte:

http://pt.wikipedia.org/wiki/Placas_de_identifica%C3%A7%C3%A3o_de_ve%C3%ADculos_no_Brasil

A limiarização pode ser descrita matematicamente com a seguinte sentença (Marques Filho, 1999):

$$g(x,y) = 1 \text{ se } f(x,y) \geq T, 0 \text{ se } f(x,y) < T$$

Onde:

$g(x,y)$: pixel da nova imagem;

$f(x,y)$: pixel da imagem original;

1: valor correspondente ao objeto;

0: valor correspondente ao fundo;

2.7.1 PONTO LIMIAR ÓTIMO

Em muitos casos o histograma de uma imagem nem sempre apresenta uma

separação bem definida entre o objeto e o fundo. E nestes casos erros de classificação são inevitáveis. O limiar ótimo tem como objetivo minimizar o máximo possível de erros na classificação.

Para determinar automaticamente o ponto limiar procura-se segmentar a imagem em duas classes diferentes pelos valores de brilho dos pixels (Mascarenhas, 1989) com a seguinte fórmula:

$$\begin{aligned}AT^2 + BT + C &= 0 \\A &= \sigma_1^2 - \sigma_2^2 \\B &= 2(\mu_1 * \sigma_1^2 - \mu_2 * \sigma_2^2) \\C &= \sigma_2^2 * \mu_2^2 - \sigma_1^2 * \mu_1^2 + 2 * \sigma_1^2 * \mu_1\end{aligned}$$

Onde:

P_1, P_2 : probabilidade das duas regiões unimodais;

μ_1, μ_2 : valores médios das regiões;

σ_1, σ_2 : desvio padrão em torno da média das duas regiões;

Dessa forma, para a determinação do limiar ótimo deve-se previamente conhecer os valores dos parâmetros de probabilidade de ocorrência, valores médios e os desvios padrões.

2.8 PERFIL DE LINHA

O perfil de linha trata-se de uma projeção da distribuição dos níveis de cinza em cada pixel contido ao longo de uma determinada linha da imagem (Cardoso, 2006).

A Figura 13 demonstra três perfis obtidos a partir da mesma imagem, em linhas diferentes.

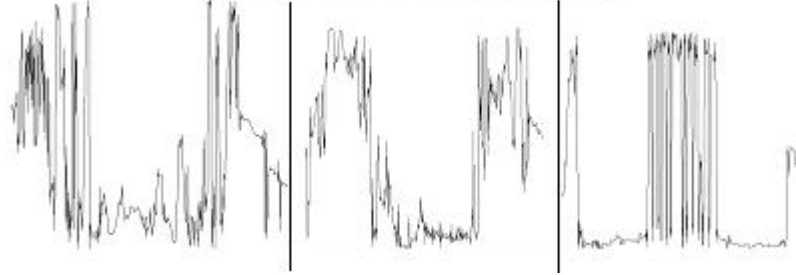


Figura 13: Perfil de Linha

3 MORFOLOGIA MATEMÁTICA

A Morfologia Matemática é o estudo de estruturas geométricas das entidades presentes em uma imagem. Consiste em extrair informações relativas à geometria e à topologia de um conjunto desconhecido, pela transformação através de um conjunto definido, o qual é chamado de elemento estruturante. Esta técnica pode ser aplicada em vários segmentos do processamento e análise de imagens com objetivos distintos como realce, filtragem, segmentação, detecção de bordas, afinamento, esqueletos morfológicos entre outras aplicações (Marques Filho, 1999).

Dentro do escopo da Morfologia Matemática (MM), o foco deste capítulo está direcionado a aplicação da MM em imagens binárias. Assim sendo, descreve-se de forma sumária a teoria da MM e se faz uma abordagem em teoria dos conjuntos, que é fundamento básico da MM Binária. Logo em seguida descreve-se a MM Binária, em seguida as operações morfológicas que são: Dilatação, Erosão, Abertura e Fechamento. Após são apresentados alguns algoritmos práticos como a Extração de Contornos, Preenchimentos de Regiões e Extração de Componentes Conectados. E finaliza abordando um pouco sobre MM em tons de cinza.

3.1 TEORIA DOS CONJUNTOS

Um conjunto é uma coleção de elementos sem repetição e sem ordem. Pode ser descrito, quando o número de elementos não for muito grande, como, por exemplo: $A = \{1, 2, 3, 4\}$, ou quando for muito grande ou infinito, através de propriedades ou restrições, demonstrado como: $X = \{x \mid x > 0\}$. (Wangenheim, 2000).

Combinação de conjuntos:

Sejam dois conjuntos $A = \{1, 2, 3, 4\}$ e $B = \{3, 4, 5, 6\}$:

3.1.1 UNIÃO

Unindo os conjuntos A e B o conjunto resultante será o conjunto que contenha todos os elementos de A e B, sendo representado da seguinte forma:

$$A \cup B = \{ x \mid (x \in A) \text{ ou } (x \in B) \}$$

Por exemplo, com os conjuntos A e B definidos anteriormente o resultado da união destes será $A \cup B = \{1, 2, 3, 4, 5, 6\}$.

3.1.2 INTERSECÇÃO

A intersecção dos conjuntos A e B será o conjunto que contenha os elementos em comum entre os dois conjuntos, sendo representado da seguinte forma:

$$A \cap B = \{ x \mid (x \in A) \text{ e } (x \in B) \}$$

Por exemplo, com os conjuntos A e B, o resultado da intersecção destes conjuntos será $A \cap B = \{3, 4\}$

3.1.3 COMPLEMENTO

O complemento de um conjunto A é o conjunto dos elementos não pertencentes a A. O complemento pode ser aplicado apenas para relações entre conjuntos e subconjuntos, onde o complemento de A são os elementos constantes em um conjunto C que contenha A, e que não estejam em A. Sendo representado da seguinte forma:

$$A^c = \{ x \mid x \notin A \}$$

Por exemplo, com o conjunto A, seja o conjunto $C = \{1, 2, 3, 4, 5\}$, o resultado do complemento será $A^c = \{5\}$

3.1.4 DIFERENÇA

A diferença de um conjunto A pelo B será o conjunto dos elementos pertencentes à A retirando-se os elementos que pertencem a B, sendo representado da seguinte forma:

$$A - B = \{ x \mid (x \in A) \text{ e } (x \notin B) \}$$

Por exemplo, com os conjuntos A e B, a diferença de A e B será $A - B = \{1, 2\}$

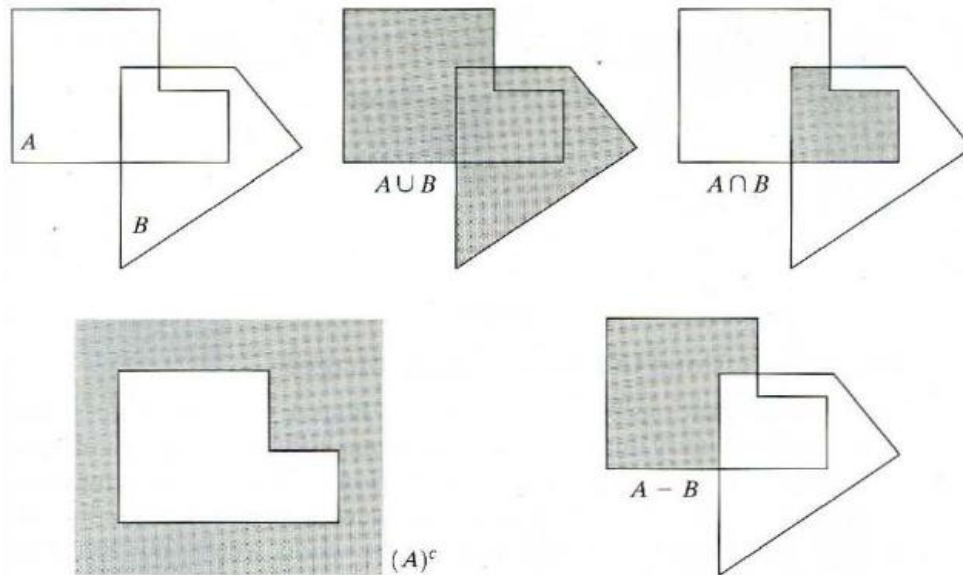


Figura 14: (a) Os conjuntos (b) União (c) Intersecção (d) Complemento (e) Diferença

Fonte: *Gonzalez, 2002*

3.2 MORFOLOGIA MATEMÁTICA BINÁRIA

A Morfologia Matemática Binária (MMB) tem como base a teoria dos conjuntos. Os conjuntos em MM representam as formas dos objetos na imagem, como o conjunto dos pixels pretos e o conjunto dos pixels brancos. A MMB consiste em aplicar operações lógicas sobre uma imagem binária utilizando-se um elemento estruturante e percorrendo-se pixel a pixel a imagem, operando no pixel corrente uma operação lógica pelo elemento estruturante. (Gonzalez, 2000)

O elemento estruturante é o conjunto utilizado para percorrer a imagem binária a ser processada, operando uma transformação morfológica. Este elemento possui um ponto que é chamado de origem.

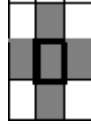


Figura 15: Elemento estruturante

(cinza representa o valor 1 e branco 0, e com contorno preto, o ponto de origem)

Mas não necessariamente o elemento de origem deverá corresponder ao ponto central do elemento estruturante. Pode ser qualquer ponto, dessa forma não é necessário que o elemento estruturante seja uma matriz quadrada de ordem ímpar.

As operações lógicas binárias ocorrem utilizando a teoria dos conjuntos, onde se pode considerar que a imagem seja um conjunto A e o elemento estruturante um conjunto B. Dessa forma percorre-se toda a imagem binária e em cada pixel é feita uma operação lógica com o elemento de origem do elemento estruturante e os pontos ao redor deste com os pixels da imagem equivalente.

Tabela 1: Operações Lógicas Básicas

a	b	a E b (*)	a OU b (+)	NÃO a
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

As operações lógicas que são utilizadas com elemento estruturante são o OU no caso da dilatação e o E para a erosão. O NÃO pode ser usado apenas sobre a imagem em processamento. A dilatação e a erosão são as duas operações fundamentais da Morfologia Matemática, as demais operações e algoritmos baseiam se nas primitivas dessas duas operações.

Nos próximos tópicos considera-se o elemento A como sendo a imagem e o elemento B como sendo o elemento estruturante, e em relação à imagem, o objeto com o valor 1 e o fundo como o valor 0.

3.3 DILATAÇÃO

A dilatação consiste em aumentar o tamanho do objeto que resulta em uma

imagem “engordada”. O seu símbolo é o \oplus . A operação de um elemento A por um elemento B é definida por:

$$A \oplus B = \{ x \mid x = a + b, a \in A, b \in B \}$$

Na imagem A, é o elemento estruturante B que irá com a sua composição definir a natureza específica da dilatação. Também pode representada pela união de $A \oplus B = A \cup B$. (Wangenheim, 2000)

Por Exemplo, seja $A = \{(0,1), (1,1), (2,1), (1,2), (2,2)\}$ e $B = \{(0,0), (1,0)\}$, o resultado da dilatação $A \oplus B$ será $\{(0,1), (1,1), (2,1), (1,2), (2,2), (3,1), (3,2)\}$, como demonstrado na Figura 16:

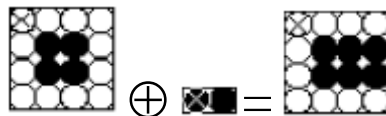


Figura 16: Exemplo de dilatação
 Fonte: Wangenheim, 2000

3.4 EROSÃO

A Erosão consiste em diminuir o objeto que resulta em uma imagem “emagrecida”. O símbolo desta operação é representado por \ominus , e a operação de um elemento A por outro elemento B é definida por:

$$A \ominus B = \{ x \mid x + b \in A \text{ para todo } b \in B \}$$

Sendo A, a imagem, e B, o elemento estruturante, pode se representar a erosão pela intersecção de $A \ominus B = A \cap B$. Por exemplo, na Figura 17 com o conjunto $A = \{(1,1), (3,3), (3,4), (4,3), (4,4)\}$ e $B = \{(0,0), (0,1), (1,0), (1,1)\}$ o resultado de $A \ominus B = \{(3,3)\}$.

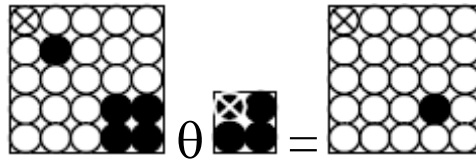


Figura 17: Exemplo de erosão

Fonte: *Wangenheim, 2000*

3.5 ABERTURA

A abertura de uma imagem binária tem por finalidade a eliminação de objetos pequenos e muito finos, quebrando-os nos seus pontos estreitos, que torna o contorno dos objetos grandes lisos e uniformes, também é aplicado para eliminação de ruídos na imagem (Wangenheim, 2000). A abertura de uma imagem A por um elemento estruturante B é definida por:

$$A \circ B = (A \oplus B) \ominus B$$

Na imagem original primeiramente ocorre operação de erosão e após, sobre o resultado da erosão, opera-se a dilatação, utiliza-se do mesmo elemento estruturante.

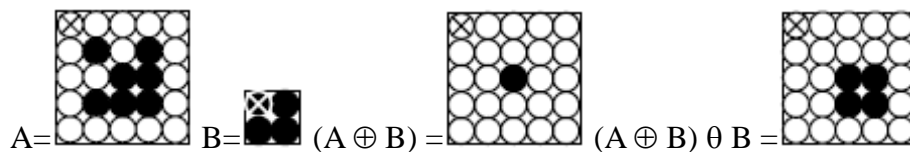


Figura 18: Demonstração de abertura

Fonte: *Wangenheim, 2000*

3.6 FECHAMENTO

Já o fechamento de uma imagem binária opera de forma oposta à abertura. Opera sobre a imagem, preenchendo buracos pequenos presentes no objeto e conectando objetos vizinhos e próximos. (Carvalho, 2006) O fechamento de uma

imagem A por um elemento estruturante B é definida por:

$$A \bullet B = (A \ominus B) \oplus B$$

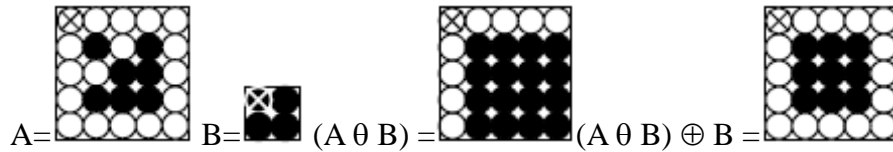


Figura 19: Demonstração de fechamento

Fonte: *Wangenheim, 2000*

Na imagem original primeiramente ocorre a operação de dilatação e após, sobre o resultado desta opera-se a erosão, com a utilização do mesmo elemento estruturante.

3.7 ALGUNS ALGORITMOS MORFOLÓGICOS

Pode-se realizar uma analogia das operações vistas nos itens anteriores com as operações básicas da matemática, onde a soma e a subtração seriam as duas operações básicas. Já a multiplicação nada mais é do que somas sucessivas e a divisão sendo subtrações sucessivas. A partir dessas quatro operações são realizados diversos cálculos de equações e estudos aprofundados na matemática. Na morfologia é semelhante, através das operações morfológicas citadas, composições de operações são realizadas, as quais podem resultar em algoritmos práticos.

Nesta sessão será apresentado o uso prático da Morfologia Matemática em processamento de imagens. No processamento de imagens binárias, a principal aplicação da MM é a extração de componentes em uma imagem que tenha utilidade na representação e descrição de formatos.

3.7.1 EXTRAÇÃO DE CONTORNOS

O contorno de um conjunto A é denotado por $\beta(A)$, e é obtido através da erosão de A pelo elemento estruturante B, e após é realizado a diferença de A por sua erosão. A extração de contornos é definida por (Marques Filho, 1999):

$$\beta(A) = A - (A \ominus B)$$

Onde A é o conjunto de elementos de uma imagem e B é o elemento estruturante, normalmente com dimensão 3×3 com todos os elementos preenchidos com valor 1.

A Figura 20 mostra o exemplo do processo de extração de contornos, onde temos a imagem A , após o elemento estruturante B , seguido da erosão de A por B e finalmente o resultado com a diferença da erosão por A .

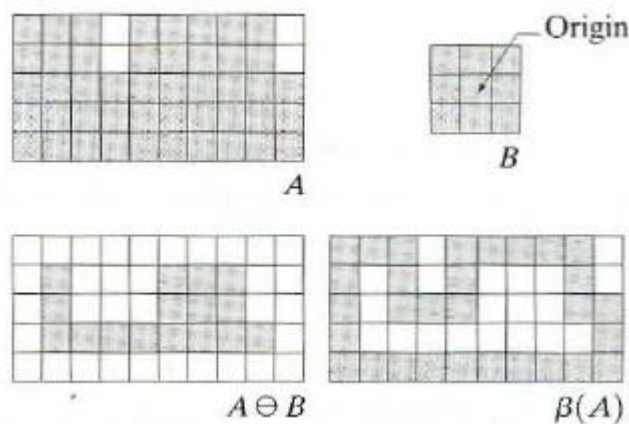


Figura 20: Demonstração do processo de extração de contornos

Fonte: *Gonzalez, 2002*

3.7.2 PREENCHIMENTO DE REGIÕES

Este algoritmo tem por objetivo preencher todos os elementos que estejam dentro de um contorno fechado com o valor 1 (Gonzalez, 2002).

Partindo-se de um ponto p , que esteja dentro deste contorno, deve se assumir que todos os pontos que não estão sobre o contorno são inicializados com o valor 0; atribui-se 1 para p , iniciando o procedimento

$$X_k = (X_{k-1} \oplus B) \cap A_c \quad (k = 1, 2, 3, \dots)$$

Onde X_0 é p . O processo termina na K -ésima iteração onde $X_k = X_{k-1}$.

A Figura 21 demonstra o processo de preenchimento de uma região dentro de um contorno, onde se tem a imagem A e o Elemento estruturante B. Em seguida o complemento de A e o preenchimento após sete iterações, ao final a união com A, finalizando o processo. O elemento estruturante para esta operação é sempre um elemento 3x3, com a origem no centro com valor 1, e os pontos paralelos à origem com valor 1.

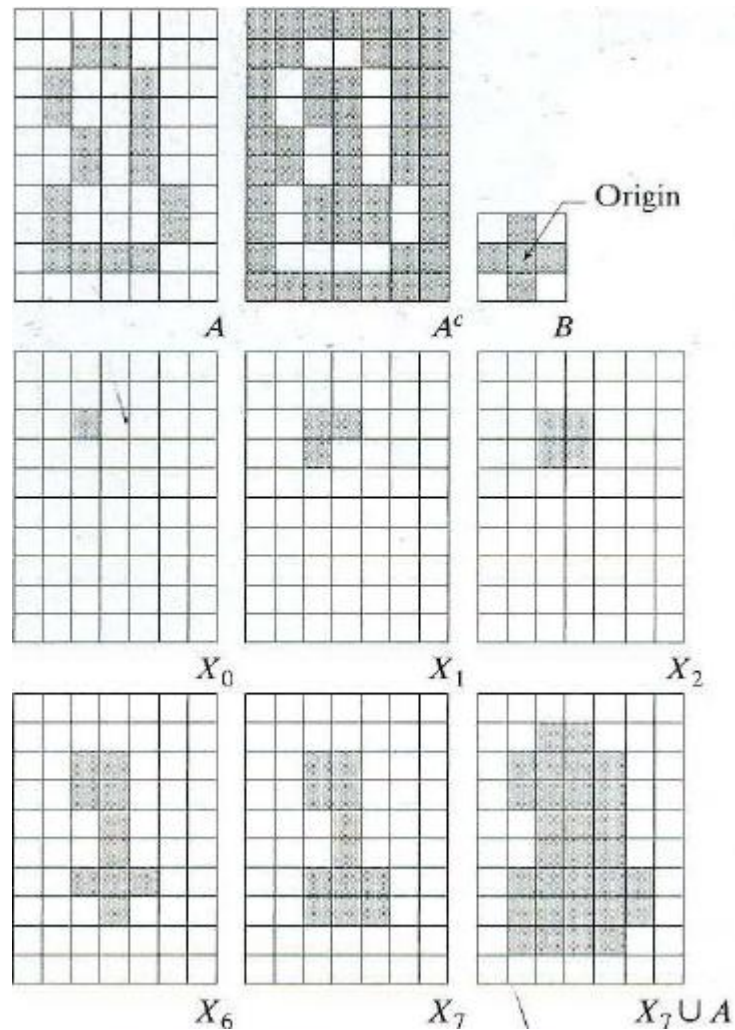


Figura 21: Demonstração do Processo de Preenchimento de Regiões

Fonte: *Gonzalez, 2002*

3.7.3 EXTRAÇÃO DE COMPONENTES CONECTADOS

Este algoritmo tem por finalidade detectar elementos em uma imagem que

estejam conectados.

Sendo Y um componente conectado em uma imagem A, parte-se de um ponto p conhecido. Dessa forma a expressão abaixo fornece todos os pontos de Y:

$$X_k = (X_{k-1} \oplus B) \cap A \quad (k = 1, 2, 3, \dots)$$

Onde $X_0 = p$ e B. O processo termina na K-ésima iteração, quando $X_k = X_{k-1}$.

A Figura 22 demonstra o processo de detecção de um elemento conectado na imagem A e o elemento estruturante B, em seguida as iterações do processo, até que todos os pontos conectados tenham sido detectados.

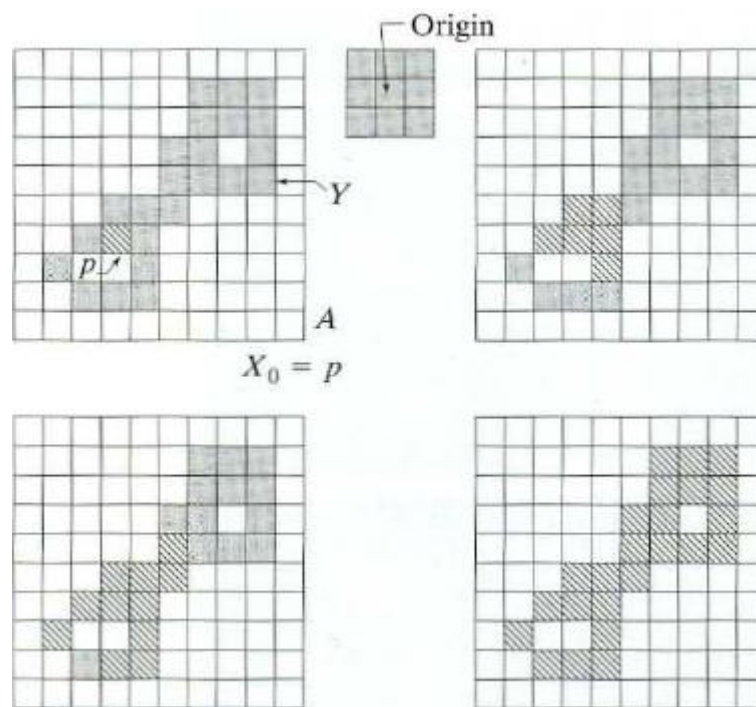


Figura. 22: Demonstração do processo de extração de componentes conectados

Fonte: *Gonzalez, 2002*

3.7.4 DEMAIS ALGORITMOS

Além dos algoritmos descritos até o momento, existem outros algoritmos nas bibliografias, mas que não serão aprofundados devido a sua aplicação não estar dentro

do escopo dos objetivos deste trabalho. A seguir alguns são apresentados.

O algoritmo do *Casco Convexo* gera a complementação da imagem para que esta seja apresentada como um retângulo. O *Afinamento*, como seu próprio nome já diz, tem o objetivo de afinar a imagem até possuir apenas um ponto e o *Espessamento* funcionando de forma dual ao Afinamento.

O algoritmo de *Esqueletonização* tem como sua finalidade representar a forma estrutural de uma região plana, reduzindo este a um grafo, com a largura de apenas um ponto.

Já a *Poda* é um complemento dos algoritmos de Afinamento e Esqueletonização, utilizado como um método de pós processamento com a finalidade de remover componentes excessivos que tenham sobrado (Marques Filho, 1999).

3.8 MORFOLOGIA MATEMÁTICA EM TONS DE CINZA

A Morfologia Matemática em tons de cinza não está no escopo deste trabalho, portanto será abordado um breve descrito sobre este assunto.

A Morfologia Matemática em tons de cinza é uma extensão da morfologia binária, porém com um grau de complexidade muito maior, tanto em conceito como em implementação. Nela, o domínio do valor de um ponto deixa de estar apenas entre 0 e 1, e passa a poder assumir qualquer valor inteiro. Assim, as operações lógicas funcionam como conversões aritméticas, ou seja, o processo de união torna-se o valor máximo e a intersecção o valor mínimo. (Wangenheim, 2000) Dessa forma, operações básicas da Morfologia Matemática Binária como a Dilatação, Erosão, Abertura e Fechamento também fazem parte das operações em tons de cinza.

Também existe a Morfologia Matemática em imagens coloridas, em que é realizado basicamente o mesmo processo que em imagens em tons de cinza, porém as operações são realizadas em cada um dos canais que compõem as cores da imagem (Wangenheim, 2000).

4 PROPLACAS

Neste capítulo é proposto um método para identificação dos caracteres específico para placas de licenças de veículos automotores brasileiros, passando desde a localização da placa no veículo em uma imagem até o reconhecimento dos caracteres da mesma. Desenvolveu-se um método que realiza várias etapas, as quais se utilizam de algoritmos e teorias revisados nos capítulos 2 e 3.

Inicialmente se apresenta os recursos utilizados, em seguida o método proposto e descreve suas devidas etapas em detalhes, como o pré-processamento, avaliação e conversões da imagem. Estas ações definem os requisitos da etapa seguinte, que é a segmentação da placa. Após a segmentação dos caracteres é feito o reconhecimento dos caracteres em relação a uma base de caracteres criada. Também é descrito como se procede ao treinamento da base de caracteres, e por fim, um breve descritivo da arquitetura do software.

4.1 RECURSOS TECNOLÓGICOS

A implementação foi desenvolvida na Linguagem de Programação C# e Framework .NET, com a versão 4.0, que é a mais recente desta tecnologia até a entrega deste trabalho. Estes recursos estão disponíveis no software Visual C# 2010 Express Edition, disponibilizada pela Microsoft através do convênio com a universidade (Academic Alliance).

Este framework possui a biblioteca System.Drawing, nativa para tratamento de imagens, e disponibiliza classes como o Bitmap, que tem a capacidade de abrir imagens matriciais, apenas passando por parâmetro o caminho do arquivo da imagem, e aceita os formatos: JPEG, GIF, BMP, PNG, EMF, Ícone, TIFF e WMF. Abaixo um exemplo de como abrir uma imagem com a classe Bitmap:

```
Bitmap Imagem = new Bitmap(@"C:\Imagens\Foto.jpg");
```

Portanto, para o funcionamento do software deste trabalho é necessário a instalação do .NET Framework 4.0.

4.2 MÉTODO PROPOSTO

No desenvolvimento foi elaborada uma seqüência lógica de etapas, e em cada etapa a utilização de algoritmos necessários. Tendo como objetivo o sucesso na localização da placa de licenciamento de um veículo em uma imagem, a segmentação correta dos caracteres e a identificação dos mesmos ou treinamento da base de caracteres.

O método proposto possui quatro etapas, que são apresentadas de forma linear. E as etapas posteriores dependem do sucesso da etapa antecessora.

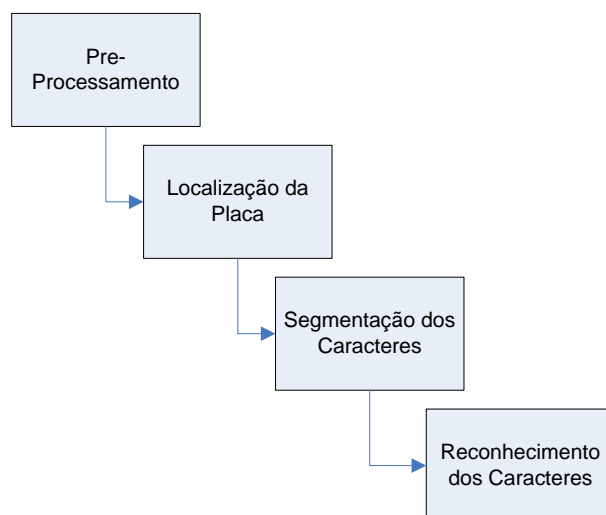


Figura 23: Fluxograma do método de reconhecimento de uma placa de veículo proposto

O Pré-Processamento tem como finalidade, verificar cores, tamanho e contraste da imagem, a fim de preparar a imagem para a próxima etapa. O objetivo da Localização da Placa é encontrar a provável parte da imagem onde está a placa do veículo e segmentá-la. A Segmentação dos Caracteres identifica na imagem os objetos que sejam as prováveis letras e números. E, por fim, o Reconhecimento dos Caracteres segmentados convertidos para caracteres ASCII.

4.3 PRÉ-PROCESSAMENTO

Antes de realizar qualquer processamento para a identificação da placa se faz necessário verificar algumas informações e também realizar algumas transformações

com objetivo de facilitar o processo de localização da placa. É realizada nesta etapa a Verificação do Tamanho da Imagem e redução da mesma, caso seja necessário, pois imagens muito grandes tornam o processamento muito demorado. Após, caso a imagem seja colorida, esta é convertida para tons de cinza, e por último é analisado o contraste linear da imagem.

4.3.1 VERIFICAÇÃO DO TAMANHO DA IMAGEM

Neste método a imagem é analisada, pois é recomendado que tenha no máximo o tamanho de 1080x810 pixels. Caso uma das dimensões seja maior, a imagem é redimensionada proporcionalmente até que uma das dimensões desta esteja com o valor máximo.

A redução da imagem utiliza um algoritmo que o .NET Framework possui, e realiza este procedimento sem maiores perdas na qualidade das mesmas.

4.3.2 CONVERSÃO DA IMAGEM PARA TONS DE CINZA

Conforme descrito no Capítulo 2, para converter uma imagem em tons de cinza é necessário realizar a média dos canais RGB (vermelho, verde e azul) em cada pixel.

Neste método se faz necessário verificar se a imagem já está em tons de cinza, onde os valores pixels nos três canais devem ser iguais. Verifica-se se todos os pixels da primeira linha da imagem. Desta forma, se pelo menos um pixel tiver com os valores dos canais diferentes, conclui-se que a imagem é colorida, caso contrário a imagem já está em tons de cinza. A importância dessa verificação é evitar a conversão da imagem, pois este é um processamento custoso.

Caso o processo anterior tenha concluído que a imagem é colorida, percorre-se toda a imagem pixel a pixel realizando a média entre os canais e convertendo-a para tons de cinza.

A seguir é demonstrado o código fonte responsável pela conversão da imagem em tons de cinza:

```
System.Drawing.Color CorAntiga = Imagem.GetPixel(x, y);  
int L = (int)((CorAntiga.R + CorAntiga.G + CorAntiga.B) / 3);  
System.Drawing.Color CorNova = System.Drawing.Color.FromArgb(L, L, L);
```

4.4 LOCALIZAÇÃO E SEGMENTAÇÃO DA PLACA

Neste processo foi utilizado o Perfil de Linha como método para orientar a localização da placa na imagem.

O Perfil de Linha da região onde a placa está localizada, necessariamente possui um comportamento padrão caracterizado por sucessivos picos de máximos e mínimos, intercalados, com distâncias e amplitudes semelhantes. Este comportamento é denominado “assinatura” (Cardoso, 2006). Pode-se observar facilmente ao analisar o Perfil de Linha.

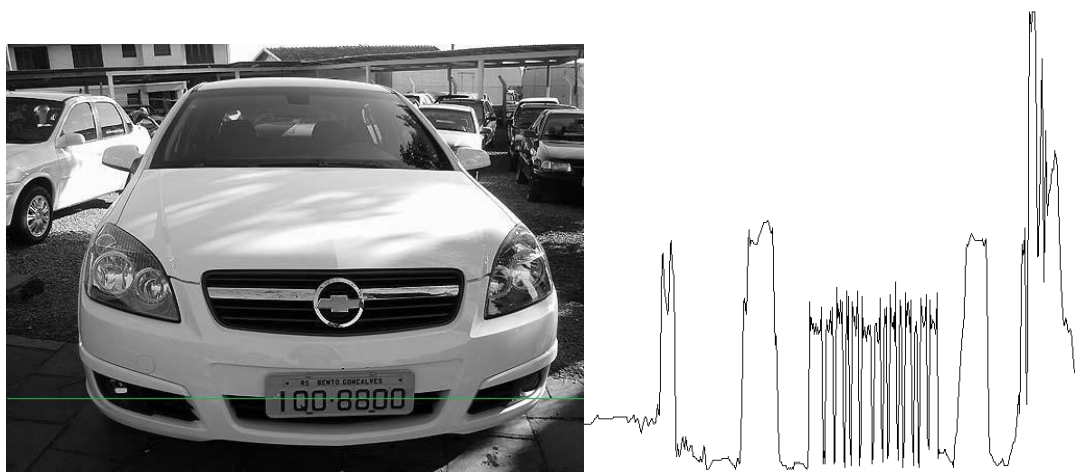


Figura 24: Perfil de uma linha que possui a placa

Este método percorre a imagem a partir da metade desta até o final buscando a placa através do perfil de linha. Porém não se faz necessário percorrer todas as linhas da imagem. Dessa forma percorre-se a imagem em um intervalo de linhas. Este intervalo é definido pelo tamanho da imagem para que seja especificado pelo tamanho padrão dos caracteres da placa nas fotos. Para uma imagem com altura de 810 pixels percorre-se a cada 10 pixels.

Para a localização da placa o método se utiliza de estatística descritiva e é representada neste trabalho por três conceitos: Média, Variância, e Desvio Padrão. Estas fórmulas são aplicadas sobre o perfil de linha.

$$\bar{x}(PL_x) = \frac{\sum_{y=0}^{n-1} PL_x[y]}{n} \quad (a)$$

$$s^2(PL_x) = \frac{\sum_{y=0}^{n-1} (PL_x[y] - \bar{x}(PL_x))^2}{n} \quad (b)$$

$$s(PL_x) = \sqrt{s^2(PL_x)} \quad (c)$$

Através destas fórmulas são calculados a média (a), variância (b) e desvio padrão (c) dos níveis de tons de cinza de cada linha percorrida na imagem, e em cada linha a partir dessas informações são traçadas no gráfico de perfil as linhas equivalentes a média, e duas linhas para os desvios padrões abaixo e acima da média.

Tendo as linhas dos desvios padrões traçadas, é possível definir os pontos de máximo e mínimo. Para que um ponto seja considerado ponto de máximo, este deverá ser maior que o desvio padrão acima da média. E o mesmo raciocínio vale para os pontos de mínimo, sendo que deverá ser o ponto mais baixo e estar abaixo do desvio padrão menor que a média. Além disso, parte-se da premissa de que os pontos de máximo e mínimo devem ser intercalados, ou seja, depois de um ponto de máximo poderá vir somente um ponto de mínimo.

Desta forma define-se que a linha da placa será a que tiver maior quantidade de pontos máximos e mínimos. E então, a partir desta linha são recortados da imagem 35 pixels para cima e para baixo da linha.

Após, o objetivo é segmentar as colunas onde começa e termina a placa do veículo. Nesta etapa é criada uma estrutura de dados indicando a localização dos pontos de máximo e de mínimo, onde é atribuído 1 para um ponto de máximo e -1 para um ponto de mínimo. Os demais pontos são atribuídos com o valor 0.

Formada a estrutura de dados, com as indicações dos pontos de máximo e mínimo, é obtida a média das distâncias entre os pontos. A partir dessa média é formada uma nova estrutura de dados onde consta 1 para os intervalos onde a distância está acima da média e 0 para as distâncias abaixo da média.

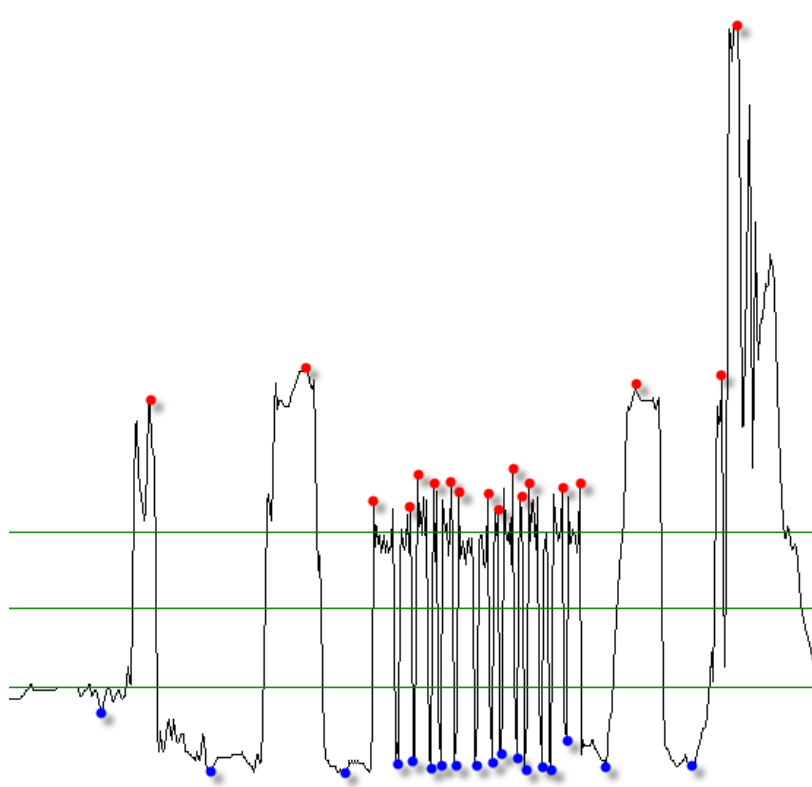


Figura 25: Perfil de linha com os pontos de máximo e mínimo destacados



Figura 26: Linha recortada a partir do perfil de linha com maior número de pontos máximos e mínimos

Com esta estrutura de dados define-se então que o intervalo da placa é a maior sequência de zeros. Assim segmenta-se a partir do início do intervalo e no fim do intervalo, adicionando mais 30 pixels para cada lado.



Figura 27: Placa localizada e segmentada

Porém, para o bom funcionamento deste método é necessário que a amostra da imagem seja colhida em um ambiente controlado, ou seja, uma amostra que contenha menor quantidade possível em riqueza de detalhes, que podem apontar falsos positivos.

As imagens ideais para este método são imagens que tenham sido fotografadas de cima para baixo. Assim ter-se-á o chão como fundo, na maioria dos casos, o que torna a imagem homogênea (Cardoso, 2006).

4.5 SEGMENTAÇÃO DOS CARACTERES

Após a obtenção da segmentação da placa do veículo na imagem, pelo processo visto anteriormente, a próxima etapa é conseguir segmentar os caracteres da placa, para que posteriormente possa ser comparado com a base de caracteres e ser realizada a identificação dos mesmos.

Nesta etapa são realizados, sobre o segmento da placa, os processos de verificação do contraste linear, limiarização e a segmentação dos componentes conexos.

A assertividade desta etapa é muito importante, pois a próxima depende da segmentação correta dos caracteres.

4.5.1 VERIFICAÇÃO DO CONTRASTE LINEAR

Inicialmente, é necessário verificar o contraste da segmentação da placa, ou seja, analisar a luminosidade deste segmento da imagem, com a finalidade de melhorar a sua nitidez.

Neste processo verifica-se a média dos tons de cinza da linha central neste segmento. Caso esta média esteja inferior a 128, ou seja, com pouca luminosidade, é aplicado o contraste linear alterando o histograma entre os tons de cinza 0 e 175, sendo esticado até o limite 255, sobre este segmento. Em primeiro momento a média ainda pode estar inferior a 128, por isso, este processo é repetido até que a média esteja com o valor satisfatório acima de 128.

A seguir o código responsável pelo procedimento avaliação do contraste linear:

```
int Media = 0, Limite = 128;
do
{
    int[] PerfilLinha = TCC.Tratamento.Fachada.PerfilLinha(Imagem,
    Imagem.Height / 2);
    Media = PerfilLinha.Media();
    if (Media < Limite)
        Imagem = TCC.Tratamento.Fachada.ContrasteLinear(Imagem, 0, 175);
} while (Media < Limite);
```

4.5.2 LIMIAZIZAÇÃO

Como descrito no Capítulo 2, a limiarização também é chamada de binarização. Nesta etapa, o segmento da placa é convertido para uma imagem binária.

Para isto, é necessário antes obter-se o valor do Limiar Ótimo através do histograma. Com essa informação, classifica se os valores de tons de cinza menor em preto e os maiores em branco.

A seguir o código fonte responsável pela limiarização:

```
System.Drawing.Color CorAntiga = Imagem.GetPixel(x, y);
int L = CorAntiga.R;
if (L >= Threshold)
    L = 255;
else
    L = 0;
System.Drawing.Color CorNova = System.Drawing.Color.FromArgb(L, L, L);
```

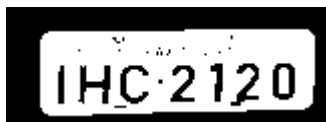


Figura 28: Placa binarizada

4.5.3 SEGMENTAÇÃO DOS CARACTERES

Este processo tem por finalidade separar os componentes conectados que são considerados como as prováveis letras e números.

A partir da imagem binária obtida dos processos anteriores, primeiramente verifica-se a existência de algum pixel com valor 1 nas bordas. Caso exista, preenche-se todos os pontos e seus vizinhos com o valor 0. Esse procedimento é tomado porque nessa etapa pressupõe-se que nenhum caractere estará contido nas bordas da imagem, e em muitos casos o pára-choque ou a cor do veículo encontra-se escura, dessa forma este seria binarizado com valor 1.

Após a eliminação do pára-choque ou a cor de veículo, ocorre o processo de segmentação dos componentes conectados. Utiliza-se o algoritmo de Componentes Conexos específico para esse problema, descrito no capítulo 3. Este procedimento gera

uma lista de imagens binárias, contendo cada elemento dessa lista um componente conectado da imagem original.

Após processamentos anteriores pressupõe-se que restou apenas as letras e alguns ruídos na imagem binária, dessa forma os componentes conexos com maior número de elementos podem ser considerados as letras e os números das placas, dessa forma seleciona-se os sete maiores componentes conexos, e os ordena por ordem crescente da posição horizontal.

I-H-C-2-1-2-0

Figura 29: Caracteres da placa identificados e segmentados

4.6 RECONHECIMENTO DOS CARACTERES

Esta é a etapa final do método proposto, que por sua vez deve converter da forma mais correta possível as imagens segmentadas da etapa anterior em caracteres da tabela ASCII.

Determinou-se que neste processo deve existir uma base de caracteres, os quais serão utilizados para a comparação com as imagens segmentadas, e o mesmo inicialmente deverá ser treinado.

4.6.1 BASE DE CARACTERES

A base de caracteres trata-se de um repositório de pequenas imagens binárias as quais armazenam matrizes binárias com o tamanho de 15x10, e a disposição dos elementos da matriz com valor 1 deve apresentar o formato de uma letra. A base de caracteres é salva em disco em arquivos serializados.



Figura 30: Representação gráfica de uma matriz na base de caracteres, onde em preto é igual a 1 e em branco a 0

Cada matriz estará vinculada a um caractere da tabela ASCII, denominado Identificador. No caso da matriz representada pela Figura 30, está vinculado ao caractere “C”. E cada identificador poderá ter quantas matrizes forem necessárias.

A quantidade de identificadores também é livre. Dessa forma temos um relacionamento de 1 para N entre os Identificadores e as matrizes.

De acordo com a legislação brasileira, as placas de licenciamento de veículos em território nacional possuem três letras e quatro números. Em especial carros, ônibus e caminhões, esses caracteres estão sempre dispostos na mesma linha e sempre apresentando inicialmente as letras e em seguida os números (Souza, 1997).

Dessa forma foi estabelecido que deve existir um arquivo de base para servir de repositório apenas para as letras e outro arquivo para os números. Pois sabendo que os três primeiros caracteres são letras não haveria necessidade de confrontar os três primeiros caracteres com toda a base, dessa forma o algoritmo de reconhecimento torna-se otimizado e mais preciso. Assim, sempre são confrontadas as três primeiras imagens segmentadas com a base de letras e as quatro imagens seguintes com a base de números.

4.6.2 RECONHECIMENTO DE CARACTERES

Antes de confrontar cada imagem segmentada, o software verifica se o tamanho destas imagens são maiores que 10X15. Caso uma destas dimensões seja maior, realiza-se a redução da imagem até que as dimensões estejam de tamanho proporcional às matrizes da base de caracteres.

Com as imagens segmentadas realiza-se um confronto com a base de caracteres, onde para cada imagem segmentada o processo percorre toda a base de caracteres (letras ou números conforme posição da imagem segmentada), matriz a matriz realizando uma contagem de igualdades entre cada elemento da matriz e da imagem. Dessa forma o

caractere da tabela ASCII a ser considerado será o Identificador da matriz que tiver maior quantidade de elementos iguais.

Esta etapa é o fim do método proposto. A Figura 31 demonstra a imagem original e o resultado obtido ao clicar em “Reconhecer”.

O software também apresenta um percentual de certeza quanto a cada caractere, e uma média geral entre os caracteres. Este percentual é calculado dividindo o valor da contagem das igualdades pelo total de elementos na matriz, que neste caso é 150.

A Figura 32 mostra o percentual de certeza de cada caractere referente à placa da Figura 31.



Figura 31: Resultado final do processamento do método proposto



Figura 32: Percentual de Certeza

4.6.3 TREINAMENTO DA BASE DE CARACTERES

O software necessita ter um grau satisfatório de assertividade, e para isso ocorrer é preciso que a base de caracteres seja alimentada. Na demonstração da Figura 33 ocorreu erro na identificação dos caracteres por “ignorância” da base de caracteres.



Figura 33: Imagem que não teve acerto na identificação dos caracteres

Nessa circunstância deve-se “ensinar” o software para que possa ter maior “conhecimento” sobre caracteres.

Ao clicar em “Treinamento” é aberta uma tela conforme Figura 34. Esta tela

apresenta as imagens segmentadas identificadas como caracteres da placa. O usuário deve manualmente informar no campo abaixo de cada imagem qual é a letra ou número correspondente a imagem. Lembrando que as três primeiras imagens são letras e as quatro seguintes são números. Ao clicar em adicionar, estas imagens são redimensionadas para 10x15, e em seguida adicionadas a base de caracteres com o seu respectivo identificador.

E após o treinamento, quando realizar novamente o reconhecimento da mesma imagem, o software irá retornar o resultado correto, conforme Figura 35.

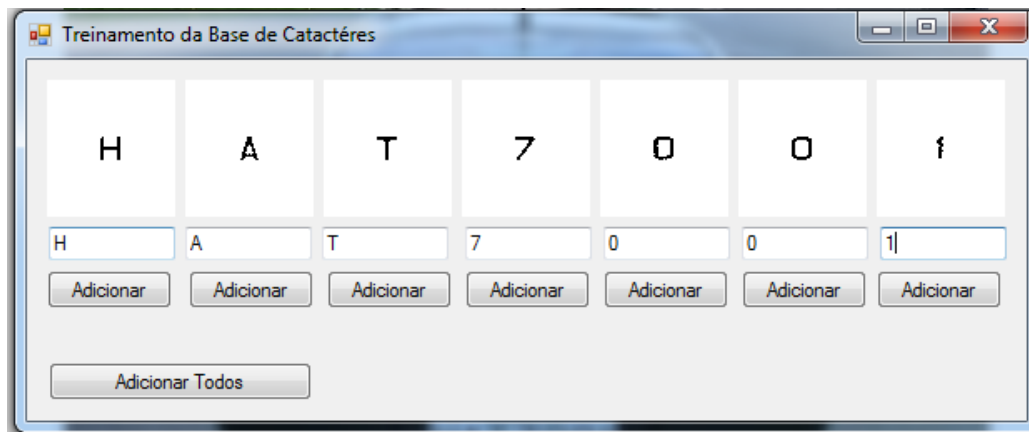


Figura 34: Treinamento da base de caracteres



Figura 35: Imagem reconhecida corretamente após treinamento do software

4.7 ARQUITETURA DO SOFTWARE

O software foi desenvolvido em três projetos de Visual Studio, os quais foram divididos conforme sua aplicabilidade e funcionalidades de implementação, sendo: TCC.dll, Proplacas.dll as bibliotecas e Proplacas.exe o programa final desktop.

Essas bibliotecas foram separadas em diferentes projetos para que houvesse maior facilidade de reaproveitamento das funcionalidades desenvolvidas em projetos futuros. Ou até mesmo para desenvolver software voltado a plataforma web, ou móvel, utilizando-se das bibliotecas já desenvolvidas para a solução desktop.

4.7.1 TCC.DLL

Esta é a biblioteca base do software, contendo as funcionalidades mais básicas de processamento de imagens utilizadas neste trabalho, e as funcionalidades de MM. Nesta biblioteca encontram-se as funcionalidades como: Contraste Linear, Limiarização, Conversão para Tons de Cinza, Perfil de Linha e Histograma.

4.7.2 PROPLACAS.DLL

Esta é uma biblioteca que reúne as funcionalidades de processamento de imagens aplicado aos métodos de Localização da Placa, Segmentação e Reconhecimento dos Caracteres. Essas funcionalidades utilizam-se da biblioteca TCC.dll descrita no item anterior.

4.7.3 PROPLACAS.EXE

Este é o programa executável desktop, que se utiliza da biblioteca Proplacas.dll, e possui a interface gráfica utilizando as funcionalidades da biblioteca.

5 TESTES E RESULTADOS

Este capítulo apresenta os resultados dos testes obtidos a partir de uma amostra de imagens contendo veículos com suas respectivas placas, as estatísticas dos resultados e também quanto à segmentação dos caracteres, e descreve algumas situações de sucesso e falha. Descreve como as imagens testadas foram obtidas e suas características, e os resultados em relação ao reconhecimento dos caracteres da placa.

5.1 AQUISIÇÃO DAS IMAGENS

As imagens utilizadas para os testes foram adquiridas em um site de venda de veículos on-line disponível no endereço www.carrosnaserra.com.br. Utilizaram-se fotos de veículos nacionais de várias categorias e cores. As imagens desta amostra, em média, possuem tamanho de 555x416 pixels.

Tais imagens foram obtidas em um ambiente não controlado, ou seja, possuem uma grande riqueza de detalhes de cenário. Podendo, dessa maneira, prejudicar a localização da placa dos veículos. Além disso, a distância da câmera até o veículo também não está padronizada.

Foram coletadas para estes testes 134 imagens de fotos de veículos e os resultados completos por imagem se encontram na tabela do Anexo A.

5.2 LOCALIZAÇÃO DA PLACA

Os testes para verificação quanto à localização da placa do veículo foram divididos em duas partes, as quais são etapas do método:

- A localização correta da linha onde se encontra a placa;
- Na linha localizada, a segmentação correta do início e do fim da placa;

Com uma amostra de 134 imagens foram obtidos os seguintes resultados:

Tabela 2: Resultados Estatísticos da Localização e Segmentação da Placa

	Localização da Linha da Placa	Segmentação da Placa
Satisfatório	101 (75%)	65 (65%)
Não Satisfatório	33 (25%)	36 (35%)
Total	134	101

Devido ao fato de que a segunda etapa depende do sucesso da primeira, os resultados desta foram realizados apenas sobre os valores satisfatórios do primeiro. Mas, no quadro geral, o percentual satisfatório de segmentação da placa é de 48%.

No exemplo da amostra apresentada na Figura 35, o software teve total sucesso em identificar a linha da placa e segmentá-la. Pode-se perceber, claramente na imagem do perfil de linha, a assinatura do local da placa e também que nessa região da assinatura os pontos de máximo e mínimo cruzam as duas linhas de desvio padrão sobre a média.

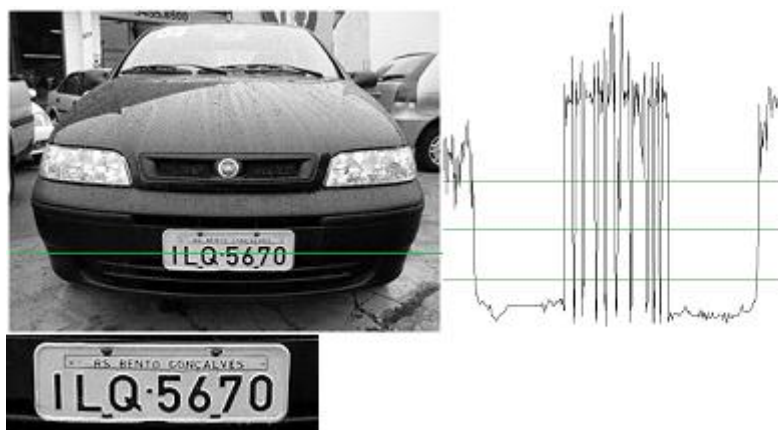


Figura 35: Sucesso ao segmentar a placa

Na amostra da Figura 36, o software não conseguiu identificar corretamente a linha da placa. Isto devido ao fato da assinatura do local da placa não estar cruzando as duas linhas demarcatórias de desvio padrão, nota-se que os pontos de mínimo possuem valores superiores ao da linha de desvio padrão que está abaixo da média.

No exemplo da Figura 37 é apresentada a ocorrência mais comum entre as falhas de segmentação. O software conseguiu identificar corretamente a linha da placa, mas ao segmentar o local da mesma, o software não obteve sucesso. Nota-se no perfil de linha da placa desta amostra que poucos supostos pontos de máximo estão cruzando a linha do desvio padrão acima da média. Desta forma o software confundiu os pontos de início

e fim da placa, e segmentou em torno do limite entre o veículo e o chão.



Figura 36: falha na localização da linha da placa

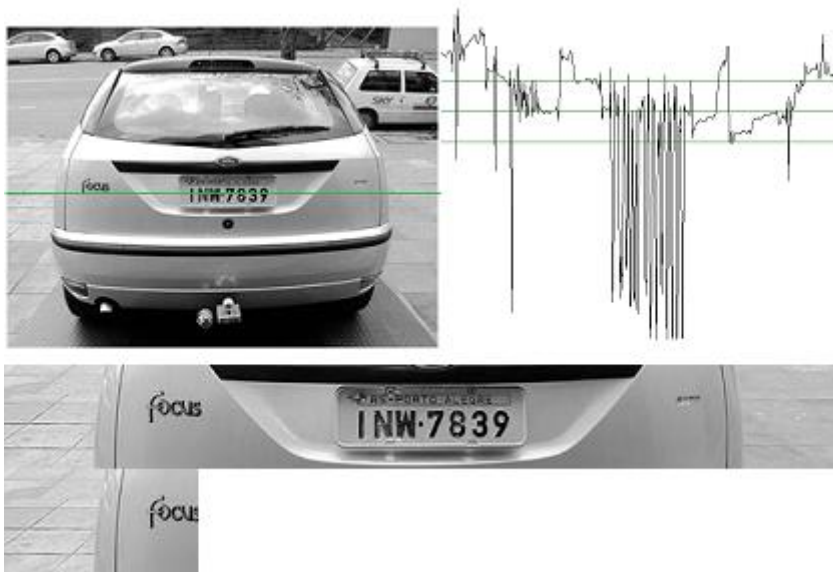


Figura 37: sucesso ao encontrar a linha e falha ao segmentar a placa

Ocorreram também situações onde o software segmentou parcialmente o local da placa, caso pouco comum entre as amostras testadas. O motivo pelo qual este fato ocorreu está relacionado à estrutura de dados binária referente às médias das distâncias. Esta é gerada a partir da análise do perfil de linha, conforme descrito no Capítulo 4, em um dos valores correspondente ao local da placa, onde a distância entre um ponto máximo e mínimo ficou um pouco acima da média entre as distâncias. Dessa forma, o software interpretou esse ponto como sendo o fim da placa.

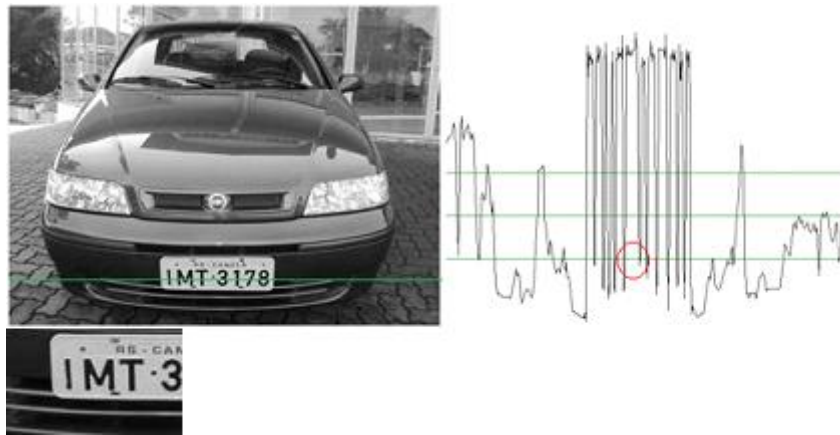


Figura 38: Demonstração de uma placa segmentada de forma parcial

Outro caso de falha é um falso positivo causado por ruídos inerentes de ambiente não controlado. A Figura 39 demonstra a tentativa de identificação da linha da placa. O perfil de linha mostra uma grande quantidade de pontos de máximo e mínimo, acima e abaixo das linhas que indica os desvios padrões, causados pelo ruído gerado pelo cascalho no solo.

Isto não significa que todas as imagens cujo chão seja cascalho irão gerar ruídos. Este foi um caso ocorrido.

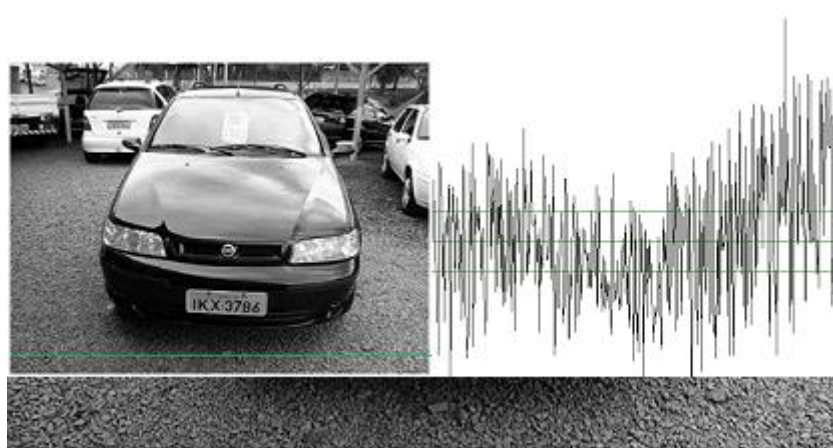


Figura 39: Falha ao localizar a linha devido ao excesso de ruído na imagem

5.3 SEGMENTAÇÃO E IDENTIFICAÇÃO DOS CARACTERES DAS PLACAS

Como descrito no Capítulo 4, o processo de segmentação dos caracteres depende do sucesso da localização da placa. Devido ao grau de acerto da etapa anterior estar em

48% e a necessidade de se utilizar todas as amostras, fez-se necessário o recorte manual do local da placa das amostras cuja localização não procedeu de forma satisfatória.

O processo de identificação dos caracteres utiliza uma base de caracteres, na qual existe a necessidade de treinamento. Portanto, estabeleceu-se uma estratégia para este teste. Do total, 70 fotos foram selecionadas para o treinamento da base de caracteres, e as demais 64 fotos para a obtenção e apresentação do relatório estatístico.

Os testes dessa etapa foram divididos em duas partes, as quais são as últimas etapas do método:

- Segmentação correta dos caracteres;
- Identificação dos caracteres contra uma base de caracteres;

Conforme descrito no capítulo anterior, o sucesso na identificação dos caracteres depende altamente do sucesso da segmentação dos mesmos. Desta forma, as estatísticas apresentadas quanto ao segundo grupo serão realizadas apenas sobre os resultados bem sucedidos da segmentação e sobre as imagens recortadas manualmente.

5.3.1 SEGMENTAÇÃO

Serão apresentados nesta sessão os exemplos de segmentação da placa, o qual as amostras irão apresentar a imagem do veículo, logo abaixo a placa segmentada, na mesma linha à direita a segmentação com aplicação de contraste linear, adiante na mesma linha, a binarização da região da mesma e por fim os caracteres segmentados.

Os resultados obtidos quando à segmentação dos caracteres em uma amostra de 134 imagens são:

Tabela 3: Resultados Estatísticos da Segmentação dos Caracteres

	Segmentação correta
Satisfatório	67 (50%)
Não Satisfatório	67 (50%)
Total	134

No exemplo abordado pela Figura 40, é apresentado um caso bem sucedido de segmentação dos caracteres da placa. Este sucesso é inerente de uma boa localização da câmera em relação à placa do veículo. Nota-se que os caracteres da placa foram

segmentados de forma bem definida.

Em outro exemplo de segmentação bem sucedida, demonstrado pela Figura 41, apresenta o sucesso quanto ao fator luminosidade. A placa do veículo encontra-se em uma região pouco iluminada. Houve diversas situações como esta, os quais foram contornados através do contraste linear.

Já na Figura 42, o contraste está pouco elevado somente entre as letras e o fundo da placa, em relação ao resto da imagem. Percebe-se que a cor do veículo é branca, e esta elevou a média. Por esse motivo, o contraste linear não obteve sucesso, em elevar o contraste dos caracteres, causando uma grande ocorrência de ruído ao binarizar.

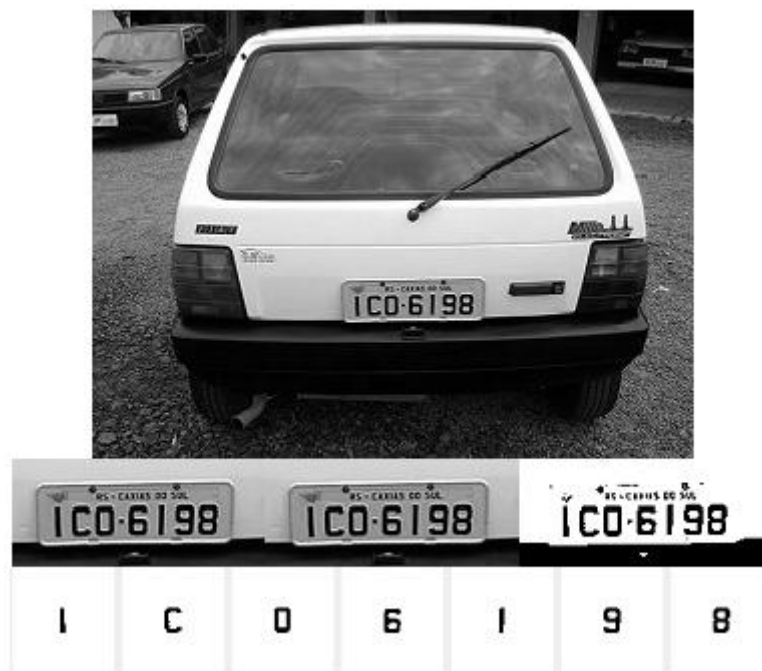


Figura 40: Sucesso ao segmentar os caracteres

A situação da Figura 43 demonstra uma falha de segmentação devido ao detalhe do furo do parafuso estar muito próxima a dois números, por este detalhe, estes dois caracteres permaneceram unidos ao binarizar.

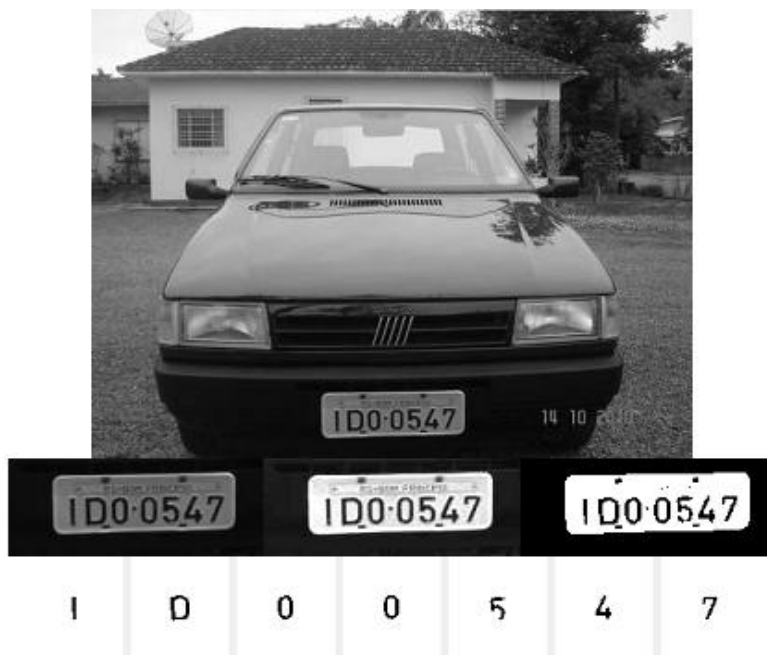


Figura 41: Sucesso ao segmentar os caracteres em imagem escura



Figura 42: Falha ao segmentar os caracteres devido ao pouco contraste entre a placa e o fundo



Figura 43: Falha ao segmentar os caracteres devido ao detalhe do parafuso

Existem casos extremos, para os quais não haveria uma solução trivial, como a má qualidade da placa do veículo, caracteres apagados, ou borrões na placa. O sucesso deste método depende muito do bom estado de conservação destas. O exemplo apresentado na Figura 44 mostra uma placa com alguns de seus caracteres parcialmente apagados, o que causou a falha na segmentação.



Figura 44: Falha ao segmentar os caracteres devido a má qualidade da placa

5.3.2 RECONHECIMENTO

Neste teste, das 134 imagens, 70 foram utilizadas para o treinamento da base de caracteres. Estas 70 amostras não foram consideradas neste teste. Entre as 64 amostras restantes, 34 tiveram a segmentação bem sucedida. Portanto, foram utilizadas 34 amostras neste teste.

Os dados estatísticos deste teste foram classificados em percentual de certeza e número de caracteres corretos. A tabela 4 demonstra a média destas classificações em relação à amostra utilizada.

Tabela 4: Resultados Estatísticos do Reconhecimento dos caracteres

Média do Percentual de Certeza	89%
Média de caracteres corretos por placa	6,09 (em uma escala de 0 a 7)

A seguir é apresentada a placa segmentada, em seguida os caracteres segmentados, depois o resultado do reconhecimento e, por fim, o resultado gerado pelo software que apresenta o percentual de certeza por caractere e geral. A Figura 45 exhibe um caso bem sucedido quanto ao reconhecimento de todos os caracteres da placa.



Figura 45: Sucesso ao reconhecer os caracteres

A Figura 46, apesar de apresentar um percentual de certeza maior que ao exemplo anterior, não obteve sucesso total no reconhecimento.



Figura 46: Falha ao reconhecer os caracteres com grau de certeza elevado

Na Figura 47 observa-se que o primeiro e o terceiro caractere não tiveram o reconhecimento correto. Observa-se que justamente nestes dois, o percentual de certeza encontra-se muito abaixo da média.



Figura 47: Falha ao reconhecer os caracteres

A Figura 48 apresenta a amostra com o percentual de certeza mais baixo ocorrido nos testes. Neste caso, apenas dois caracteres foram reconhecidos corretamente. Observa-se que, mesmo os caracteres corretos possuem um percentual de certeza baixo.

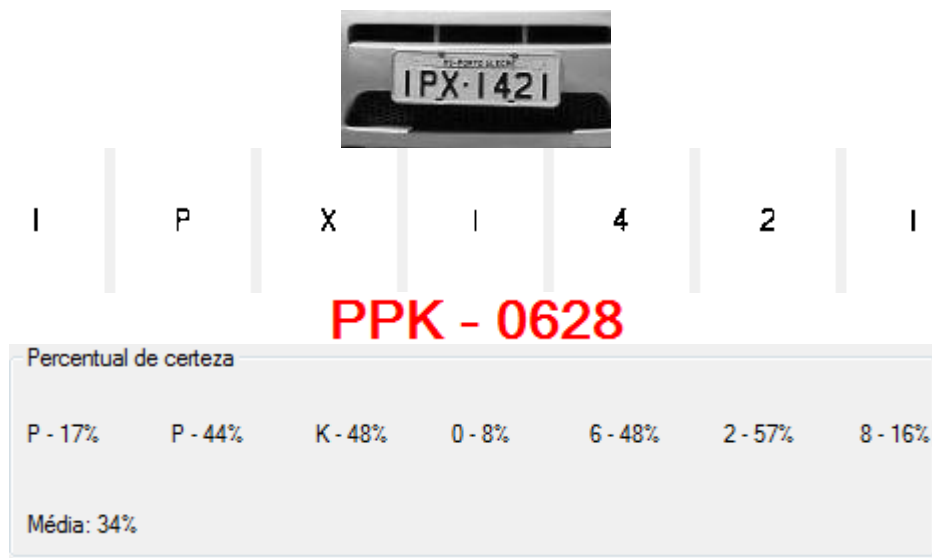


Figura 48: Percentual de certeza mais baixo entre as amostras

E por fim, na Figura 49, a amostra com o maior percentual de certeza obtido nos testes. Com um percentual de certeza elevado em todos os caracteres.

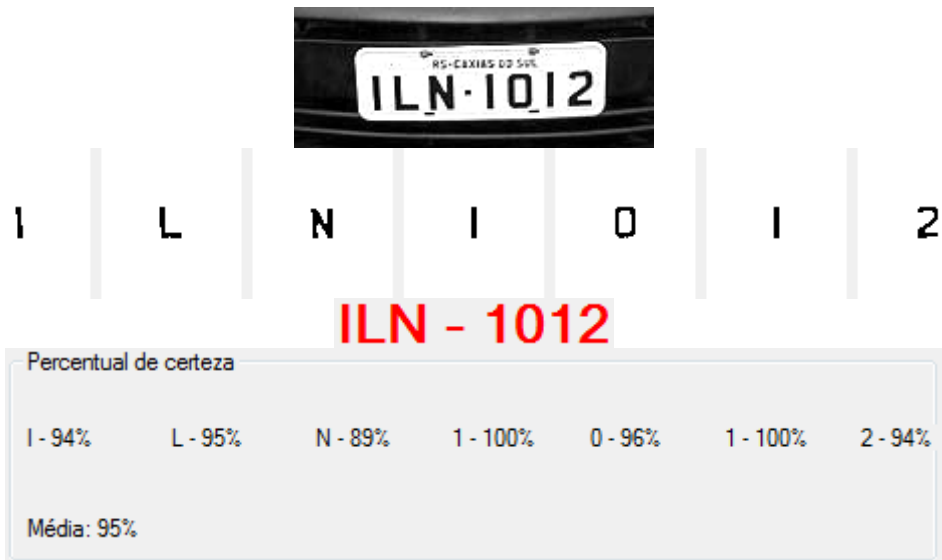


Figura 49: Percentual de certeza mais elevado entre as amostras

6 CONCLUSÕES GERAIS

A abordagem apresentada neste estudo empregou análise de perfil de linha para localização da placa do veículo para segmentação automática e o reconhecimento dos caracteres da mesma, a partir de uma base de caracteres dispostos em matrizes binárias armazenados em arquivos serializados, o qual necessita de um treinamento.

No desenvolvimento deste trabalho foi necessário um aprendizado aprofundado de alguns conceitos e teorias quanto à área de processamento de imagens e uma pesquisa de certos métodos para localização de placas de veículos em imagens abordados em outros trabalhos.

Os resultados foram gerados a partir de uma base de 134 amostras de imagens de veículos obtidas de um ambiente não controlado. Ou seja, imagens cujo cenário não é padronizado e a distância da câmera até o veículo também é variável. E, em alguns casos, até mesmo o ângulo não está perpendicular à placa e a luminosidade.

Conclui-se que este método exige, para obtenção de resultados satisfatórios, que as imagens dos veículos sejam obtidas em um ambiente controlado, a partir do contexto no qual o veículo se encontra. E a distância da câmera à placa igualmente precisa de um padrão, pois quanto mais distante a câmera, menor a qualidade da imagem no local da mesma.

Quanto ao método utilizado para o reconhecimento dos caracteres, a conclusão é que o método pode ser bastante eficiente a partir do momento em que a base de caracteres possua uma quantidade maior de amostras entre letras e números. Pois, com o número reduzido de amostras separadas para o treinamento, já se obteve um resultado satisfatório com uma média de acerto de 87% dos caracteres de cada placa nos testes.

A partir do princípio da complexidade do método proposto em função do tempo disponível para as pesquisas e o desenvolvimento deste, junto com o perfil das amostras utilizadas, conclui-se que o objetivo deste trabalho foi alcançado de forma bastante satisfatória. Afinal, o estudo feito para a realização deste trabalho foi bastante esclarecedor quanto ao caminho para se iniciar e continuar um software com a finalidade de reconhecimento de padrões em imagens digitais.

Propõem-se para continuação deste trabalho, alguns desafios relacionados ao aperfeiçoamento em todas as etapas do método. Entre estes, a realização de uma pesquisa mais aprofundada no método de segmentação das placas, a fim de melhorar o

seu grau de assertividade. Também a realização de pesquisas na etapa de segmentação dos caracteres e um treinamento da base de caracteres, com uma maior quantidade de amostras, bem como o desenvolvimento para o reconhecimento de placas de motocicletas que possuem uma configuração diferente quanto à disposição dos caracteres. Um desafio interessante para o aperfeiçoamento do método é o desenvolvimento da identificação de placas que estejam inclinadas.

Uma solução comercial com este método pode ser aplicada na identificação automática de veículos em estacionamentos para controle de acesso, controle de entrada e saída de veículos em empresas com fluxo elevado de veículos, o qual automatizaria uma atividade humana passível de erros e proporcionaria agilidade no tráfego.

7 REFERÊNCIAS

BITTENCOURT, Anderson Luis Schvindt. **Segmentação em Imagens Digitais de Veículos Obtidas por Controladores Eletrônicos de Velocidade**. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Santa Catarina. Florianópolis, 2003.

CARDOSO, Edgard Davidson Costa. **Localização de Placas de Veículos Baseado em Métodos Estatísticos**. Dissertação (Graduação em Sistemas de Informação) - Universidade Católica de Minas Gerais. Belo Horizonte, 2006.

CARVALHO, John Edson Ribeiro de. **Uma abordagem de segmentação de placas de automóveis baseada em morfologia matemática**. Dissertação (Mestrado em Computação) - Universidade Federal Fluminense. Niterói. 2006.

FALCÃO, Alexandre Xavier. **Introdução ao Processamento de Imagem Digital**, 2003. Disponível em <<http://www.dcc.unicamp.br/~cpg/material-didatico/mo443/200302/>>. Acesso em 15 ago. 2010.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. 2.ed. New Jersey: Prentice Hall, 2002.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de imagens digitais**. São Paulo: E. Blücher, 2000.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital image processing**. Massachusetts: Addison, 1992.

MANFIO, Felipe Haddad. **Reconhecimento de Caracteres Alfanuméricos Para Identificação de Placas de Veículos**. Dissertação (Graduação em Engenharia da Computação), Universidade do Norte do Paraná, Londrina. 2004.

MASCARENHAS, Nelson D.A.; VELASCO, Flávio R.D. **Processamento digital de imagens**. 2.ed. Buenos Aires, AG: Kapelusz, 1989.

MARQUES FILHO, Ogê; VIEIRA FILHO, Hugo. **Processamento digital de imagens**. Rio de Janeiro: Brasport, 1999.

MICROSOFT. **Bitmap Class**. Disponível em <<http://msdn.microsoft.com/en-us/library/system.drawing.bitmap.aspx>>. Acesso em 08 nov. 2010.

OSAWA, Daniel Minoru. **Processamento e Reconhecimento de Imagens Digitais da Retina Humana**. Dissertação (Graduação em Ciência da Computação) - Universidade Federal Fluminense. Niterói, 2006.

SOUZA, Fernando Peixoto Coelho de. **Localização e leitura automática de caracteres alfanuméricos – uma aplicação na identificação de veículos**. Dissertação (Pós-Graduação em Engenharia Elétrica) - Universidade Federal Do Rio Grande Do Sul, Porto Alegre, 2000.

SOUZA, Rui Barbosa de. **Código de trânsito brasileiro: Lei n.9.053, 23 de setembro de 1997**. 2.ed. rev. e ampl. Porto Alegre: Rígel, 1997.

WANGENHEIM, A. V. **Introdução a Visão Computacional**. Universidade Federal do Rio Grande do Sul, 2000

WILLRICH, R.; **Sistemas Multimídia Distribuídos**. Florianópolis, 2001.

Carros na Serra. Disponível em <<http://www.carrosnaserra.com.br>>. Acesso em 10 nov. 2010.

ANEXO A

Tabela 5: Resultados Completos das Amostras

Imagem	Encontra Linha da Placa	Segmenta	Segmentação Caracteres	Identificação	Caracteres Corretos
ADL6399	1	0	0	Treinamento	
AGP9603	1	1	0	Treinamento	
BLT4974	1	1			
BNU2878	0		1	Treinamento	
CAQ4000	1	1	1		
CBF9674	1	1	0	Treinamento	
CIX5954	0		1	Treinamento	
CLS5679	0		0	Treinamento	
CON8238	0				
DBU8803	1	0			
DCV1091	0				
DFU9976	1	1	0	Treinamento	
DKF0408	0		1	Treinamento	
DKG5757	1	1	1	Treinamento	
EBE9749	1	0	1	Treinamento	
EBE9749-2	1	0	0		
EDV4851	1	1	1	Treinamento	
ENZ0556	0		0	Treinamento	
GKU8114	1	1	1	Treinamento	
HAT7001	1	1	0	Treinamento	
HDM5643	1	0	0	Treinamento	
HGG0812	1	0	0	Treinamento	
HPB3804	1	1	1	Treinamento	
COM8775	1	1	0	Treinamento	
ICO6198	1	1	1	Treinamento	
ICQ8546	1	1	0	Treinamento	
IDO0547	1	1	1	Treinamento	
IDR2384	0		1	Treinamento	
IEA5942	1	0	0	Treinamento	
IEF2450	1	1			
IFB2460	1	1	0	Treinamento	
IFE8532	1	1	1	Treinamento	
IFW4401	1	1	0	Treinamento	
IGC2428	1	1	1	Treinamento	
IGC2967	0		0	Treinamento	
IGI2371	1	0	1	Treinamento	

IGR7171	1	1	1	Treinamento	
IGV5672	1	0	1	Treinamento	
IHA2611	1	1	1	Treinamento	
IHH8223	1	0	0	Treinamento	
IHP1418	1	0	1	Treinamento	
IHP5290	1	1	0	Treinamento	
IIT2562	1	1	1	Treinamento	
IIV9868	1	1	1	Treinamento	
IIZ0837	1	1	1	Treinamento	
IJA0449	0		0	Treinamento	
IJG1783	1	0	1	Treinamento	
IJG6812	0		1	Treinamento	
IJH6781	1	1			
IJK2596	1	0	1	Treinamento	
IJL1734	1	0			
IJL4098	1	0			
IJM3023	1	1	1	Treinamento	
IJQ6861	1	1	1	Treinamento	
IKC6487	0				
IKD7364	1	0	1	Treinamento	
IKE0113	1	1	0	Treinamento	
IKP7763	1	0	1	Treinamento	
IKP7982	1	0	1	Treinamento	
IKW5946	1	1	0	Treinamento	
IKX3786	0		0	Treinamento	
IKY8429	1	0	0	Treinamento	
ILA6588	1	0	0	Treinamento	
ILD1220	0		1	Treinamento	
ILE3259	0		0	Treinamento	
ILI3409	1	1	1	69%	5
ILJ4098	1	0	1	90%	5
ILQ2977	0		0		
ILQ5670	1	1	1	90%	6
ILQ5670-2	1	1			
ILQ7990	1	1	0		
ILR3166	1	1	1	92%	7
ILR3166	1	1			
IMC1518	1	1	1	90%	5
IMM3938	0		0		
IMP2009	0		0		
IMR3827	1	0	0		
IMT3178	1	0	1	88%	6
ING6258	0		0		
INJ1341	1	1	1	94%	6
INL0079	1	1	0		

INL1012	1		1	95%	7
INM4330	1	0	0		
INM9527	1	1	0		
INT3752	1	1	1	88%	6
INT3752-2	1	0	0		
INW7839	1	0	0		
INY9184	1	1	1	84%	7
INY9184-2	1	1	1	88%	7
INZ2781	1	1	0		
IOA5695	1	1	0		
IOA8107	1	1	0		
IOE0944	1	1	1	82%	6
IOF3565	0		1	93%	7
IOJ7176	1	0	0		
IOM2871	0		1	93%	7
IOP2026	1	1	0		
IOP6620	1	1	0		
IOX1654	0		1	92%	5
IOX1925	1	1	1	92%	6
IOX6262	1	1	1	91%	7
IPD5151	0		1	94%	6
IPD5151-2	1	0	1	94%	7
IPE3502	0		0		
IPF4644	1	0	1	91%	6
IPF7159	1	1	1	91%	5
IPK9337	0		1	92%	7
IPO2399	1	0	1	92%	7
DBY6285	1	1	1	Treinamento	
IPX1421	1	1	1	34%	2
IQG5447	1	1	1	83%	7
IQO8800	0	0	1	88%	5
IQP4864	1	1	1	39%	2
IRA1994	1	1	0		
IRC7439	1	1	1	88%	7
IRC7439	1	1			
IRD4072	1	1	1	94%	7
IRM0506	1	1	1	78%	6
IRX0222	1	1	0		
IRZ1980	0		1	90%	7
IVG7117	1	1	1	94%	7
IXN3333	0		0		
JAA0100	1	0	0		
JAC6868	0		1	88%	6
JAI6090	1	0			
JAX6666	1	1	0		

KEE1210	1	0	0		
KFO8586	0		1	83%	5
LOK9315	0	1	0		
MBA4093	0		0		
MCR1402	1	0	0		
MDF1429	0				
MFO7520	1	0	0		
MFO7520-2	1	0	1	89%	3
	101	65	67	89%	6,09