

DETECÇÃO DE IMAGENS DIGITALIZADAS DE PLACAS DE AUTOMÓVEIS
POR MEIO DE FILTRAGEM CASADA

Bruno Borba Leite

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO
DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM
CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

Prof. José Manoel de Seixas, D.Sc.

Prof. Gelson Vieira Mendonça, Ph.D.

Prof. Marcelo Portes de Albuquerque, Dr. Ing.

Prof. Augusto Santiago Cerqueira, D.Sc.

RIO DE JANEIRO, RJ - BRASIL
OUTUBRO DE 2005

LEITE, BRUNO BORBA

Detecção de Imagens Digitalizadas de Placas
de Automóveis por Meio de Filtragem Casada
[Rio de Janeiro] 2005

XI, 104 p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia Elétrica, 2005)

Dissertação - Universidade Federal do Rio
de Janeiro, COPPE

1. Processamento de sinais e imagens
2. Filtros casados

I. COPPE/UFRJ II. Título (série)

Agradecimentos

Gostaria de expressar meus agradecimentos ao meu orientador José Manoel de Seixas da COPPE pela confiança, discussões e pelas críticas que foram fundamentais para minha aprendizagem, além de toda insistência e luta para que esse trabalho fosse concluído. O apoio recebido em todos os momentos foi fundamental e sem ele não teria conseguido chegar até aqui.

A todo pessoal de suporte do PEE e um em especial à Solange, a quem tanto trabalho dei e que me ajudou de forma imprescindível.

Agradeço aos amigos Clayton de Medeiros Vasconcelos, Aline da Rocha Gesualdi, Marcelo Portes de Albuquerque, Márcio Portes de Albuquerque e Eugenio S. Caner que diretamente participaram desse trabalho e pelo apoio em todas as etapas de desenvolvimento. Quero agradecer ao Engenheiro Carlos Tadeu Santos Alves pelo apoio e amizade.

Um agradecimento especial à minha mãe e à Larissa que nunca deixaram me abater diante das dificuldades neste árduo caminho e pela presença incondicional durante meu período de internação.

Aos amigos do INdT André Araújo, André Mota, Jaro, Leandro, Mateus, Roberto, Tsuyoshi, Wesley, Ozeneide que nesse último ano contribuíram para meu crescimento profissional e pessoal, principalmente ao Túlio que tem sido como um irmão para mim e que esteve sempre ao meu lado desde a minha chegada e durante toda a fase da minha doença.

Aos que estiveram comigo durante a fase de internação hospitalar: Léo, Marcus, José Luiz, Carvalhal, Mário, Flávia, Cecília, Bruno, Renata e Sérgio, Renata e Márcio, Yuri, Roney, Daniel, Genésio e Fernando e a toda família Moreira.

Ao meu irmão, meus avós, tios, primos e amigos que sempre me apoiaram. Enfim, a todos aqueles que de alguma forma contribuíram para a minha formação e a realização deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

DETECÇÃO DE IMAGENS DIGITALIZADAS DE PLACAS DE AUTOMÓVEIS
POR MEIO DE FILTRAGEM CASADA

Bruno Borba Leite

Outubro/2005

Orientador: José Manoel de Seixas

Programa: Engenharia Elétrica

Técnicas de visão computacional possuem, cada vez mais, um papel importante no desenvolvimento de sistemas de automação. Este trabalho aborda a implementação da etapa de localização automática da imagem da placa de um veículo particular brasileiro a partir da imagem digital do veículo, preparando a etapa posterior de identificação dos caracteres da placa. Este estudo utiliza a análise espectral de linhas da imagem e filtragem casada. Os resultados obtidos apresentam uma eficiência de identificação acima de 81% para imagens adquiridas em uma situação prática e em boas condições.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

DIGITAL VEHICLE PLATE IMAGE DETECTION BASED ON MATCHING
FILTERS

Bruno Borba Leite

October/2005

Advisor: José Manoel de Seixas

Department: Electrical Engineering

Machine vision techniques play an increasing role in the development of automating systems. This work aims at locating automatically Brazilian private vehicle plates from their digital images. It is the first step for identifying the license plate in a fully automatic system. This study employs line spectral analysis and matching filters. The results achieved show efficiency above 81% for images acquired in a practical situation and in good conditions.

Índice

1 Introdução	1
1.1 Motivação	1
1.2 O que foi feito e a metodologia adotada	3
1.3 O documento	5
2 Apresentação do Sistema de Localização	7
2.1 O estado da arte no mundo	7
2.2 Especificações do CONTRAN	15
2.3 Especificação do sistema de localização	17
2.4 Aquisição das imagens	18
2.5 Criação da base de dados	20
3 Pré-Processamento	23
3.1 Análise freqüencial	23
3.2 Janela deslizante	32
3.3 O algoritmo	34
3.4 Parâmetros do pré-processamento	37
3.5 Resultados	39
4 Filtro Casado	42
4.1 Teoria de filtros casados	42
4.2 Projeto dos filtros casados	44
4.2.1 Análise estatística	45
4.2.1.1 Placa	45
4.2.1.2 Traço	47
4.2.1.3 Parafusos	53
4.2.1.4 Caracteres	55
4.2.2 Filtros projetados	56
4.2.2.1 Placa	58
4.2.2.2 Traço	59
4.2.2.3 Caracteres	61
5 Aplicação do Filtro Casado	62
5.1 Normalização de energia	62

5.2 Algoritmo de varredura do filtro casado	66
6 Resultados	70
6.1 Medição da eficiência	71
6.1.1 Filtro baseado na placa inteira	71
6.1.2 Filtro baseado no traço	74
6.1.3 Filtro baseado nos caracteres	76
6.2 Eficiência	78
6.2.1 Filtro baseado no traço	78
6.2.1.1 Filtro 1	79
6.2.1.2 Filtro 2	79
6.2.2 Filtro baseado na placa	80
6.2.2.1 Filtro 1	81
6.2.2.2 Filtro 2	82
6.2.2.3 Filtro 3	83
6.2.3 Filtro baseado nos caracteres	84
6.2.4 Escolha do melhor filtro	85
6.3 Limiares de decisão	91
6.3.1 Filtro baseado na placa	91
6.3.1 Filtro baseado nos caracteres	93
7 Conclusão	96
7.1 O banco de dados	96
7.2 Pré-processamento	97
7.3 Filtro casado	97
7.4 Pós-processamento	98
7.5 Implementações	98
7.6 Trabalhos futuros	99
8 Referências Bibliográficas	101

Lista de Figuras

2.1 Exemplo de um sistema de reconhecimento de automóveis por imagem	9
2.2 Modelo de placa de veículos brasileiros, incluindo as dimensões de altura, largura e espaçamento dos caracteres, segundo as normas definidas pelo CONTRAN.	16
2.3 Arquitetura do sistema de localização de placas de veículos.	18
3.1 Imagens com diferentes frequências espaciais	25
3.2 Exemplo de levantamento de parâmetros de frequência	26
3.3 Histograma de validação da faixa de frequência horizontal sobre as placas do conjunto de desenvolvimento.	27
3.4 Imagem original I e a imagem F resultante, formada pelo módulo da resposta em frequencial de cada linha.	28
3.5 Exemplo de uma imagem de transformadas já filtradas (imagem B).	31
3.6 Exemplo do somatório feito pela equação (3.11).	33
3.7 Exemplo do somatório feito pela equação (3.12), utilizando o método de preenchimento com zeros.	34
3.8 Diagrama com os passos do algoritmo, juntamente com seu conjunto de entradas.	36
3.9 Histograma de altura dos caracteres das placas da base de treinamento.	38
3.10 Ilustração dos valores de tamanho da janela deslizante (T), distância entre as respostas ($distResp$) e tamanho do recorte ($tamRecorte$).	39
3.11 Exemplo de região errada localizada pelo pré-processamento.	40
4.1 Gráfico de distribuição de largura e altura das placas da base de desenvolvimento.	46
4.2 Limites propostos para o padrão de filtro de traço.	47
4.3 Limites alternativos propostos para o padrão de filtro de traço.	48
4.4 Limites propostos para o padrão de um terceiro de filtro de traço.	48
4.5 Gráfico de distribuição de largura e altura dos traços.	49

4.6 Gráfico de distribuição de largura e altura dos traços do filtro 1.	50
4.7 Gráfico de distribuição de largura e altura dos traços do filtro 2.	51
4.8 Gráfico de distribuição de largura e altura dos traços do filtro 1.	52
4.9 Ilustração das tiras que contêm os parafusos.	53
4.10 Distribuição de distâncias entre os parafusos superiores e inferiores.	54
4.11 Filtro 1, tamanho de 248 x 43 <i>pixels</i> .	59
4.12 Filtro 2, tamanho de 252 x 43 <i>pixels</i> .	59
4.13 Filtro 3, tamanho de 228 x 43 <i>pixels</i> .	59
4.14 Filtro 1.	60
4.15 Filtro 2.	60
4.16 Filtro baseado nos caracteres.	61
5.1 Exemplo de filtro de tamanho NxM varrendo uma imagem de tamanho PxQ com um passo maior que 1, onde $P > N$ e $Q = M$.	67
5.2 Exemplo de filtro de tamanho NxM varrendo uma imagem de tamanho PxQ com passos maior que 1 <i>pixel</i> , onde $P > N$ e $Q > M$.	69
6.1 Os pontos azuis na figura (da esquerda para a direita): (CXRS, CYRS), (CSX ₁ , CSY ₁), (CEX ₇ , CEY ₇) e (CXRE, CYRE).	73
6.2 CXResp é o ponto médio entre XRS e XRE	76
6.3 CYResp é o ponto médio entre YRS e YRE	76
6.4 A figura representa um exemplo de acerto na localização de caracteres.	78
6.5 Distribuição dos acertos para o filtro 2.1	87
6.6 Distância dos pontos ideais do filtro 2.1	88
6.7 Distribuição dos acertos para o filtro 2.2	89
6.8 Distância dos pontos ideais do filtro 2.2	90
6.9 Resposta da correlação entre o filtro e áreas da imagem que contêm a placa.	92
6.10 Resposta da correlação entre o filtro e áreas da imagem que não contêm a placa (ruído).	93

Lista de Tabelas

2.1 Largura dos caracteres em milímetros segundo especificações do CONTRAN. A tolerância é de 10%.	16
2.2 Campos registrados para cada imagem da base de dados.	21
3.1 Eficiências de localização do pré-processamento.	39
4.1 Tamanhos (largura x altura) definidos para os filtros baseados em cada um dos caracteres (em pixel).	55
6.1 Eficiências de localização do Filtro 1.	79
6.2 Eficiências de localização do Filtro 2.	79
6.3 Eficiências de localização do Filtro 1.	81
6.4 Eficiências de localização do Filtro 2.	82
6.5 Eficiências de localização do Filtro 3.	83
6.6 Eficiências de localização dos filtros de caracteres. Alguns caracteres não ocorreram durante os testes, por isso não há eficiências associadas a eles.	85
6.7 Filtros de maior eficiência de localização.	86
6.8 Valores médios e desvios padrões da resposta a acertos e erros dos filtros de caracteres.	94

Capítulo 1

Introdução

O constante crescimento do fluxo de veículos nas cidades torna o monitoramento dos mesmos uma necessidade de crescente importância. Este monitoramento tem como principais finalidades: controle de fluxo, fiscalização e identificação dos veículos [1].

Atualmente, já existem no mercado alguns sistemas que se propõem a realizar este monitoramento através de imagens digitalizadas. Contudo, as eficiências alcançadas, ainda insatisfatórias, fazem com que tais produtos não estejam tão difundidos.

1.1 Motivação

A identificação de veículos, até os dias de hoje, tem sido feita, predominantemente, através de passes eletrônicos¹. As técnicas de visão computacional, processamento de sinais e imagens e reconhecimento de padrões, ainda pouco utilizadas, podem ser vistas como outra abordagem para a solução deste problema. Em casos onde o uso do passe eletrônico se torna inviável, a solução poderia estar na identificação automática dos veículos através destas técnicas. Este enfoque baseado em processamento de sinais, apesar de ser mais sensível às más condições atmosféricas, como nevoeiro, chuva intensa e variação de luminosidade durante o período de um dia, vem

¹ Dispositivo eletrônico fixado no automóvel com finalidade de emitir, através de rádio frequência, o código do usuário.

encontrando crescente aceitação por parte dos técnicos e autoridades ligados ao estudo e controle de tráfego, além de órgãos de segurança pública.

Uma ampla gama de aplicações pode ser identificada para este tipo de sistema, como: detectar infratores, controlar pedágios e estacionamentos, efetuar estudo de origem/destino, medir e planejar o fluxo do trânsito, encontrar carros roubados, etc [2].

Está em andamento um projeto, no âmbito de uma colaboração entre o CBPF (Centro Brasileiro de Pesquisas Físicas), através do LPDSI (Laboratório de Processamento Digital de Sinais e Imagens), e a UFRJ (Universidade Federal do Rio de Janeiro), através do LPS (Laboratório de Processamento de Sinais), para a identificação de veículos automotores particulares por meio do processamento digital de imagem e técnicas de reconhecimento de padrões.

O início deste projeto focou no reconhecimento de caracteres em placas de veículos particulares brasileiros (placas de fundo cinza claro e caracteres pretos) [2]. Partindo da imagem da placa, os caracteres foram localizados e classificados por meio de técnicas de processamento digital de imagens e redes neurais artificiais.

As eficiências experimentais obtidas nesse trabalho, mostradas abaixo, motivaram a implementação das outras etapas requeridas numa aplicação prática.

- Algarismos: 98%
- Letras: 96%

- Placas: 85%

Seguido a esses resultados, um estudo que consistia em localizar um eixo horizontal que cortasse o centro da placa foi apresentado, atingindo eficiências de até 96% [1].

Ambos os estudos supracitados foram implementados fazendo-se uso de bases de imagens consideradas ideais, uma vez que eram compostas por imagens editadas (com o intuito de corrigir problemas relacionados a angulação e iluminação) de carros estacionados (parados).

1.2 O que foi feito e a metodologia adotada

A proposta deste trabalho é apresentar uma técnica de localização de placas de veículos automotores particulares brasileiros em imagens digitalizadas adquiridas em condições reais, sem qualquer tipo de correção ou condicionamento pós-aquisição das imagens.

O projeto em discussão foi dividido em três partes:

1. Aquisição das imagens;
2. Criação da base de dados;
3. Localização da placa na imagem e o respectivo recorte.
 - (i) Pré-processamento;
 - (ii) Localização.

A aquisição das imagens foi feita numa praça de pedágio do Rio de Janeiro, e se tentou fazer com que a resolução das placas nas imagens respeitasse pelo menos o valor proposto em [2], de 200 x 65 *pixels*. Com as imagens adquiridas, passou-se à construção da base de dados, que foi feita de forma manual. Tendo em mãos as principais informações espaciais e estatísticas sobre as imagens que compunham a base de dados, alguns filtros casados foram propostos e implementados. Para o projeto do filtro adotou-se uma simplificação do problema, considerando-se uma aproximação determinística para os padrões com os quais os filtros se encontram casados. Esses filtros, aliados a técnicas freqüenciais de pré-processamento e de janelas deslizantes, acabaram por compor o sistema de localização de placas.

Desta forma, o sistema de localização se resume a um pré-processamento de filtragem freqüencial horizontal (1D) e varreduras de janelas deslizantes. Da imagem pré-processada são extraídas regiões de maior probabilidade de ocorrência da placa segundo análise freqüencial. São sobre essas regiões extraídas da imagem original que os filtros casados, propostos neste trabalho, atuarão. A saída esperada do filtro é a região da placa. Nessas saídas, serão feitos os estudos de eficiência do sistema.

Todas as ferramentas utilizadas neste trabalho foram desenvolvidas em Matlab. Essas ferramentas têm como finalidade automatizar a análise dos dados, a construção dos filtros e as medidas de performance das técnicas propostas.

1.3 O documento

O segundo capítulo, que trata da apresentação do sistema, será iniciado mostrando o estado da arte dos sistemas de identificação de placa de veículos automotores no mundo. Em seguida, abordaremos a fase de especificação do sistema. Por último, será abordada a criação da base de dados, bem como seus respectivos critérios, levando em consideração a aquisição das imagens. Esta base de dados servirá como referência na construção dos filtros casados e nas análises de resultados do estudo. Serão mostradas as bases de desenvolvimento e de teste, bem como o critério de separação desses conjuntos.

O terceiro capítulo fala das técnicas de pré-processamento frequencial utilizadas pelo sistema proposto neste estudo. O quarto capítulo se iniciará com uma abordagem teórica sobre filtros casados, que servirá como apoio para a discussão dos filtros projetados. Em seguida, passaremos à análise estatística dos dados e padrões utilizados para a construção dos filtros e, por último, será mostrada a construção dos mesmos.

No quinto capítulo, trataremos do algoritmo de aplicação dos filtros casados nas imagens que compõem o conjunto de teste. No sexto capítulo, será especificado e proposto o método de medição de eficiência dos resultados. Por último, os resultados são mostrados e ocorre a escolha do melhor filtro, juntamente com conjunto de parâmetros de entrada do pré-processamento, com base nas eficiências alcançadas.

O sétimo, e último capítulo, diz respeito à conclusão deste trabalho, analisando os erros e acertos e apontando melhorias para uma próxima etapa de desenvolvimento do sistema de identificação de veículos automotores particulares brasileiros.

Capítulo 2

Apresentação do Sistema de Localização

Neste capítulo, faremos a especificação do sistema de localização, baseado em estudos anteriores e nas resoluções do CONTRAN. Em seguida, será discutida a aquisição das imagens e a construção da base de dados. Todavia, iniciaremos nossa discussão apresentando o estado da arte no mundo, isto é, o desenvolvimento de projetos similares a este por empresas e centros de pesquisa e desenvolvimento, no Brasil e no mundo.

2.1 O Estado da Arte no Mundo

Basicamente, existem dois tipos de sistema de reconhecimento de veículos: sistemas de RF² e sistemas de detecção por imagens. Os sistemas de RF são compostos por RF ID's, colocados no pára-brisas dianteiro dos veículos, e detectores que, geralmente, são colocados em praças de pedágio. Ao passar pelos detectores, o RF ID é identificado e, no caso de pedágios, libera a passagem do automóvel pela via expressa. Todavia, esse tipo de sistema apresenta uma falha de segurança, dado que os RF ID's podem ser retirados dos veículos a que estão associados e colocados em outros que não estejam registrados na base de dados. Logo, não é possível associar o RF ID às placas dos veículos. Além disso, mesmo desconsiderando a possibilidade de se trocar os

² Rádio frequência

identificadores entre os veículos, seria necessário que todo veículo possuísse um dispositivo deste instalado e por consequência que tenha sido previamente cadastrado.

Já os sistemas de imagem não necessitam da instalação de dispositivos nos veículos e tão pouco fazer qualquer tipo de associação de identificadores com o automóvel. Ele simplesmente reconhece o veículo pela placa de licença, obrigatório em todos os veículos que circulam pelas vias públicas brasileiras.

A configuração mínima, normalmente utilizada em sistemas de reconhecimento de placas de veículos automotores por imagem, é composta por um sensor para detectar a passagem do veículo (*trigger* do sistema), que pode ser indutivo ou elétrico, uma câmera de vídeo, analógica ou digital, cuja função é fazer a captura da imagem (no momento em que o *trigger* aciona o sistema), uma placa digitalizadora de imagens, responsável pela aquisição da imagem e, por último, um computador (processador) que funciona como analisador de tráfego, já que fica responsável pela localização da placa e reconhecimento dos caracteres nas imagens adquiridas. Tal arquitetura é ilustrada na figura 2.1. Eventualmente, o sistema pode ser móvel.

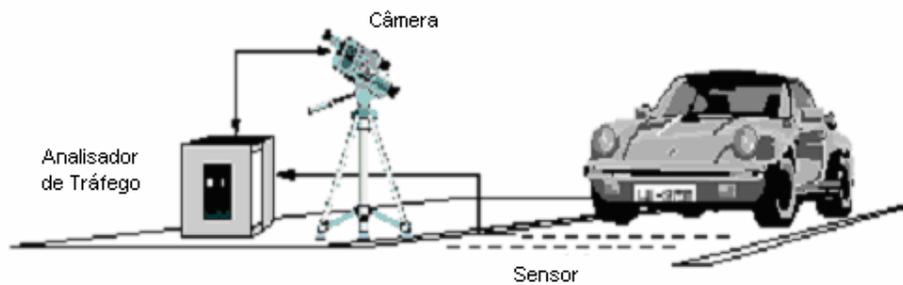


Figura 2.1: Exemplo de um sistema de reconhecimento de automóveis por imagem.

Atualmente, diversas empresas e centros de pesquisa e desenvolvimento nacionais e internacionais trabalham e oferecem sistemas de reconhecimento automático de placa de veículos automotores. Contudo, a maior parte das características técnicas disponibilizadas não é feita de forma padronizada. Deste modo, é inviável montar uma análise comparativa entre eles.

São diversas as técnicas de localização de placas e reconhecimento de padrões utilizadas por estes tipos de sistemas. Vejamos algumas técnicas existentes no Brasil e no mundo aplicadas neste campo:

- Na Turquia, dois pesquisadores da *International Computer Institute Ege University*, desenvolveram um sistema de segmentação de placas de veículos baseado em quatro etapas [3]:
 - i. Pré-processamento;
 - ii. Localizador de regiões aproximadas;
 - iii. Extração da placa;

iv. Segmentação dos caracteres.

O pré-processamento consiste em converter a imagem adquirida em RGB e em modo *field*³ para uma imagem em níveis de cinza. Em seguida, a imagem é redimensionada (640 x 468). Para finalizar o pré-processamento, a imagem sofre uma equalização de histograma⁴.

O localizador de regiões aproximadas usa um algoritmo de detecção de bordas verticais (filtro *Sobel*) [4]. Uma janela deslizante varre a imagem processada, tentando localizar regiões de *pixels* brancos com altas intensidades. A região de maior resposta à janela é guardada. Em seguida, há uma compressão da escala de cinza a fim de facilitar o trabalho da binarização, que ocorre logo em seguida. Na saída desta etapa, são fornecidas as coordenadas da imagem candidata.

Quando a imagem candidata é extraída da imagem total, ela sofre uma rotação, para corrigir possíveis inclinações, e depois uma binarização. Em seguida, é deslocada em 3 *pixels* para a direita e depois subtraída da candidata original. A imagem resultante mostra as sombras dos caracteres na placa. Para eliminar o ruído de *salt and pepper*, um filtro mediano passa-alta [4], e em seguida uma dilatação morfológica, são aplicados sobre ela.

³ Imagem com resolução de 640 x 312 *pixels*. Geralmente utilizada para adquirir imagens de objetos em movimento.

⁴ Operação que consiste em melhorar o contraste de uma imagem, transformando seus valores em uma imagem de intensidade, para que a imagem de saída case com um histograma específico [4].

A última etapa trata de extrair os caracteres a partir de uma projeção vertical e localização dos máximos e mínimos.

Essa técnica, utilizando uma base de dados reais, alcançou eficiências de localização de até 73%.

- Um outro trabalho, desenvolvido no Brasil, apresenta uma outra abordagem para segmentar placas de veículos de uma imagem estática, baseado em métodos granulométricos [5]. Este método usa técnicas de morfologia matemática, combinando um algoritmo rápido de granulometria e uma análise de resíduos morfológicos e é dividido nas seguintes etapas:
 - i. Pré-processamento
 - Conversão da imagem para níveis de cinza;
 - Filtragem *Top-hat* para ressaltar os vales da imagem [4];
 - Diminuição do nível de cinza;
 - ii. Peneiramento por granulometria
 - Obtenção dos marcadores dos dígitos da placa do veículo;
 - iii. Marcação do conteúdo da placa;
 - iv. Localização dos componentes
 - Cálculo do histograma de distribuição de níveis de cinza;
 - Determinação do nível de cinza correspondente à região de fundo da imagem;

- Busca por grupos de *pixels* cujo nível de cinza seja de maior intensidade que a região de fundo;
- Anotação da posição dos *pixels* mais à esquerda, à direita, acima e abaixo do grupo de *pixels* encontrados;
- v. Redução dos números de componentes
 - Eliminação das componentes fora do tamanho padrão;
- vi. Localização da placa do veículo.

Esta técnica, em imagens obtidas a partir de veículos estacionados, chegou a eficiências de 75%. Seu ponto positivo é que ela é capaz de localizar mais de uma placa por imagem.

- Um outro sistema desenvolvido no Brasil apresenta uma nova metodologia para localização de placas de veículos. O SIAV [6] captura as imagens e procura por variações padronizadas de intensidade (busca por linhas horizontais que possuam um padrão tonal conhecido) para restringir a área que será processada na etapa de binarização local adaptativa de reconhecimento dos caracteres e aumentar, assim, a velocidade de processamento e análise. Um algoritmo de procura por dígitos é empregado e um sistema de decisão, em função do número de caracteres encontrados, é usado para adaptar, ou não, o limiar da etapa de binarização. Em uma base de dados de 500 imagens, com tamanho de 320 x 240 *pixels*, obtidas em condições reais, o sistema apresenta uma eficiência de 89,20%.

- Em [7], é proposta a localização da placa a partir da correlação entre as linhas que compõem uma imagem binarizada e uma onda quadrada com largura igual a duas vezes a largura média dos caracteres. A janela 1D, onde é feita a correlação, tem o tamanho da placa. A base de dados é formada por 40 imagens de carros estacionados e a eficiência de localização de um ponto dentro da placa é de 97%.

- Em outro artigo, é proposta a localização da placa usando um algoritmo de segmentação parcial [8]. Para tal, o método usa a seguinte seqüência de filtros morfológicos e binarização:
 - i. *Bottom-hat* (com o elemento estruturante maior que os caracteres);
 - ii. Binarização (utilizando o método de Otsu [9]);
 - iii. *Close* binário (com elemento estruturante linear)
 - O resultado será um retângulo branco no local da placa e alguns ruídos no restante da imagem;
 - iv. Pós-processamento
 - Filtra os ruídos restantes através de informação de tamanho;

Segundo o artigo, no caso de placas que possuam fundo escuro, deve-se utilizar o algoritmo de *White top-hat* [4] que corresponde a:

$$I_F = I - \text{Open}(I) \quad (2.1)$$

onde: I = imagem processada;

O algoritmo apresenta eficiência de 96%.

- Em Israel, a empresa *Htsol* desenvolveu o *Seecar*, que é um sistema de identificação de placas de licenças de vários países. Este sistema pode ser utilizado tanto em pedágios, estacionamentos, como também em auto-estradas e possui eficiência de reconhecimento elevada para todas as placas de licença dos países para o qual ele foi projetado. Este sistema utiliza uma iluminação infravermelha e uma câmera para aquisição da imagem do veículo. Um software de processamento de imagens, desenvolvido pela mesma empresa, analisa a imagem extraindo a informação da placa [10].

De modo geral, as técnicas aqui apresentadas utilizam imagens em níveis de cinza. Outro ponto importante é que nem todos os trabalhos apresentados utilizam imagens adquiridas em condições reais em seus testes. Isto pode acabar por mascarar algumas eficiências, dado que em condições reais, o algoritmo deve ser mais robusto, para suportar as variações de iluminação durante o período de um dia e até mesmo as variações climáticas.

Outro dado inexistente em todos os textos lidos é a resolução da imagem da placa. Em [2], foi proposta uma resolução e é a partir deste dado que basearemos o nosso projeto.

2.2 Especificações do CONTRAN

Antes de discutirmos a aquisição das imagens e construção da base de dados, é interessante ter-se um conhecimento das normas e padrões utilizados nas placas de veículos automotores brasileiros. Por isso, abaixo, são descritas algumas dessas especificações estabelecidos pelo CONTRAN.

O capítulo IX, Seção III, artigo 96 do Código de Trânsito Brasileiro (Lei Nº 9.503 de 23/09/97) trata da identificação dos veículos brasileiros. O artigo 115 diz: “O veículo será identificado externamente por meio de placas dianteira e traseira, sendo estas lacradas em sua estrutura, obedecidas as especificações e modelos estabelecidos pelo CONTRAN (Conselho Nacional de Trânsito)”.

O artigo 1º da resolução 045/98 do CONTRAN estabelece o Sistema de Placas de Identificação de Veículos. Este artigo estabelece que tanto a placa traseira, quanto a dianteira devem conter caracteres alfanuméricos individualizados, divididos em dois grupos. O primeiro deve ser composto por três caracteres, resultantes do arranjo com repetição de vinte e seis letras, tomadas três a três. Já o segundo, por quatro caracteres, resultantes do arranjo com repetição de dez algarismos, tomados quatro a quatro, conforme ilustra a figura 2.2. Além dos caracteres previstos neste artigo, ambas as placas devem conter, gravados em tarjetas removíveis a elas afixadas, a sigla da Unidade da Federação e o nome do Município de registro do veículo, exceção feita às placas de veículos oficiais.



Figura 2.2: Modelo de placa de veículos brasileiros, incluindo as dimensões de altura, largura e espaçamento dos caracteres, segundo as normas definidas pelo CONTRAN.

O artigo 2º desta mesma resolução especifica as dimensões, cores e demais características das placas (veja a figura 2.2). As tolerâncias nas dimensões das placas e caracteres alfanuméricos são fixadas em até 10%. Com isso, as dimensões da placa são fixadas em: (i) altura $hp = 130 \pm 13$ mm; (ii) comprimento $cp = 400 \pm 40$ m; (iii) altura dos caracteres $h = 63$ mm e (iv) largura l dos caracteres, a qual se encontra na tabela 2.1.

Tabela 2.1: Largura dos caracteres em milímetros, segundo especificações do CONTRAN. A tolerância é de 10%.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
54	44	44	43	40	40	45	45	10	36	49	40	54	47	45	44	51	46	46
T	U	V	W	X	Y	Z			1	2	3	4	5	6	7	8	9	0
44	45	49	49	49	47	40			18	36	37	40	36	36	36	38	36	36

Agora, conhecendo as especificações necessárias para o projeto, bem como os valores dos tamanhos padrões, segundo o CONTRAN, já é possível fazer uma estimativa das alturas dos caracteres.

2.3 Especificação do Sistema de Localização

Na maioria das vezes, um sistema de identificação de placas é dividido em três etapas distintas, realizadas em seqüência:

- i. Aquisição da imagem do veículo;
- ii. Localização da placa;
- iii. Recorte dos caracteres de licença;
- iv. Reconhecimento individual dos caracteres e identificação completa da placa.

Baseado nesta divisão clássica de etapas, este trabalho tratará apenas das duas primeiras, a etapa de aquisição das imagens, feita em condições reais, e a etapa de localização da placa nessas imagens, uma vez que, em [2], as últimas duas etapas já foram abordadas.

Neste projeto, como trabalharemos com imagens *offline*, é necessário fazer a aquisição das imagens e o respectivo cadastramento das mesmas para que seja criada a base de dados. A criação desta base de dados será tratada mais profundamente no próximo capítulo.

O sistema de localização foi desenvolvido segundo a seguinte arquitetura:

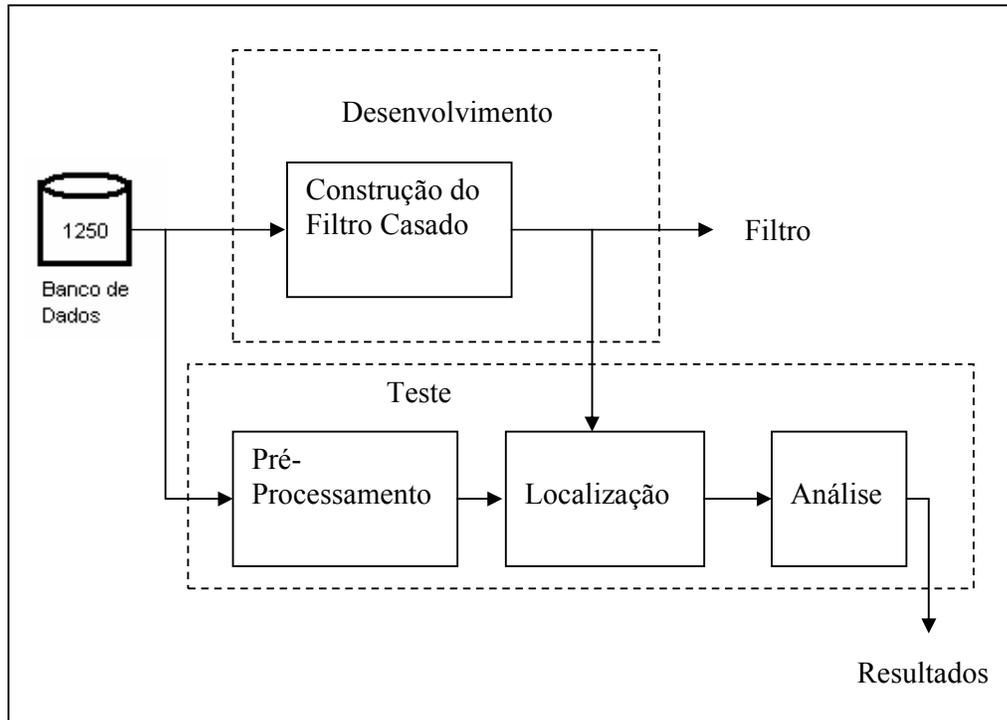


Figura 2.3: Arquitetura do sistema de localização de placas de veículos.

A base de desenvolvimento será composta por 90% das imagens (1125), enquanto que o restante, 10% (125 imagens), serão utilizadas como base de teste.

A seguir, com base na arquitetura apresentada, abordaremos a criação da base de dados.

2.4 Aquisição das imagens

A aquisição das imagens foi feita na faixa expressa da praça de pedágio da Ponte Rio-Niterói durante os dias 27 e 28 de Novembro de 2002. Foi necessário colocar uma câmera de vídeo analógica, conectada a uma placa de aquisição de imagens, instalada

em um computador pessoal. O sinal de “*trigger*” para aquisição foi gerado por um sensor indutivo já existente na via.

A aquisição se deu de forma ininterrupta durante este período, gerando mais de 25.000 imagens. Nem todas as imagens atendem ao interesse deste trabalho, já que alguns dos problemas abaixo foram encontrados:

- Imagens que não contêm carro – o sensor foi acionado por funcionários da concessionária;
- Imagens de carros sem placas – carros que trafegam irregularmente;
- Imagens com placas cortadas – as placas não estão completamente contidas no Campo de Visão;
- Imagens fora de foco – alguns carros, em alta velocidade, fazem com que o sistema não seja suficientemente rápido para adquirir a imagem no ponto correto;
- Imagens de placas de ônibus, placas oficiais, outros – existem imagens de placas diversas, nem todas são de veículos particulares (caracteres pretos e fundo cinza), que é o objeto de estudo deste trabalho.

Esses problemas, juntamente com a necessidade de levantar informações de posicionamento das placas nas imagens, bem como a dos caracteres, motivaram a construção de uma base de dados, que será descrita abaixo.

2.5 Criação da base de dados

Para criar a base de dados, foi necessário classificar as imagens, além de registrar as informações espaciais de interesse. Para isso, foi desenvolvido um programa que auxilia no cadastramento das imagens [11]. As informações de interesse para classificação das imagens são:

Tabela 2.2: Campos registrados para cada imagem da base de dados.

Campo	Conteúdo
Nome do Arquivo	Indica o nome do arquivo
Placa na Imagem	Completa, parcial, não existe
Qualidade da Placa	Boa, com sombras, amassada, ilegível
Cor do Pára-choques	Escuro, claro
Tipo de Placa	Cinza, vermelha, outras
Número da Placa	<i>String</i> de ASCII com os caracteres da placa
Traço	Coordenadas (x,y) do traço separador (em <i>pixels</i>)
Caracteres	Coordenadas superiores-esquerdas e inferiores-direitas dos caracteres que compõem a placa (em <i>pixels</i>)
Placa	Coordenada superior-esquerda e inferior-direita da placa (em <i>pixels</i>)

Até o início dos trabalhos, em torno de 5000 imagens haviam sido cadastradas. Destas, 5000 imagens, 1250 foram classificadas como placas cinza, completas e boas. É em cima dessas 1250 imagens que o trabalho será feito.

Nos próximos capítulos, falaremos da etapa de desenvolvimento e dos estudos feitos para a construção dos filtros casados, bem como o desenvolvimento do pré-

processamento e a etapa de localização, baseada nos filtros casados construídos. Ainda será discutida a eficiência do sistema, a partir da análise de localização.

Capítulo 3

Pré-Processamento

Neste capítulo, será abordado o pré-processamento utilizado na imagem original, com o intuito de restringir a área de busca do filtro casado pela imagem da placa sob análise. Isto trará tanto benefícios do ponto de vista da velocidade, diminuindo o tempo de varredura de uma imagem original pelo algoritmo de detecção, bem como do ponto de vista da eficiência de detecção, evitando-se a possível detecção de algumas áreas de ruído (ausência de placa). A discussão se iniciará com a filtragem freqüencial das linhas, passando em seguida pela utilização do algoritmo de janela deslizante e, pela escolha das regiões de maior probabilidade de ocorrência da placa. Por último, falaremos sobre a escolha do valor dos parâmetros e as eficiências encontradas.

3.1 Análise freqüencial

A técnica de pré-processamento proposta consiste, basicamente, em filtrar os componentes de freqüência indesejados, formando uma nova imagem que mostre apenas as freqüências que estejam dentro de uma faixa de interesse, definida mais adiante. Vale lembrar que, como as imagens a serem processadas apresentam informação significativa de ruído⁵, apenas a filtragem freqüencial não será suficiente para a localização da placa, uma vez que ainda existirão componentes de ruído dentro dessa faixa. Conseqüentemente, quanto mais estreita essa faixa, menor quantidade de informação indesejada restará. Todavia, não podemos estreitar tanto, ao ponto de

⁵ Ruído, no nosso caso de estudo, é toda e qualquer informação que não seja a placa alvo.

começar a eliminar informações relevantes da placa. Assim, objetiva-se eliminar as informações de ruído e preservar aquilo que for importante [1].

Tipicamente, no processamento digital de sinais, considera-se que a faixa de frequência varia dentro do intervalo abaixo:

$$\omega \in [-\pi, \pi], \text{ onde } \omega = \text{frequência angular digital} \quad (3.1)$$

Neste caso, fica claro que a maior componente possui frequência igual a π e que foi amostrada a uma frequência de 2π . No processamento digital de imagens, é comum se normalizar a frequência de amostragem com o valor 1 (o período também é igual a 1), logo a faixa de frequência varia dentro da faixa abaixo:

$$\omega \in [-0,5, 0,5] \quad (3.2)$$

Abaixo segue uma imagem representando uma frequência de 0,5 e outra que exemplifica o porquê de não se conseguir frequências maiores:

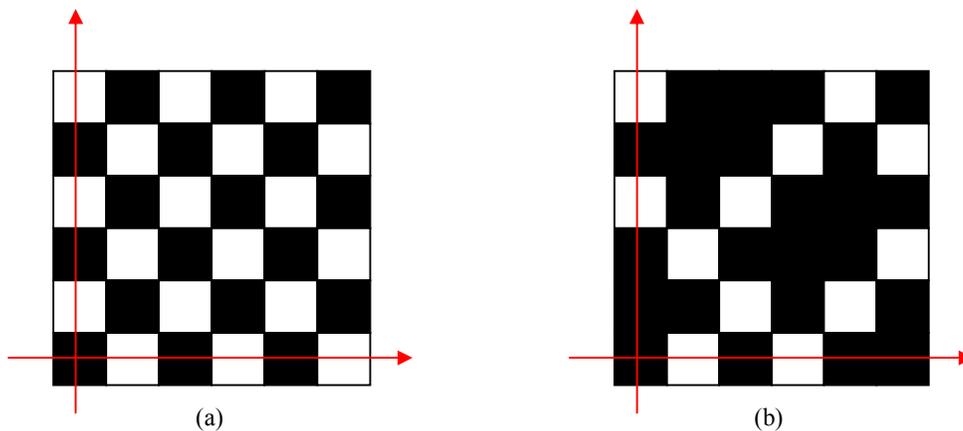


Figura 3.1: Imagens com diferentes frequências espaciais (a) Imagem que possui $\omega = 0,5$; (b) Imagem que possui $\omega < 0,5$.

Não é possível alterar a figura 3.1(a) sem que a frequência diminua, já que qualquer alteração fará com que a variação entre os *pixels* diminua também. O mesmo aconteceria se tentássemos fazer o contrário do que foi feito na figura (b), isto é, apagar alguns pixels.

Como mostrado em [1], a fim de estudar a variação entre o fundo e caracteres contidos nas imagens das placas, o primeiro passo consiste em fazer o recorte para obter apenas as imagens das placas de algumas imagens da base de dados.

Tendo em mãos as novas imagens, formadas apenas por placas selecionadas do conjunto de desenvolvimento, calculou-se a DFT em três linhas de cada imagem, escolhidas aleatoriamente dentro da área dos caracteres de identificação da placa, com a finalidade de identificar a faixa de frequência característica.

A figura 3.2, mostrada a seguir, exemplifica tal processo. São extraídas três linhas diferentes da placa. Os gráficos superiores dos pares de gráfico representam as funções de luminosidade (em níveis de cinza) de cada linha e os inferiores representam as componentes freqüenciais.

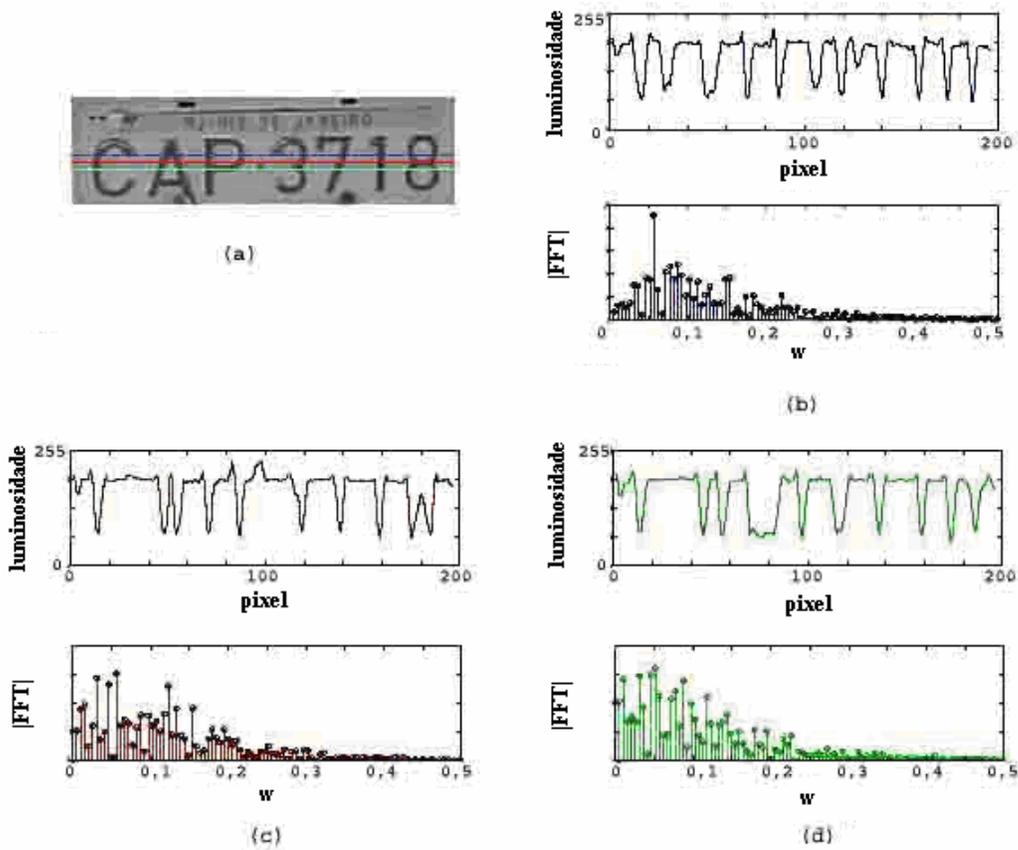


Figura 3.2: Exemplo de levantamento de parâmetros de freqüência: Imagem da placa recortada, cortada por três linhas escolhidas aleatoriamente, dentro da área dos caracteres de identificação (a) e os gráficos da função de luminosidade, em níveis de cinza, por pixel (superior) e de seus respectivos módulos de potência por freqüência digital (inferior) da 1ª linha (b); da 2ª linha (c) e da 3ª linha (d).

Para o levantamento da freqüência característica das placas na base de desenvolvimento atual, escolheu-se analisar a linha que passasse pelo traço separador

das letras e números. Essa escolha se deu pelo fato que, com essa linha, seria garantido que todos os caracteres fossem incluídos na análise. Calculando a DFT sobre esta linha das placas do conjunto de desenvolvimento, foi possível construir um gráfico acumulativo (mostrado na figura 3.3) e, através de inspeção direta, verificar que realmente existe uma faixa de frequência característica bem definida e que essa faixa de frequência característica horizontal da placa $[f_1, f_2]$ está entre 0,03 e 0,06.

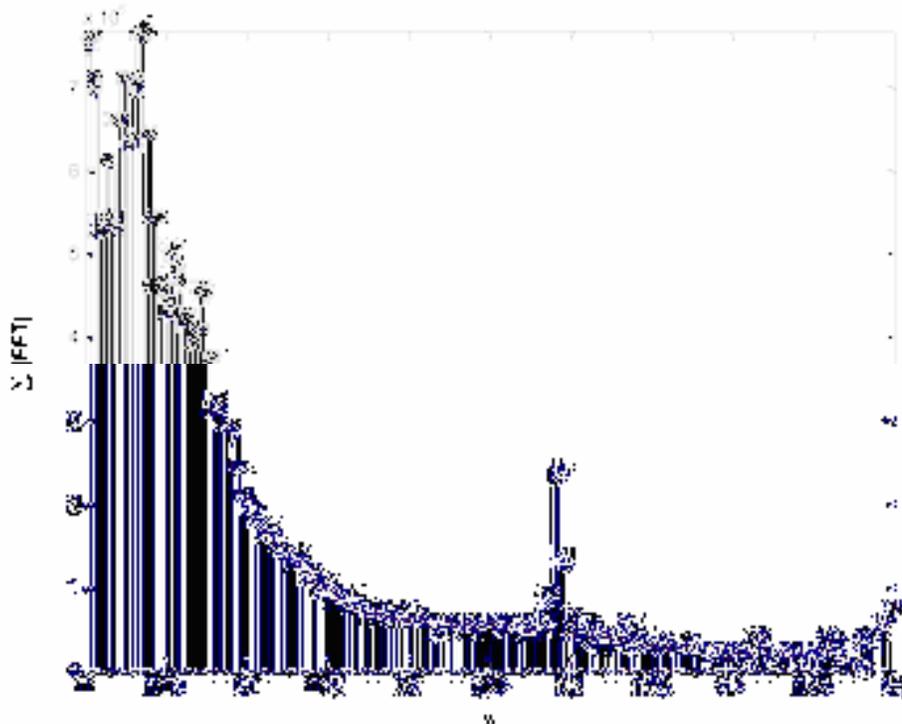


Figura 3.3: Gráfico acumulativo para validação da faixa de frequência horizontal sobre as placas do conjunto de desenvolvimento.

No gráfico acima, é possível perceber uma componente de frequência representativa perto de 0,3. Ao fazer a aquisição das imagens, a câmera introduziu um ruído, através de rajadas nas imagens, com uma frequência bem definida. Todavia, este ruído é

facilmente filtrado (através da filtragem passa-faixa que será utilizada no pré-processamento) e não interferirá no resultado final.

Tendo em mãos a análise freqüencial, podemos começar a detalhar a maneira como a filtragem é utilizada no algoritmo: sobre cada linha da imagem original $I(x,y)$ é aplicada a DFT, gerando uma nova imagem $F(x,y)$ composta apenas pelo módulo da resposta em freqüência de cada linha⁶, segundo a fórmula abaixo:

$$F(k,y) = |DFT_x[I(x,y)]| \quad , 0 \leq y \leq N-1 \quad (3.3)$$

onde:

$$DFT_x = \text{DFT na direção } x$$

Na figura abaixo, são mostradas a imagem original I e a imagem F formada pelo módulo das respostas em freqüência de cada linha:



Figura 3.4: Imagem original I e a imagem F resultante, formada pelo módulo da resposta em freqüencial de cada linha.

⁶ Em momento algum foram levadas em consideração informações sobre as fases dos sinais.

Como o módulo da saída da DFT é sempre simétrica entre as frequências normalizadas -0,5 e 0,5, a imagem resultante também o será. Por isso, optamos apenas por representá-la entre 0 e 0,5.

Como a relação abaixo existe:

$$\mathbf{I}(x,y) \leftrightarrow \mathbf{F}(k,y), \text{ onde } 0 \leq k \leq 0,5 \quad (3.4)$$

isto é, a transformada da linha y de \mathbf{I} corresponde à linha y em \mathbf{F} , é possível perceber, através da figura 3.4, que, em \mathbf{F} , na altura da placa em \mathbf{I} , temos componentes que se destacam dentro de uma faixa de frequência específica. Essas componentes, como visto anteriormente, variam entre 0,03 e 0,06. Como a faixa de frequência está bem caracterizada, já é possível começar a projetar filtros passa-banda que preservem apenas esta faixa.

O que facilita o trabalho na construção do filtro é o fato de, após a filtragem, não haver reconstituição do sinal, isto é, após feita DFT e em seguida a filtragem das componentes frequenciais, não há a necessidade de se reconstruir a imagem original (voltando a trabalhar no domínio do espaço) através de uma DFT inversa. Uma vez feita a DFT 1D na imagem, construímos uma nova imagem e é em cima dela que todo o processamento é feito. Podemos, inclusive, abstrair que estamos trabalhando no domínio da frequência.

Abaixo é mostrado através de (3.5) e (3.6) a resposta impulsional para o caso de um filtro passa-baixa:

$$h[n] = \frac{\text{sen}[\omega n]}{\pi n} \quad , \text{ para } n \neq 0 \quad (3.5)$$

$$h[n] = \frac{\omega}{\pi} \quad , \text{ para } n = 0 \quad (3.6)$$

onde:

ω = frequência de corte do filtro passa-baixa

Ao aproximar essas seqüências infinitamente longas através do truncamento de $h[n]$, isto é, através do truncamento abrupto da Série de Fourier infinita, um efeito oscilatório, conhecido como fenômeno de Gibbs aparece [12]. Além disso, já foi mostrado que a máxima magnitude referente à oscilação é basicamente constante, mesmo variando o comprimento da seqüência truncada [12].

A partir das características do problema, que nos permitem utilizar uma janela ideal, conforme explicado acima, podemos simplesmente apresentá-las abaixo, através de (3.7), como sendo o filtro escolhido:

$$J(k) = U(k - f_1) - U(k - f_2) \quad (3.7)$$

onde:

$U(k)$ = Degrau Unitário

f_1 = frequência de corte inferior

f_2 = frequência de corte superior

Com isso, a nova imagem **B** (figura 3.5), formada pela imagem **F** filtrada, é dada por (3.8), (3.9) e (3.10):

$$\mathbf{B}(k,y) = 0 \quad , \text{ para } k < f_1 \quad (3.8)$$

$$\mathbf{B}(k,y) = \mathbf{F}(k,y) \quad , \text{ para } f_1 \leq k \leq f_2 \quad (3.9)$$

$$\mathbf{B}(k,y) = 0 \quad , \text{ para } k > f_2 \quad (3.10)$$

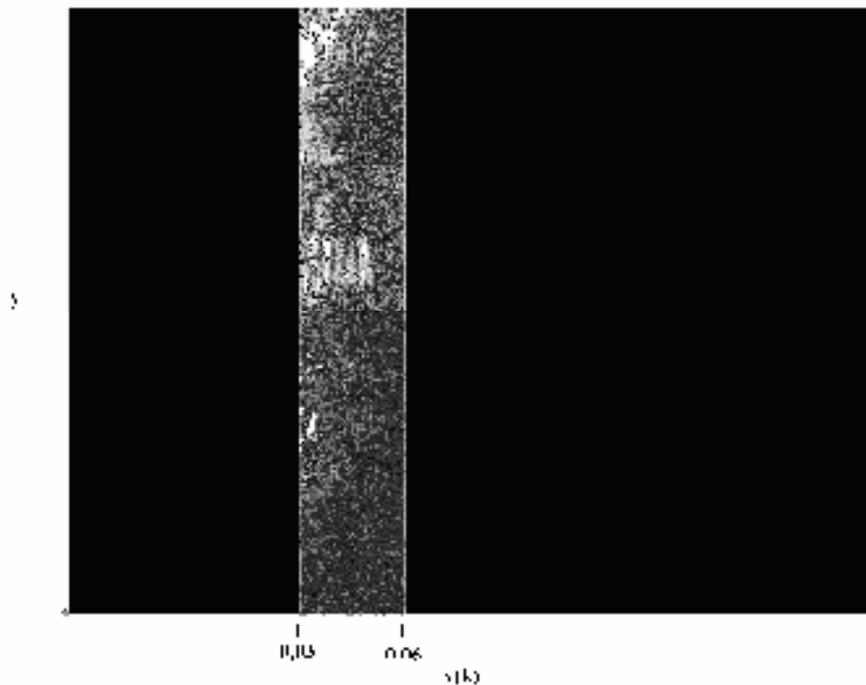


Figura 3.5: Exemplo de uma imagem já filtrada (imagem **B**).

Formada a imagem **B**, podemos passar para a próxima seção, onde será explicada a função de janela deslizante.

3.2 Janela deslizando

Antes de começar a discussão sobre a função da janela deslizando, devemos falar sobre o pré-processamento que a imagem **B** sofre, a fim de transformá-la em um vetor. Isso é necessário porque, em nosso caso, optou-se por trabalhar com janelas deslizantes de dimensão unitária [1].

A imagem **B**, uma matriz de 320 x 240, tem valores nulos fora da faixa de frequência de interesse e possui os mesmos valores de **F** para valores dentro da faixa, conforme indicam (3.8), (3.9) e (3.10). Para identificar a linha que contém a maior energia dentro da faixa de interesse, basta fazer o somatório dos elementos de cada linha e identificar qual somatório recebeu o valor mais alto. Para receber esses valores, se fez necessário criar um vetor, cujos elementos correspondem aos somatórios das linhas da imagem.

De uma maneira mais formal, podemos formular que:

$$v(y) = \sum_{k=0}^{M-1} B(k, y) \quad , 0 \leq y \leq N-1 \quad (3.11)$$

onde:

$$M = \text{Largura da imagem } \mathbf{B}$$

A figura 3.6 mostra um exemplo da operação feita por (3.11):

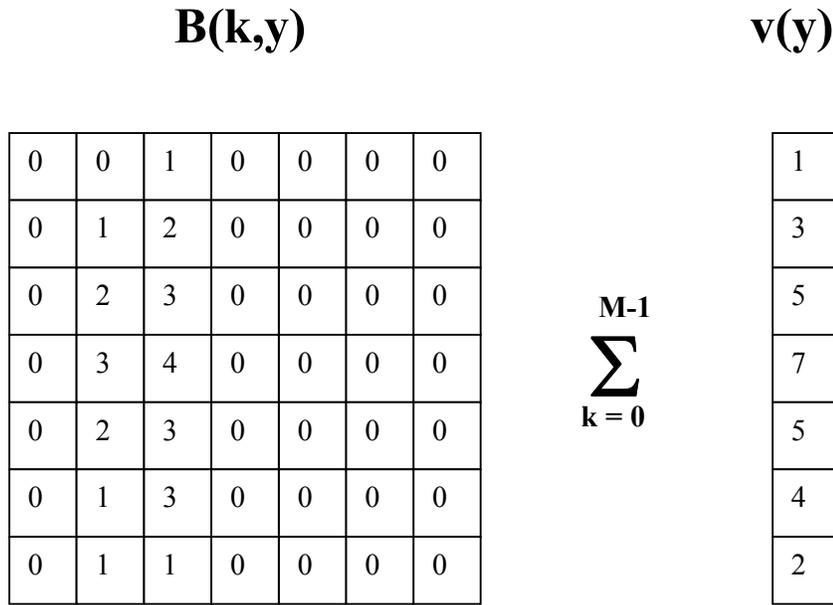


Figura 3.6: Exemplo do somatório feito pela equação (3.11).

É justamente sobre o vetor v que uma janela deslizante é aplicada. Esta janela tem como objetivo calcular as energias das regiões que compõem a imagem já filtrada.

A idéia básica da janela deslizante é formar um novo vetor s que contenha informações de intensidade das regiões do vetor v . O processamento consiste em uma janela, com T elementos, que varre o vetor v em passos de um elemento. Para cada passo é feito o somatório dos elementos contidos dentro da janela. Por fim, o valor deste somatório é passado para o vetor s :

$$s(y) = \sum_{y=i}^{T+i} v(y) \quad , 0 \leq y \leq N-T+1 \quad (3.12)$$

onde:

T = Tamanho da janela aplicada

Tendo em mãos o vetor s e o tamanho T da janela, já é possível estimar a altura de ocorrência da placa na imagem. Para isso, escolhe-se, em s o elemento de maior valor. O índice deste elemento corresponde ao início da região, na imagem, que contém a maior intensidade, dentro da faixa de frequência de interesse. Conforme dito acima, é importante notar que a informação corresponde ao início da região com maior intensidade, e é mais interessante que o pré-processamento retorne o centro desta região. Encontrar este ponto é simples: basta somar ao índice escolhido, metade do tamanho da janela utilizada.

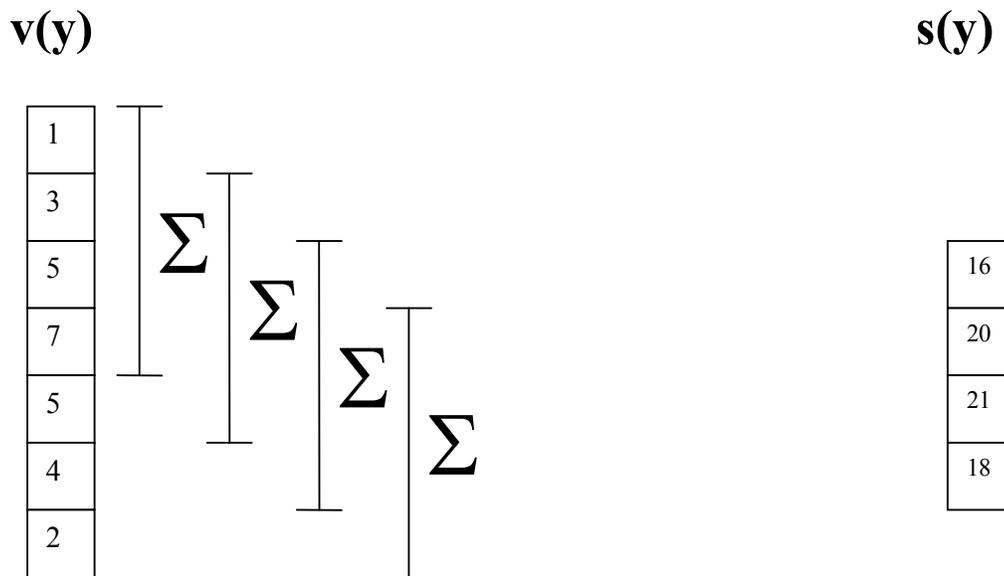


Figura 3.7: Exemplo do somatório feito pela equação (3.12), utilizando o método de preenchimento com zeros.

3.3 O algoritmo

A figura 3.8 apresenta os passos do algoritmo de pré-processamento, juntamente com seus conjuntos de entrada e saída. Sobre a imagem adquirida em níveis de cinza, aplica-se uma DFT 1D na direção horizontal. O resultado dessa transformada é uma

nova imagem, apresentando as componentes de frequência horizontal da imagem anterior. Sobre a imagem da transformada, um filtro passa-faixa ideal com banda passante entre f_1 e f_2 é usado. Nesta imagem filtrada, é feito um pré-processamento, que consiste em calcular o somatório de energia de cada linha, gerando, dessa forma, um vetor contendo as informações de energia de cada linha da imagem filtrada. Sobre este vetor é aplicado uma janela deslizante, de tamanho T , a fim de calcular as energias de cada região. A partir dessas informações de energia das regiões, o algoritmo escolhe como regiões de maior probabilidade de ocorrência da placa aquelas de maior concentração de energia. Desta forma, o algoritmo pode retornar mais de uma área como saída. É sobre estas áreas selecionadas no pré-processamento que o filtro casado atuará.

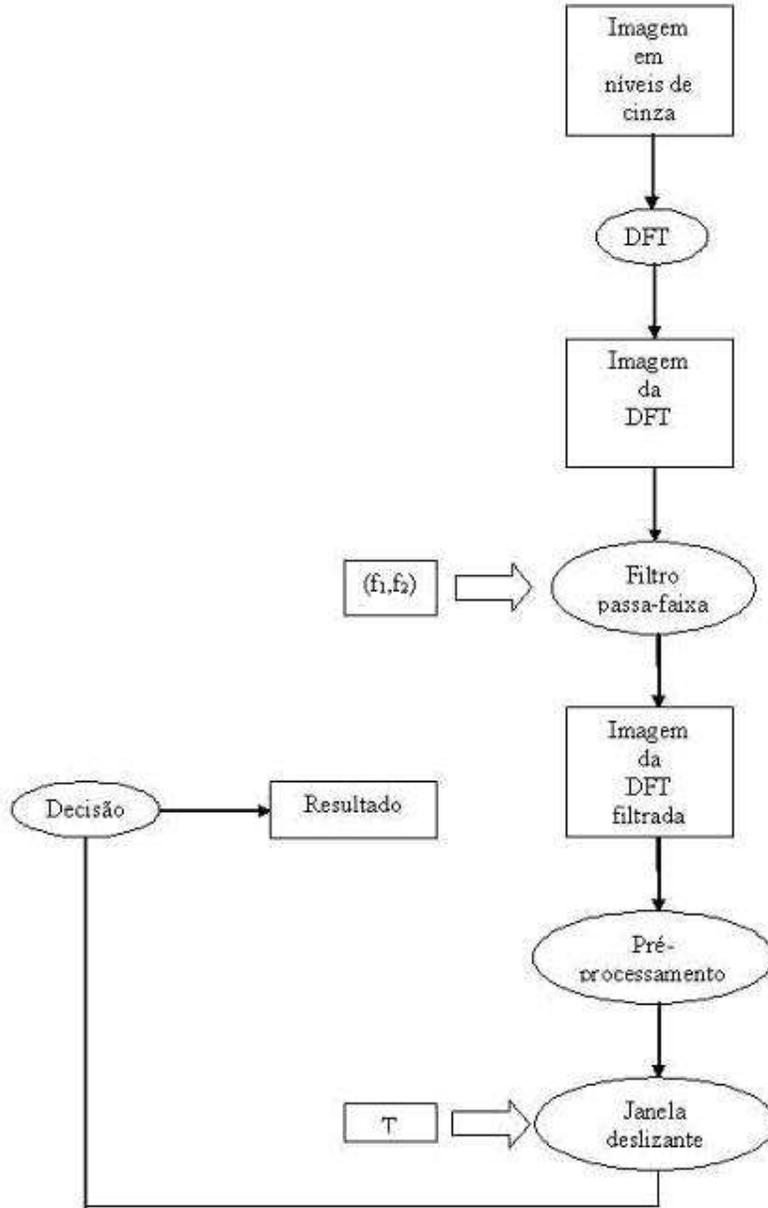


Figura 3.8: Diagrama com os passos do algoritmo, juntamente com seu conjunto de entradas.

3.4 Parâmetros do pré-processamento

No pré-processamento, alguns parâmetros, como a faixa de frequência do filtro e o tamanho da janela deslizante, devem ser determinados. Nesta seção, serão apresentados esses parâmetros, os métodos de escolha de seus valores e, por fim, os valores adotados.

Basicamente, são seis parâmetros utilizados pelos algoritmos: limite inferior do filtro⁷ (ll), limite superior do filtro⁸ (ul), tamanho da janela deslizante⁹ (T), número de respostas de áreas da imagem (numResp), distância entre as respostas¹⁰ (distResp) e tamanho do recorte¹¹ (tamRecorte). Estes últimos três parâmetros são ilustrados mais a seguir na figura 3.10.

A análise dos dois primeiros valores foi apresentada anteriormente, na discussão sobre a análise frequencial, por isso passaremos para a análise do tamanho da janela deslizante.

O objetivo da janela é localizar as regiões de maior energia dentro da faixa de frequência escolhida. Como a faixa de frequência desejada é consequência do contraste entre as letras (pretas) e o fundo da placa (cinza claro), é de se imaginar que a janela (vertical) deve ter tamanho máximo igual à altura média dos caracteres. Desta forma, uma análise de altura dos caracteres da base de desenvolvimento foi realizada, obtendo-se (ver figura 3.9):

$$\text{Altura} = 20 \pm 1 \text{ (pixel)};$$

Desta maneira, definiu-se que o tamanho da janela como 21 *pixels* (ou, eventualmente, menos).

⁷ Frequência normalizada

⁸ Frequência normalizada

⁹ Número de pixels

¹⁰ Distância entre as áreas de resposta (em pixel)

¹¹ Número de linhas recortadas da imagem (em pixel)

Os parâmetros número de respostas, distância entre as respostas e tamanho do recorte foram levantados de forma empírica. Todavia, algumas considerações iniciais devem ser levadas em conta. Quanto maior o número de respostas desejadas, maior será a probabilidade do pré-processamento encontrar uma região de placa, porém maior será o número de regiões, e conseqüentemente, a área de procura do filtro.

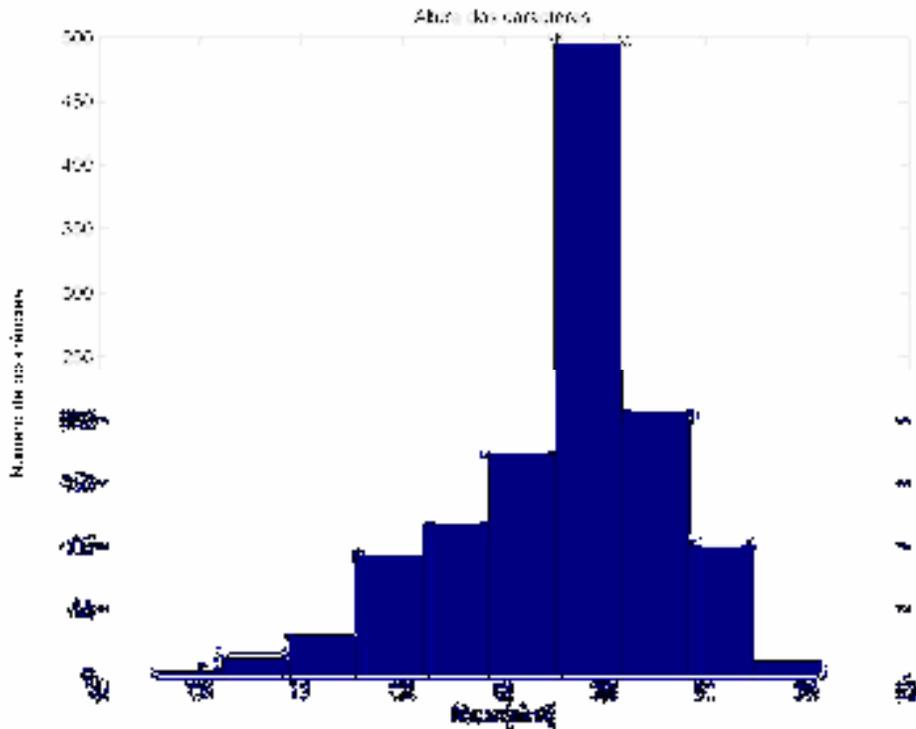


Figura 3.9: Histograma de altura dos caracteres das placas da base de treinamento.

Por outro lado, a distância entre os recortes diz respeito à escolha das distâncias entre as regiões de maior probabilidade de ocorrência da placa. Se escolhermos um valor muito pequeno, a probabilidade das regiões escolhidas serem próximas (ou com grande sobreposição) cresce. Por isso, tende-se a procurar os picos de energia, escolhendo distâncias maiores. Por último, o tamanho do recorte refere-se à área, em torno da linha escolhida, que será recortada. Se escolhermos uma região muito pequena, não recortaremos a placa corretamente e se escolhermos uma região muito grande, o filtro atuará sobre uma área maior.

A figura a seguir ilustra alguns dos parâmetros aqui citados:

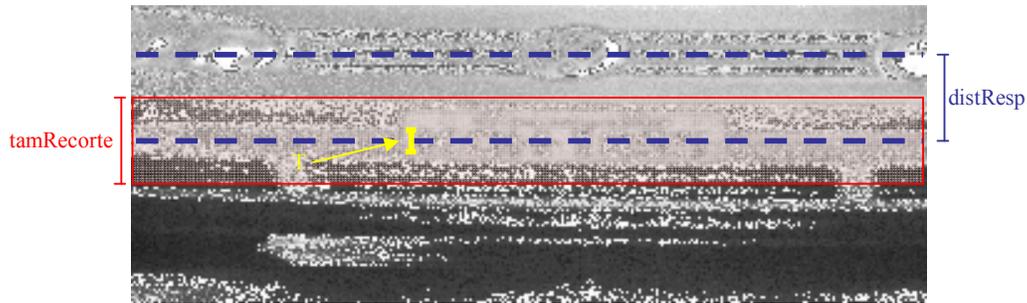


Figura 3.10: Ilustração dos valores de tamanho da janela deslizante (T), distância entre as respostas (distResp) e tamanho do recorte (tamRecorte).

3.5 Resultados

Sob o conjunto de teste, um conjunto de 125 imagens, a tabela abaixo mostra a eficiência¹² dos algoritmos de pré-processamento para alguns conjuntos de parâmetros¹³:

Tabela 3.1: Eficiências de localização do pré-processamento na base de testes.

ll	ul	T	numResp	distResp	tamRrecorte	eficiência
0,03	0,06	18	2	30	60	0,840
0,03	0,06	18	2	35	50	0,736
0,03	0,06	18	2	35	60	0,840
0,03	0,06	18	2	35	65	0,888
0,03	0,06	18	3	35	60	0,920
0,03	0,06	18	4	35	60	0,928
0,03	0,06	18	3	30	50	0,112
0,03	0,06	19	2	30	50	0,032
0,03	0,06	20	2	30	50	0,856
0,03	0,07	20	2	30	50	0,832
0,02	0,08	20	3	30	50	0,872
0,02	0,08	20	2	30	50	0,744

¹² A eficiência do pré-processamento consiste em localizar regiões que contenham completamente a placa. Se existir uma região, dentro daquelas retornadas pelo algoritmo, que contenha por completo a placa, considera-se um acerto, caso contrário, é considerado um erro.

¹³ Os resultados da tabela apresentam incerteza de 0,008.

Geralmente, o algoritmo erra quando encontra áreas com frequências características próximas às da placa. Podemos exemplificar esses erros com áreas que contenham faróis e/ou logomarcas de fabricantes. A figura abaixo ilustra esses erros:

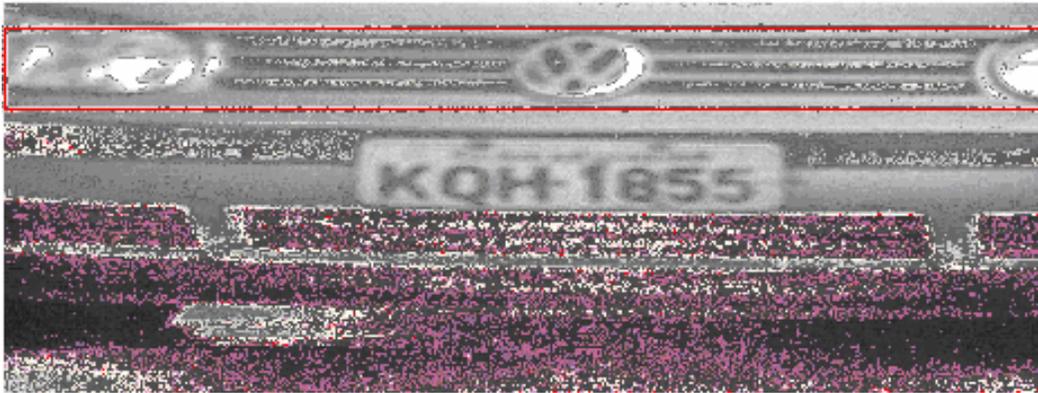


Figura 3.11: Exemplo de região errada localizada pelo pré-processamento

A partir da tabela 3.1, já é possível definir os melhores parâmetros do pré-processamento. Os resultados da tabela apresentam uma incerteza de 0,8%, desta forma destacamos dois resultados: 92,8% e 92,0% (ambos dentro da mesma faixa de incerteza). A grande diferença entre os parâmetros de entrada dos dois resultados apresentados é o número de respostas do pré-processamento. O conjunto de parâmetros de entrada que gera 92,8% de eficiência tem um número de respostas na etapa de pré-processamento igual a 4, o que faz com que a área de atuação do filtro seja maior, aumentando a probabilidade de erro da filtragem por filtro casado (para a eficiência de 92,0%, o filtro casado atuará numa área menor, uma vez que para essa eficiência de pré-processamento o número de respostas é igual a 3). Por isso é importante ressaltar que apesar de um conjunto de parâmetros apresentar maior eficiência de localização na etapa de pré-processamento, pode não se mostrar tão eficiente na etapa final (etapa do filtro casado), já que o número de regiões de busca, e por consequência a área de procura, será maior.

Outro ponto importante é que esta eficiência, a eficiência de pré-processamento, é limitante para a eficiência total de localização do sistema. Mesmo que o filtro casado obtenha uma eficiência de localização de 100% em sua etapa, a eficiência do sistema

estará limitada a 92,8%, exclusivamente por conta dos limites de eficiência do pré-processamento.

Nos próximos capítulos, quando analisarmos as eficiências dos filtros casados desenvolvidos, será mostrada a importância de se utilizar o pré-processamento. Serão mostrados resultados da utilização do filtro, direto nas imagens originais, sem exclusão de áreas feita pelo pré-processamento. Com os resultados alcançados, justificaremos plenamente a utilização deste pré-processamento.

Capítulo 4

Filtro Casado

Neste capítulo, será abordada a construção dos filtros casados, baseado nas análises estatísticas de padrões existentes nas imagens dos veículos. Iniciaremos nossa discussão revendo os aspectos básicos de filtragem ótima, com o intuito de facilitar o entendimento da escolha deste tipo de técnica de identificação de padrões.

4.1 Filtros casados

As origens da teoria de detecção de sinais vêm da 2ª Guerra Mundial quando pesquisadores começaram a explorar a possibilidade de substituir a decisão humana, na detecção de sinais mostrados nas telas de radares, por dispositivos de decisão automática. Um dos resultados mais antigos neste campo é o filtro casado, introduzido pelo físico D.O. North em 1943 [13].

O objetivo dos filtros casados é maximizar a relação sinal-ruído da saída, para um sinal (padrão) conhecido corrompido por ruído, através da comparação de dois sinais (o sinal de entrada e o sinal do filtro). De uma forma simplificada, o filtro casado é um correlator.

Para demonstrar a equivalência entre um correlator e um filtro casado, consideremos um filtro linear e invariante no tempo com uma resposta impulsional $h(t)$. Com o sinal

recebido $x(t)$ usado como entrada do filtro causal, a saída resultante, $y(t)$, é definida pela integral de convolução [14]:

$$y(t) = \int_0^t x(\tau)h(t - \tau)d\tau \quad (4.1)$$

Pela definição de filtros casados, a resposta impulsional $h(t)$ de um filtro casado linear invariante no tempo para um sinal de entrada $\phi(t)$ é uma versão invertida no tempo e atrasada da entrada $\phi(t)$. Suponhamos que ajustemos a resposta ao impulso para:

$$h(t) = \phi(T - t) \quad (4.2)$$

A saída resultante do filtro é:

$$y(t) = \int_0^t x(\tau)\phi(T - t + \tau)d\tau \quad (4.3)$$

Amostrando essa saída com um período de $t = T$, temos:

$$y(T) = \int_0^T x(\tau)\phi(\tau)d\tau \quad (4.4)$$

Uma vez que, por definição, $\phi(t)$ é zero fora do intervalo $0 \leq t \leq T$, concluímos que $y(t)$ é, na verdade, a saída do correlator y , produzido pelo sinal recebido $x(t)$, como é mostrado por:

$$y(T) = \int_0^T x(\tau)\phi(\tau)d\tau \quad (4.5)$$

Para o nosso caso de estudo, o caso em duas dimensões (2D), podemos extrapolar a equação (4.5) e chegamos a:

$$z(X, Y) = \int_0^X \int_0^Y I(x, y)\phi(x, y)dydx \quad (4.6)$$

onde: z – saída do filtro

I – imagem de entrada

ϕ – filtro casado

$0 \leq x \leq X$

$0 \leq y \leq Y$

4.2 Projeto dos filtros casados

Antes de passar para o projeto dos filtros propostos neste trabalho, é necessário fazer uma análise dos dados disponíveis, buscando estimar as flutuações dos parâmetros relevantes ao projeto dos filtros. A análise se restringe ao conjunto de desenvolvimento ou a um subconjunto deste.

4.2.1 Análise estatística

Nesta análise, as flutuações estatísticas da placa, do traço separador de letras e números contidos na placa, dos parafusos que fixam a placa ao veículo, bem como os caracteres que estão contidos na placa serão avaliadas.

4.2.1.1 Placa

Utilizando as informações de posicionamento marcadas na base de dados criada, foi possível estimar a flutuação nas dimensões das placas. Para este estudo, foram utilizadas as 1125 imagens da base de desenvolvimento.

A partir das medidas de cada placa, foi obtida a distribuição da largura e da altura das placas, calculando-se estimadores (média e o valor RMS). Os resultados são mostrados na figura 4.1:

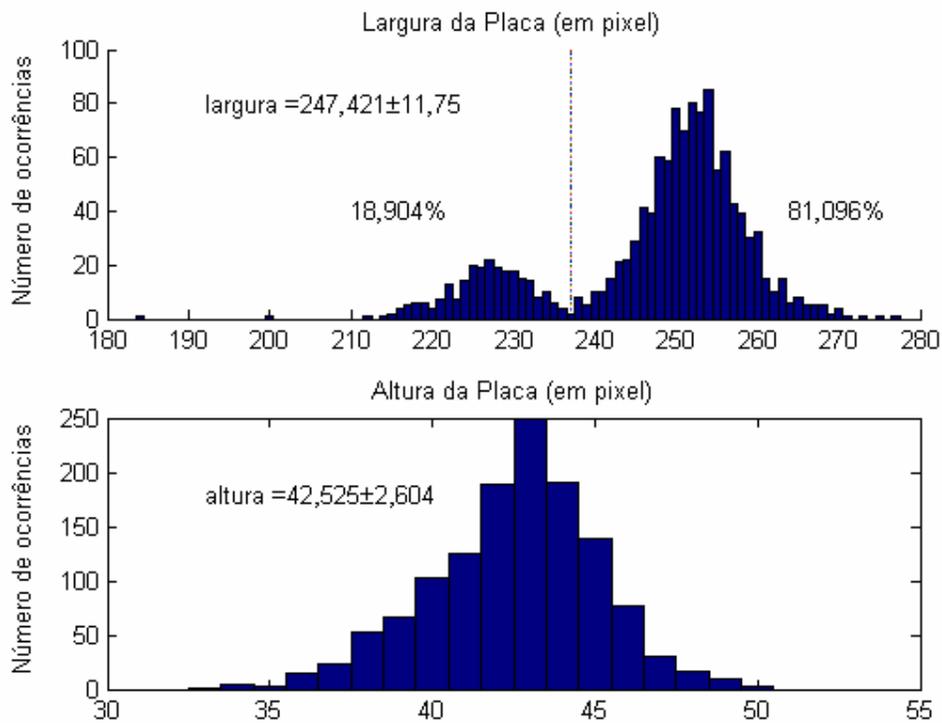


Figura 4.1: Gráfico de distribuição de largura e altura das placas da base de desenvolvimento. No gráfico de distribuições das larguras, a amostragem nos leva a duas distribuições normais, com o limiar de separação em 237.

Neste estudo, verificou-se que a distribuição de alturas das placas apresentou uma forma próxima de uma normal, mas o mais importante é que a distribuição apresentou um valor RMS pequeno, em torno de 6% em relação ao valor médio. Já no caso da largura, esta apresentou uma distribuição próxima de duas normais, com um limiar de separação em 237, onde 18,904% das placas estão na normal inferior e 81,096% estão na normal superior. Todavia, verificou-se que o valor RMS da distribuição está em torno de 4,7% do valor médio, sendo, inclusive, menor que o valor RMS obtido no caso da altura.

A partir dessas novas informações obtidas, parâmetros de redimensionamento poderão ser utilizados para a construção de novos filtros da placa. Baseados nas informações de

todas as amostras do conjunto de desenvolvimento, serão construídos três filtros: um com a largura média de todas as amostras, outro com a largura média da distribuição das placas de largura maior que 237 *pixels* e um terceiro com a largura média da distribuição das placas com largura menor que 237 *pixels*.

4.2.1.2 Traço

Foram feitos alguns estudos estatísticos sobre o traço. O primeiro em relação ao tamanho do traço (largura e altura) e o restante em relação à distância do traço às bordas superior e inferior da placa e os caracteres adjacentes a ele (a última letra e o primeiro número da placa).

Com base nas informações contidas no banco de dados, três diferentes filtros são propostos. O primeiro usa como referência os limites superior e inferior da placa, bem como o limite lateral direito da última letra e o limite lateral esquerdo do primeiro número. A figura abaixo (figura 4.2) ilustra esse primeiro caso.

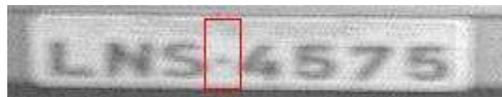


Figura 4.2: Limites propostos para o padrão do filtro de traço.

O segundo filtro passa a usar apenas referências espaciais dos caracteres, por isso os limites superior e o lateral esquerdo do recorte são baseados nos limites superior e direito da última letra enquanto que os limites inferior e lateral direito do recorte

baseados nos limites inferior e esquerdo do primeiro número. A próxima figura 4.3 ilustra o segundo caso:

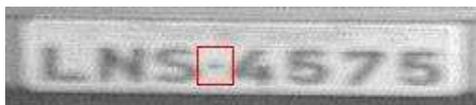


Figura 4.3: Limites alternativos propostos para o padrão do filtro de traço.

O terceiro filtro proposto é uma variação do segundo, com o intuito de corrigir problemas de inclinação das placas. Por isso, ao contrário do segundo caso, quando os limites superior e inferior eram obtidos, respectivamente, através das informações de limite superior da última letra e limite inferior do primeiro número, neste caso usamos como referência de limite superior a média entre os limites superiores da última letra e do primeiro número e como limite inferior, a média entre os limites inferiores da última letra e do primeiro número. A figura 4.4 ilustra esse caso:



Figura 4.4: Limites propostos para o padrão de um terceiro filtro de traço.

Como veremos mais adiante, a variação de tamanho entre o segundo e o terceiro filtro é muito pequena e não deve alterar as eficiências de localização da placa.

Como não havia a informação de tamanho do traço na base de dados, foi necessário fazer um pré-levantamento manual de alguns deles. Para isso, foram escolhidas 48 placas da base para dar início ao estudo.

O gráfico abaixo mostra a análise de tamanhos obtidos para 48 placas:

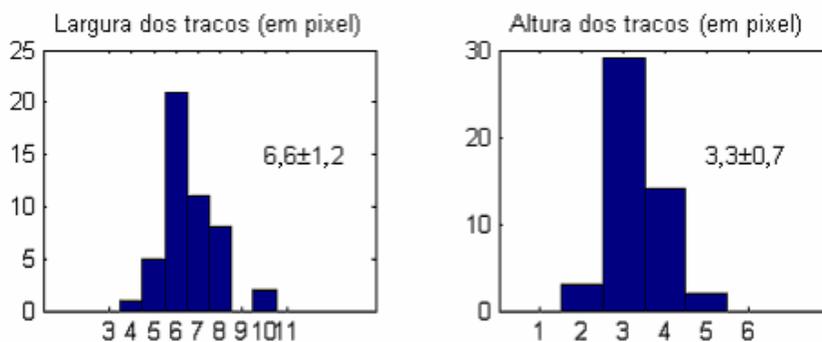


Figura 4.5: Distribuição de largura e altura dos traços.

O valor RMS alto (em torno de 20%) das dimensões do traço fez com que a construção deste filtro (filtro baseado no traço, sem as informações de fundo adjacentes) fosse abandonada, já que o recorte automático não seria eficiente.

No estudo relativo às distâncias do traço em relação às bordas da placa e dos caracteres adjacentes, a base de dados criada já possui todas as informações de posicionamento utilizadas, por isso, o trabalho consistiu em levantar o tamanho ideal de cada filtro para que as eficiências de localização fossem otimizadas.

Para esse desenvolvimento foram utilizadas as 1125 imagens da base de desenvolvimento.

(a) Construção do primeiro filtro

Baseado nas informações de tamanho, obtidas através de histograma, cálculos de distância média e valor RMS, nesse primeiro estudo, chegamos às seguintes dimensões médias:

$$\text{Largura} = 15,5 \pm 3,5 \text{ pixels}$$

$$\text{Altura} = 43,0 \pm 3,0 \text{ pixels}$$

A figura 4.6 ilustra os tamanhos obtidos.

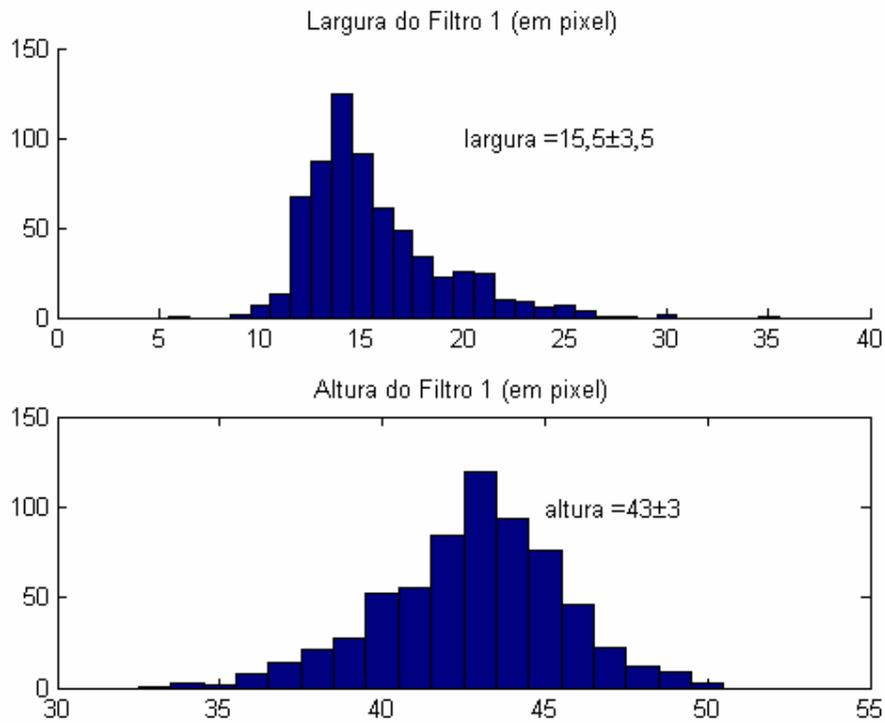


Figura 4.6: Distribuição de largura e altura dos traços do filtro 1.

(b) Construção do segundo filtro

Baseado nas informações de tamanho, obtidas através de histograma, cálculos de distância média e valor RMS, nesse primeiro estudo, chegamos às seguintes dimensões médias no estudo do segundo tipo de filtro casado no traço:

$$\text{Largura} = 15,5 \pm 3,5 \text{ pixels}$$

$$\text{Altura} = 19,0 \pm 1,0 \text{ pixels}$$

A figura 4.7 ilustra os tamanhos obtidos.

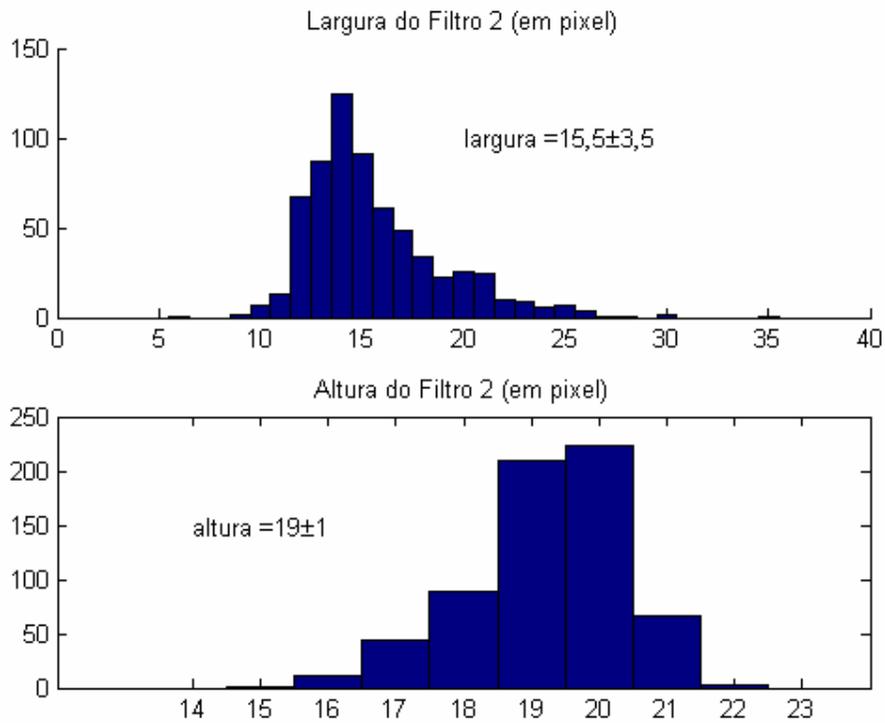


Figura 4.7: Gráfico de distribuição de largura e altura dos traços do filtro 2.

(c) Construção do terceiro filtro

Baseado nas informações de tamanho, obtidas através de histograma, cálculos de distância média e valor RMS, nesse primeiro estudo chegamos às seguintes dimensões médias no estudo do terceiro tipo de filtro baseado no traço

Largura = $15,5 \pm 3,5$ pixels

Altura = $20,0 \pm 1,0$ pixels

A figura 4.8 ilustra os tamanhos obtidos.

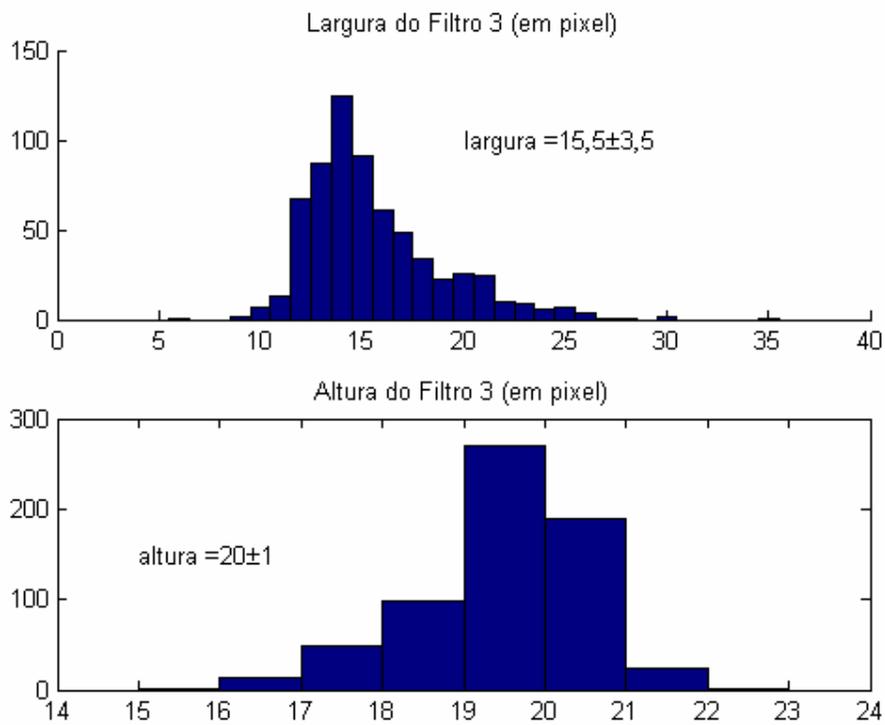


Figura 4.8: Distribuição de largura e altura dos traços do filtro 3.

É possível verificar que a variação de tamanho entre o segundo e o terceiro filtro é muito pequena (1 *pixel* de altura), o que acaba mostrando que o ângulo de inclinação das placas do conjunto de desenvolvimento é muito pequeno.

4.2.1.3 Parafusos

A princípio, o posicionamento dos parafusos das placas não estava inserido na base de dados. Por isso, se fez necessário o cadastramento manual de alguns dos parafusos das placas da base de desenvolvimento. Cada placa possui 4 parafusos fixadores, dois superiores e dois inferiores. Os dois filtros aqui planejados representam duas “tiras”, uma contendo os parafusos superiores e a outra contendo os parafusos inferiores. A figura abaixo ilustra as tiras extraídas das placas:

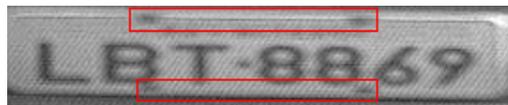


Figura 4.9: Ilustração das tiras que contêm os parafusos.

Durante o processo de medição do tamanho dos parafusos, foi encontrada uma grande dificuldade de diferenciar o parafuso do furo em que esse está encaixado, por isso resolveu-se fazer a medição não mais do tamanho da tira que continha os parafusos, mas sim da que continha os furos que existem nas placas. Por isso, daqui por diante não falaremos mais de parafusos e sim de furos.

Nessa análise, foi feito o cálculo das distâncias entre os furos superiores e também entre os inferiores. A figura 4.10 mostra os resultados encontrados para um grupo de 64 placas:

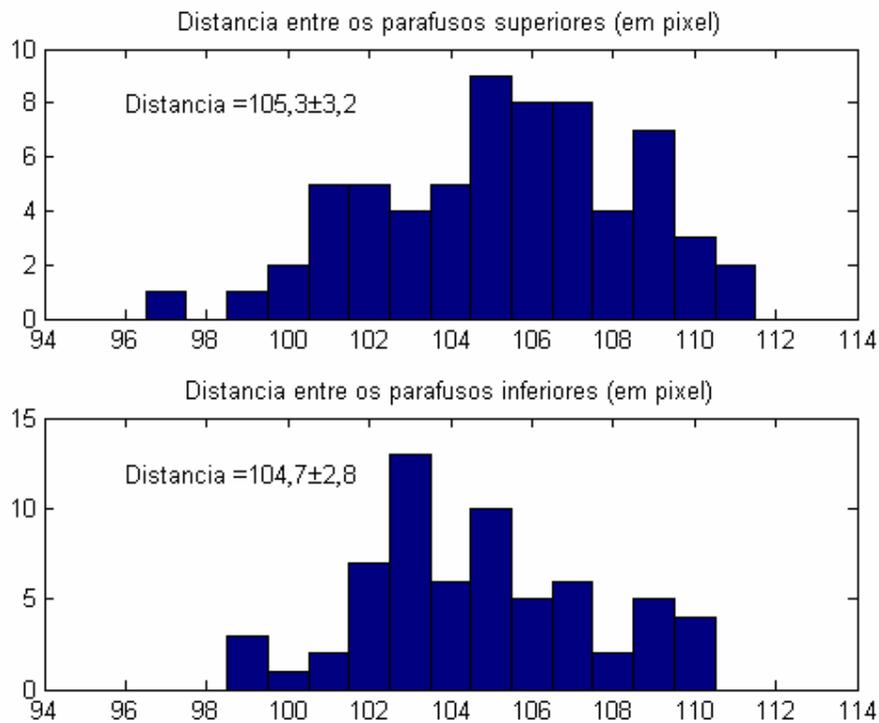


Figura 4.10: Distribuição de distâncias entre os parafusos superiores e inferiores.

Através do estudo de ocorrência de objetos (traços e furos), verificou-se que a grande vantagem do traço como filtro é que ele ocorre em todas as imagens, ao contrário dos furos que não ocorrem¹⁴ em aproximadamente 10% das placas. Por isso, filtros baseados em parafusos não serão utilizados neste trabalho.

¹⁴ Invisíveis ao olho humano durante a inspeção das imagens. Isso não significa que as placas não apresentem furos.

4.2.1.4 Caracteres

No caso dos filtros baseados nos caracteres, também foram utilizadas as informações de posicionamento marcadas na base de dados existente. Para este estudo, foram utilizadas as 1125 imagens da base de desenvolvimento e todos os caracteres dessas placas foram medidos. Neste caso, para não haver perda de informação optou-se por estabelecer os tamanhos dos filtros baseados nos tamanhos das maiores amostras de cada classe. Dessa forma, os redimensionamentos das amostras necessários para a construção dos filtros não provocariam perda de informação já que as amostras menores teriam seus tamanhos aumentados. A tabela 4.1 mostra o tamanho escolhido para cada um dos filtros a serem desenvolvidos:

Tabela 4.1: Tamanhos (largura x altura) definidos para os filtros baseados em cada um dos caracteres (em pixel).

A	B	C	D	E	F	G	H	I
36x22	31x21	30x22	32x22	30x22	29x22	32x22	36x22	12x22
J	K	L	M	N	O	P	Q	R
26x21	37x24	34x23	39x22	34x22	33x23	31x22	36x22	33x23
S	T	U	V	W	X	Y	Z	1
32x21	32x22	31x22	32x21	34x21	34x21	30x22	33x22	19x23
2	3	4	5	6	7	8	9	0
26x22	29x22	28x22	27x22	26x22	28x22	26x23	28x22	26x22

4.2.2 Filtros projetados

Filtros casados podem ser desenvolvidos determinística ou estocasticamente. Filtros casados estocásticos, geralmente, são baseados em análise de componentes principais e levam em consideração apenas as componentes que com maior energia, isto é, componentes que trazem a maior parte da informação relevante do sinal.

Os filtros determinísticos são mais simples e, na maioria das vezes, são baseados em análises estatísticas do padrão que se quer casar. O resultado obtido é tão pior quanto maior for a flutuação entre os sinais que são usados para desenvolver o filtro.

No nosso caso, trabalharemos com filtros casados determinísticos baseados em valores médios das amostras do conjunto de desenvolvimento, sem levar em consideração a flutuação entre as placas que foram usadas para desenvolvimento dos filtros.

Desta forma, os filtros projetados, baseados nos padrões escolhidos, serão desenvolvidos a partir das médias de todas as amostras que compõem o conjunto de desenvolvimento. Com os resultados obtidos no item anterior, foi decidido desenvolver filtros baseados na imagem da placa inteira, na faixa que contém o traço separador das letras e dos números e nos caracteres da placa.

Conforme dito anteriormente, não será desenvolvido um filtro baseado nos parafusos, uma vez que estes não são observáveis em todas as amostras.

Nesta etapa de projeto, é necessário que se determine o tamanho dos filtros, uma vez que as amostras possuem tamanhos diferentes, devido às variações de proximidade dos veículos da câmera no momento da aquisição, causadas pelas diferentes velocidades entre os veículos. Por isso, para que formem os filtros, as amostras devem ser redimensionadas para o tamanho do filtro escolhido.

O redimensionamento utilizado é baseado em interpolação e pode ser das seguintes formas: *nearest neighbor*, *bilinear*, *bicubic* e *alias*. O redimensionamento através de *nearest neighbor* apenas considera o valor do *pixel* de onde se localiza o ponto. Já o baseado em interpolação *bilinear* é baseada na média ponderada dos pixels de uma vizinhança 2x2, enquanto que o *bicubic* se baseia numa vizinhança 4x4. É importante notar que o número de *pixels* altera a complexidade do algoritmo. Por último, a interpolação *alias* é utilizada para diminuir as perdas de informação que ocorrem ao se diminuir o tamanho da imagem. É necessário que seja utilizado um filtro passa-baixas para minimizar este efeito.

Foi observado durante a construção dos filtros casados que os diversos métodos de redimensionamento não alteram de forma significativa os filtros desenvolvidos, por isso optou-se por utilizar o método mais simples de interpolação no redimensionamento, o *nearest neighbor*.

Como a rotina de filtro casado é mais eficiente (rápida) para filtros com largura múltipla de 4, já que o algoritmo de correlação do Matlab também o é, as amostras terão que ser redimensionadas para tamanhos com larguras múltiplas de 4.

Os filtros projetados serão formados a partir da expressão abaixo, já com as amostras da base de desenvolvimento redimensionadas:

$$I_F = \sum_{i=1}^N I_i / N \quad (4.7)$$

onde: I_F = filtro projeto

I_i = i-ésima imagem (amostra)

N = número total de amostras

4.2.2.1 Placa

Como foi mostrado anteriormente, existem dois grupos distintos na distribuição de largura das placas, um formado por placas com largura maior que *237 pixels* e outro formado pelas placas com larguras menores que *237 pixels*. Isso nos levou a projetar 3 filtros. O primeiro baseado no tamanho médio de todas as amostras do conjunto de desenvolvimento, o segundo baseado no tamanho médio das amostras cujas placas têm largura maior que *237 pixels* e o terceiro baseado no tamanho médio das amostras cujas placas têm largura menor que *237 pixels*. Daí, chegamos aos seguintes resultados:

Filtro 1: (247,0 ± 12,0) x (43,0 ± 3,0) pixels

Filtro 2: (252,0 ± 6,0) x (43,0 ± 3,0) pixels

Filtro 3: (226,0 ± 6,0) x (43,0 ± 3,0) pixels

Baseado no fato de a função de correlação cruzada sugerir filtros com largura múltiplas de 4, escolhemos os seguintes tamanho de filtro:

Filtro 1: 248 x 43 pixels

Filtro 2: 252 x 43 pixels

Filtro 3: 228 x 43 pixels

Desta forma, chegamos aos filtros para placas mostrados abaixo:



Figura 4.11: Filtro 1, tamanho de 248 x 43 *pixels*.



Figura 4.12: Filtro 2, tamanho de 252 x 43 *pixels*.



Figura 4.13: Filtro 3, tamanho de 228 x 43 *pixels*.

4.2.2.2 Traço

Para os três filtros casados desenvolvidos a partir do traço que separa as letras e os números, chegamos aos seguintes resultados, respeitando a condição da largura do filtro ser múltipla de 4 (para que haja otimização do algoritmo de correlação cruzada):

Filtro 1: 16 x 43 pixels

Filtro 2: 16 x 19 pixels

Filtro 3: 16 x 19 pixels

Com esses resultados em mãos, é possível verificar que os filtros 2 e 3 tem o mesmo tamanho, e como as imagens são redimensionadas antes de serem computadas na construção do filtro casado, isso faz com que os filtros 2 e 3 sejam iguais. Daí, não há a necessidade de testar os dois filtros. Por isso, de agora em diante, não levaremos em consideração o filtro 3 do traço.

Desta forma, chegamos aos filtros mostrados abaixo:



Figura 4.14: Filtro 1.



Figura 4.15: Filtro 2.

4.2.2.3 Caracteres

As dimensões escolhidas para cada um dos filtros foram apresentadas na tabela 4.1.

Abaixo são mostrados os filtros obtidos:

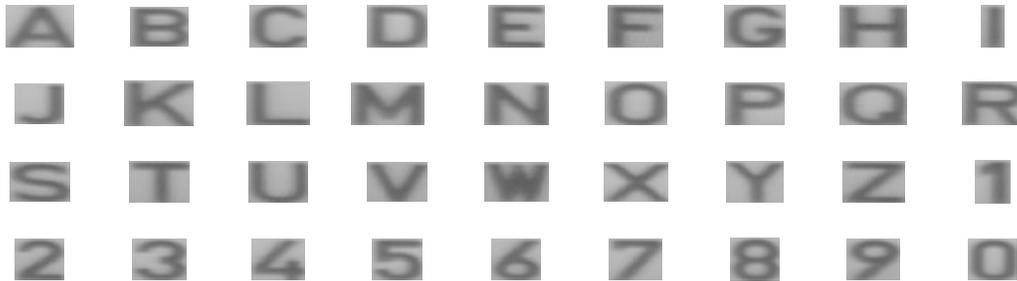


Figura 4.16: Filtro baseado nos caracteres.

Aqui é interessante notar que a flutuação entre as amostras de uma mesma classe é pequena, dado que os caracteres nos filtros estão bem nítidos e de imediata identificação, ao contrário do que houve com os filtros baseados na placa e no traço.

Tendo em mãos os filtros a serem testados, juntamente com as rotinas de pré-processamento, podemos passar para o próximo capítulo, onde discutiremos a aplicação dos filtros casados nas imagens.

Capítulo 5

Aplicação do Filtro Casado

No capítulo anterior, tratamos da análise de padrões e construção dos filtros casados propostos para este trabalho. Agora, antes de passarmos para a análise dos resultados obtidos e as eficiências de localização alcançadas, ainda é preciso mostrar como os filtros casados projetados são aplicados nas imagens.

De modo simplificado, o filtro varrerá a imagem e guardará os valores máximos de correlação de cada região filtrada. Ao final, a região que possuir o maior valor de correlação será a eleita como a de maior probabilidade de ocorrência da placa na imagem.

Todavia, antes de aplicarmos os filtros nas imagens de teste, devemos cuidar para que regiões de maior energia não sejam favorecidas durante o cálculo da correlação. Por isso, neste capítulo, iniciaremos nossa discussão tratando da normalização de energia.

5.1 Normalização de energia

O processo de aplicação de filtros casados consiste em calcular as correlações cruzadas entre os sinais e o filtro casado. A correlação cruzada de dois processos randômicos $X(t)$ e $Y(t)$, com funções de auto-correlação $R_X(t,u)$ e $R_Y(t,u)$, respectivamente, é definida por:

$$R_{XY}(t, u) = E[X(t)Y(u)] \quad (5.1)$$

e

$$R_{YX}(t, u) = E[Y(t)X(u)] \quad (5.2)$$

onde t e u representam dois valores de tempo em que o processo é observado [14]. Ou de forma discreta:

$$R_{xy}(m) = \sum_{n=0}^{N-m-1} x_{n+m} y_n^* \quad (5.3)$$

para $n \geq 0$;

Desta forma, é possível verificar que a correlação cruzada nada mais é do que o somatório de produtos. Por isso, sinais com maior nível de energia geram correlações com maiores níveis de energia também. Se o filtro casado for utilizado desta maneira, a maior correlação cruzada será referente ao sinal com maior energia.

O exemplo abaixo ilustra esse problema:

Dado um filtro $f(n) = [1 \ 2 \ 3]$ e dois sinais $x_1(n) = [1 \ 2 \ 3]$ e $x_2(n) = [10 \ 10 \ 10]$, temos que as respectivas correlações cruzadas entre o filtro e os sinais são:

$$R_{fx1} = [3 \ 8 \ 14 \ 8 \ 3]$$

e

$$R_{fx2} = [10 \ 50 \ 60 \ 30 \ 30]$$

Os valores máximos das correlações cruzadas calculadas são 14 e 60. Não há dúvidas que o sinal x_1 “casa” melhor com o filtro, uma vez que tanto x_1 , quanto f representam o mesmo sinal. Todavia, x_2 , por possuir maior energia, acaba gerando correlações maiores.

Para evitar este tipo de problema, é prática comum normalizar as energias dos sinais que serão casados. Neste trabalho serão propostas duas normalizações distintas: a clássica, onde a saída do filtro é normalizada pela energia do sinal que gerou o filtro (com o intuito de maximizar a relação sinal/ruído) e uma alternativa, onde a normalização de energia consiste em calcular a energia de cada sinal de entrada e dividir este sinal, ainda na entrada do filtro, pela raiz quadrada de sua energia, conforme indicam as equações abaixo:

$$E = \frac{\sum_{n=1}^N x(n)^2}{N} \quad (5.4)$$

onde: N é o número de amostras do sinal;

$$x_{Norm}(n) = x(n) / \sqrt{E} \quad (5.5)$$

Essa normalização faz com que todos os sinais tenham energia unitária. Os filtros, apesar de não precisarem ser normalizados em energia, uma vez que são comuns a todas as operações de correlação aplicadas nos sinais, neste trabalho, sofreram a normalização.

Desta forma, se refizermos os cálculos para o exemplo acima, sem que o filtro seja normalizado¹⁵, teremos os seguintes sinais normalizados em energia:

$$x_{1\text{Norm}}(n) = [0,463 \ 0,926 \ 1,389]$$

e

$$x_{2\text{Norm}}(n) = [1 \ 1 \ 1]$$

E como resposta ao filtro:

$$R_{fx1} = [1,389 \ 3,704 \ 6,482 \ 3,704 \ 1,389]$$

e

$$R_{fx2} = [1 \ 5 \ 6 \ 3 \ 3]$$

Agora, os valores máximos das respectivas correlações cruzadas são 6,482 e 6, demonstrando que, quando ambos os sinais estão normalizados em energia, x_1 “casa” melhor com o filtro.

Podemos estender esses cálculos para o caso bidimensional (2D). Desta forma, para aplicações em imagens teremos que:

$$E = \frac{\sum_{i=1}^N \sum_{j=1}^M [I(i, j)]^2}{N * M} \quad (5.6)$$

onde: a imagem I tem resolução $N \times M$ *pixels*;

¹⁵ Caso o filtro seja normalizado neste ponto, para fins de comparação de resultados, o mesmo filtro deveria ser normalizado no exemplo anterior também.

$$I_{Norm} = I / \sqrt{E} \quad (5.7)$$

A análise de normalização de energia mostrada no exemplo acima acaba nos levando a uma outra conclusão esperada, e bastante importante para a simplificação da complexidade do algoritmo de filtro casado desenvolvido. Para sinais com tamanhos iguais ou maiores ao do filtro, os valores máximos de correlação ocorrem justamente quando o filtro está “contido completamente” no sinal, isto é, quando todos os pontos do filtro estão sendo usados no cálculo da correlação.

Com isso, não se faz necessário o cálculo da correlação para todos os valores possíveis, fazendo com que o cálculo da correlação seja simplificado ao cálculo de produtos escalares para o caso onde o filtro está “contido completamente” no sinal, justamente nos pontos que ocorrem os maiores valores de correlação.

Com essa conclusão, podemos passar para o algoritmo de varredura do filtro na imagem.

5.2 Algoritmo de varredura do filtro casado

O cálculo da correlação entre um sinal de tamanho n e um filtro de tamanho m gera como resposta um sinal de tamanho $n + m - 1$. No exemplo acima tínhamos que $n = m = 3$. O sinal resultante tinha tamanho 5, conforme a relação aqui mostrada. Além disso, foi verificado que o valor máximo de correlação ocorreu quando os três pontos

do filtro estavam “completamente contidos” no sinal. Tirando proveito desse resultado, a filtragem das imagens foi simplificada.

Normalmente, um filtro de tamanho $N \times M$ varreria uma imagem conforme mostra a figura a seguir:

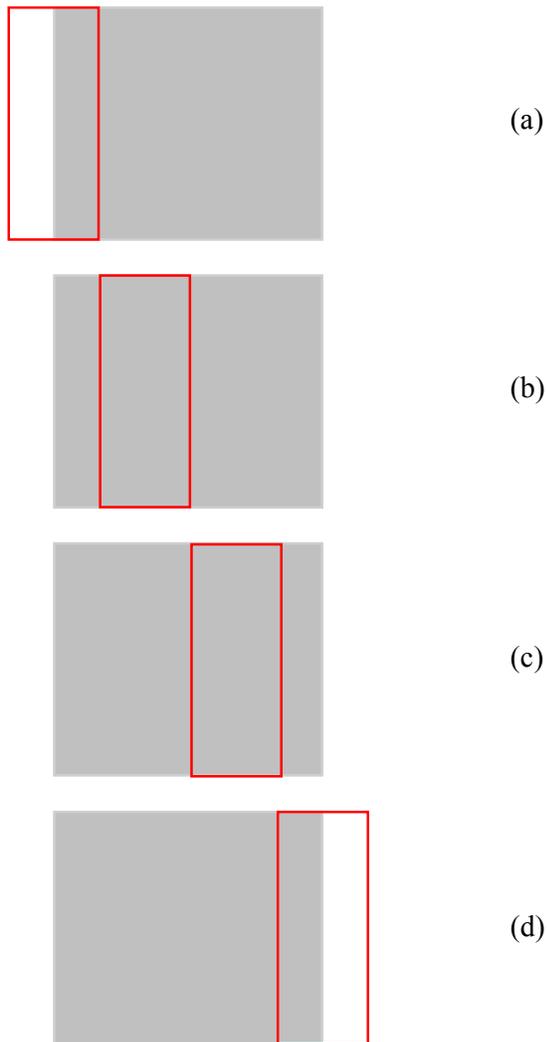


Figura 5.1: Exemplo de filtro de tamanho $N \times M$ varrendo uma imagem de tamanho $P \times Q$ com um passo maior que 1, onde $P > N$ e $Q = M$.

Na figura 5.1, os valores de correlação são maiores em (b) e (c), uma vez que em (a) e (d) o filtro não está “completamente contido” na imagem (ainda mais se garantirmos que o sinal a ser casado está completamente dentro da imagem. No nosso caso, garantimos que a placa está completa em todas as imagens). Calculando a correlação apenas para esses casos, o sinal de saída terá o tamanho de $P-N+1$.

Neste projeto, o filtro só é utilizado a partir do ponto em que esteja “completamente contido” na imagem. Com isso, para um filtro de tamanho $N \times M$ e uma imagem de tamanho $P \times Q$, onde $P > N$ e $Q > M$, temos um total de correlações igual a $(P-N+1) \times (Q-N+1)$.

Após cada correlação, o filtro sofre um deslocamento de 1 *pixel* (passo de tamanho 1) horizontalmente, até o último ponto em que esteja “completamente contido” na imagem. Ao final da varredura desta faixa horizontal, o filtro volta para o ponto horizontal inicial e é deslocado verticalmente de 1 *pixel* e começa a varrer a nova faixa horizontal.

Esse procedimento se repete até que o filtro atinja o último ponto da imagem em que esteja “completamente contido”, no canto inferior direito. Para cada correlação calculada, o valor encontrado é guardado em um índice de uma matriz. Desta forma, o primeiro valor calculado na primeira faixa é guardado no ponto (1,1) da matriz de resultados, o segundo valor no índice (1,2) e assim até o final da linha. O primeiro valor da segunda linha é guardado em (2,1), o segundo (2,2) e assim até o final da imagem, sempre repetindo esse raciocínio. A figura 5.2 ilustra melhor esse algoritmo.

Tendo em mãos os filtros a serem testados, juntamente com as rotinas de pré-processamento, além do método de aplicação dos filtros nas imagens, podemos passar para o próximo capítulo, onde discutiremos os resultados e eficiências de localização do sistema proposto.

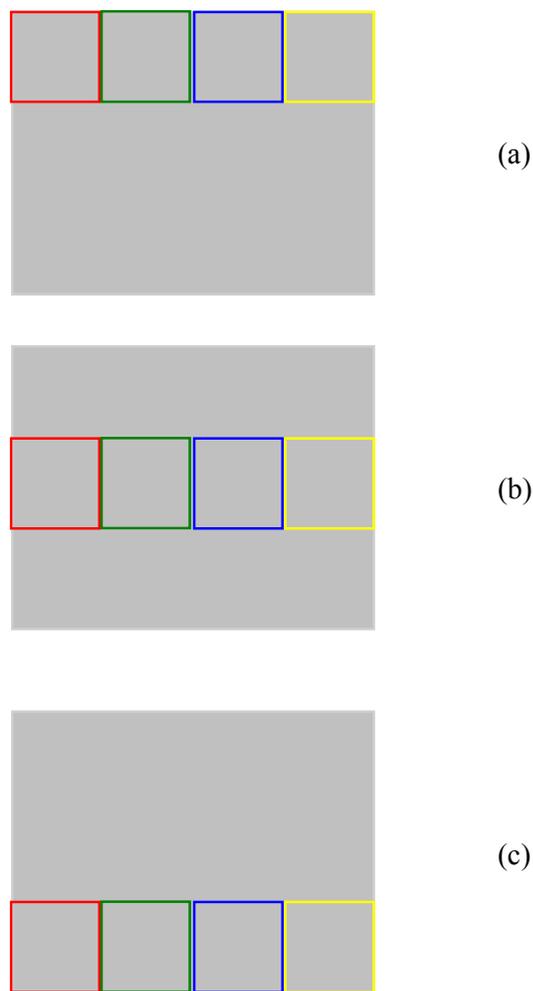


Figura 5.2: Exemplo de filtro de tamanho $N \times M$ varrendo uma imagem de tamanho $P \times Q$ com passos maior que 1 *pixel*, onde $P > N$ e $Q > M$. Cada quadrado de uma linha representa um passo horizontal (deslocamento de 1 *pixel*).

Capítulo 6

Resultados

Nos capítulos anteriores, foram propostos alguns filtros casados determinísticos, baseados nos caracteres da placa, no traço que separa letras e algarismos da placa e os caracteres contidos na placa. Neste capítulo, serão analisadas as eficiências de localização destes filtros. Desta forma, o objetivo deste capítulo é apresentar os resultados obtidos por diferentes enfoques de aplicação do método de pré-processamento e com os filtros desenvolvidos, buscando relacioná-los com suas respectivas eficiências.

Para automatizar o estudo de eficiência, encontra-se primeiramente o padrão (placa, traço ou caractere) segundo a aplicação do filtro casado sob teste. O resultado de saída de cada imagem processada (a região de maior correlação) é salvo e as coordenadas obtidas são comparadas com as coordenadas da placa da mesma imagem, que estão contidas na base de dados. Desta forma, após aplicar o filtro em todas as imagens de teste, um relatório é produzido. Em seguida, a partir dos dados do relatório, calcula-se a eficiência de localização de cada filtro. Por fim, uma análise estatística dos acertos é realizada, através de medidas de distância entre os resultados do processamento e os dados da base.

6.1 Medição da eficiência

Para cada um dos tipos de filtros desenvolvidos, foi proposta uma análise de eficiência diferente. Isto é fácil de entender, uma vez que estamos trabalhando com filtros distintos, e principalmente, com informações e tamanhos diferentes. Há o filtro que procura pela placa inteira na imagem e compara o resultado com a informação da placa inteira na base de dados, enquanto que há outro que procura por caracteres e compara a saída com o posicionamento dos caracteres na base de dados e há ainda o filtro que tenta localizar o traço que separa as letras dos números. Na base de dados, o traço é localizado a partir de seu ponto central (como um elemento pontual) e não por suas informações de borda (como no caso das placas e dos caracteres).

Iniciaremos nossa discussão pela análise de eficiência do filtro relativo à placa inteira, depois passaremos para a análise do filtro de traço e por último analisaremos os filtros baseados nos caracteres.

6.1.1 Filtro baseado na placa inteira

Neste caso, como referência para a análise de acertos, utilizou-se a informação de posicionamento dos primeiro e último caracteres. Considerou-se um acerto quando o resultado obtido pelo filtro contivesse o ponto de início do primeiro caractere (CSX_1 e CSY_1 , ambos existentes na base de dados)¹⁶ e o ponto de fim do último caractere

¹⁶ CSX – Character Start X (coordenada X de início do caractere)

CSY – Character Start Y (coordenada Y de início do caractere)

CEX – Character End X (coordenada X de fim do caractere)

CEY – Character End Y (coordenada Y de fim do caractere)

(CEX_7 e CEY_7 , também existentes na base de dados). Daí, consideramos um acerto se as quatro condições abaixo forem satisfeitas:

$$CSX_1 \geq CXRS$$

$$CSY_1 \geq CYRS$$

$$CEX_7 \leq CXRE$$

$$CEY_7 \leq CYRE$$

onde:

CSX_1 – Coordenada X inicial do 1º caractere da base de dados

CSY_1 – Coordenada Y inicial do 1º caractere da base de dados

CSX_7 – Coordenada X final do 7º caractere da base de dados

CSY_7 – Coordenada Y final do 7º caractere da base de dados

e

$CXRS$ – Coordenada X inicial do resultado

$CYRS$ – Coordenada Y inicial do resultado

$CXRE$ – Coordenada X final do resultado

$CYRE$ – Coordenada Y final do resultado

De forma gráfica podemos representar um acerto da seguinte forma:

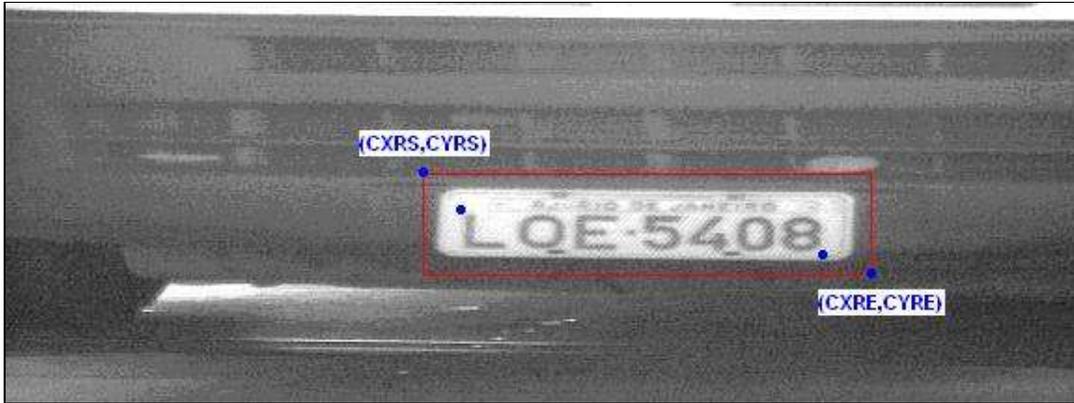


Figura 6.1: Os pontos na figura (da esquerda para a direita): (CXRS, CYRS), (CSX₁, CSY₁), (CEX₇, CEY₇) e (CXRE, CYRE).

A análise feita a partir destes pontos serve para nos dizer se houve um acerto ou um erro, mas não nos dá noção da “qualidade do acerto”, isto é, não nos diz o quanto o resultado se aproximou da resposta ideal (contida na base de dados), por isso, chamaremos essa análise de quantitativa.

A outra análise, a qualitativa, que conforme explicado acima, mostra o quanto o resultado do filtro se aproximou do ideal, será dado por:

$$d_{XS} = CSX_{Placa} - CXRS$$

$$d_{YS} = CSY_{Placa} - CYRS$$

$$d_{XE} = CEX_{Placa} - CXRE$$

$$d_{YE} = CEY_{Placa} - CYRE$$

Desta forma, a análise de eficiência do filtro em questão será feita a partir destas duas análises. Vale lembrar que a análise qualitativa é apenas feita nas respostas que produziram um acerto.

Resumindo:

Análise quantitativa – define se houve a localização correta ou não da placa

Análise qualitativa – se houve localização correta da placa, define o quão próximo do ideal esse acerto se encontra.

6.1.2 Filtro baseado no traço

A análise de eficiência do filtro baseado no traço, conforme dito anteriormente, difere da análise do filtro da placa, uma vez que na base de dados, o traço é representado pela coordenada de seu centro. Por isso, as formulações para a análise quantitativa e para a análise qualitativa não são as mesmas. No caso da análise quantitativa, temos que, para que haja um acerto (equivalente a uma localização), as seguintes condições devem ser satisfeitas:

$$CXTraco \geq XRS$$

$$CYTraco \geq YRS$$

$$CXTraco \leq XRE$$

$$CYTraco \leq YRE$$

onde:

XRS - Coordenada X inicial do resultado

YRS - Coordenada Y inicial do resultado

XRE - Coordenada X final do resultado

YRE - Coordenada X final do resultado

e

CSXTraco – Coordenada X do traço contida na base de dados

CSYTraco – Coordenada Y do traço contida na base de dados

Lembrando que o resultado produzido pelo filtro é uma área e devemos comparar o resultado com a informação do traço na base de dados, que é representado por seu ponto central.

A análise qualitativa da resposta deste filtro é dada por:

$$d_x = CXTraco - CXResp$$

$$d_y = CYTraco - CYResp$$

onde:

$$CXResp = (XRS + XRE) / 2$$

$$CYResp = (YRS + YRE) / 2$$

As figuras a seguir ilustram melhor os valores de CXResp e CYResp:

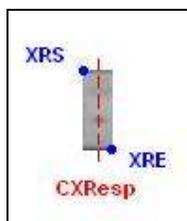


Figura 6.2: CXResp é o ponto médio entre XRS e XRE

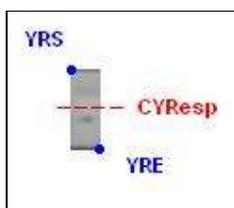


Figura 6.3: CYResp é o ponto médio entre YRS e YRE

Ou seja, a análise qualitativa compara a distância entre a informação pontual da base de dados com o ponto médio da resposta do filtro. Utilizando este enfoque, a medida de distância em y apresentará um erro maior, já que no filtro, o traço não se encontra no ponto médio do eixo y. Na verdade, no filtro, o traço se encontra um pouco deslocado para baixo.

6.1.3 Filtro baseado nos caracteres

Neste caso, da mesma forma que o filtro baseado na placa inteira, utilizou-se a informação de posicionamento dos caracteres na base de dados como referência para a análise de acertos. Considerou-se um acerto quando o resultado obtido pelo filtro contivesse o ponto de início do caractere (CSX e CSY, ambos existentes na base de

dados)¹⁷ e o ponto de fim do último caractere (CEX e CEY, também existentes na base de dados). Daí, consideramos um acerto se as quatro condições abaixo forem satisfeitas:

$$CSX \geq CXRS$$

$$CSY \geq CYRS$$

$$CEX \leq CXRE$$

$$CEY \leq CYRE$$

onde:

CSX – Coordenada X inicial do caractere na base de dados

CSY – Coordenada Y inicial do caractere na base de dados

CXRS – Coordenada X final do caractere na base de dados

CYRS – Coordenada Y final do caractere na base de dados

e

CXRS – Coordenada X inicial do resultado

CYRS – Coordenada Y inicial do resultado

CXRE – Coordenada X final do resultado

CYRE – Coordenada Y final do resultado

De forma gráfica podemos representar um acerto da seguinte forma:

¹⁷ CSX – Character Start X (coordenada X de início do caractere)
CSY – Character Start Y (coordenada Y de início do caractere)
CEX – Character End X (coordenada X de fim do caractere)
CEY – Character End Y (coordenada Y de fim do caractere)



Figura 6.4: A figura representa um exemplo de acerto na localização de caracteres.

No caso dos filtros baseados nos caracteres só há necessidade de se fazer o estudo quantitativo da eficiência dado que há somente um filtro proposto para cada caractere. O estudo qualitativo serviria apenas para uma comparação entre filtros que buscassem o mesmo sinal a ser casado.

6.2 Eficiências

Como temos três tipos de filtros distintos, um tipo baseado na placa, outro baseado no traço separador das letras e dos números e outro baseado nos caracteres, dividiremos a apresentação dos resultados em três seções, cada uma tratando de um filtro específico.

6.2.1 Filtro baseado no traço

Foram desenvolvidos dois filtros baseados no traço: um que tem a altura igual à da placa (Filtro 1) e outro com altura igual à dos caracteres (Filtro 2).

6.2.1.1 Filtro 1

A tabela 6.1 apresenta as eficiências alcançadas para um conjunto de parâmetros de pré-processamento¹⁸:

Tabela 6.1: Eficiências de localização do Filtro 1.

pré-process	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Não	-	-	-	-	-	-	0,082
Sim	0,03	0,06	18	3	35	60	0,4
Sim	0,03	0,06	19	2	35	50	0,048
Sim	0,03	0,06	20	3	35	60	0,4

6.2.1.2 Filtro 2

A tabela 6.2 apresenta as eficiências alcançadas para um conjunto de parâmetros de pré-processamento¹⁹:

Tabela 6.2: Eficiências de localização do Filtro 2.

pré-process	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Não	-	-	-	-	-	-	0,068
Sim	0,03	0,06	18	3	35	60	0,232
Sim	0,03	0,06	20	2	35	60	0,24

No caso dos dois filtros aqui apresentados, não há um tipo de erro específico ocorrendo. O filtro não tem como saída “pedaços” de caracteres como “T” ou “L”. O filtro retorna ruídos de fundo como resposta.

¹⁸ Os resultados da tabela apresentam incerteza de 0,008.

¹⁹ Os resultados da tabela apresentam incerteza de 0,008.

Os resultados aqui apresentados foram obtidos através da normalização proposta neste trabalho, isto é, normalizando a energia de cada sinal antes deste ser casado com o filtro. A baixa eficiência alcançada desmotivou qualquer tentativa de se utilizar a normalização de energia clássica, dada a aproximação determinística do problema.

Por apresentar uma eficiência tão baixa, não se faz necessário a análise qualitativa dos filtros. Podemos, assim, passar para a apresentação dos resultados obtidos com os filtros baseados na placa.

6.2.2 Filtro baseado na placa

Os filtros baseados nas placas carregam consigo maior quantidade de informação, uma vez que possuem as informações dos caracteres e do fundo da placa, daí se esperar que possam atingir uma eficiência superior às alcançadas pelos filtros baseados no traço. Todavia, por serem maiores, são mais custosos computacionalmente e, conseqüentemente, são mais lentos no processamento.

Serão feitas as análises dos três filtros projetados: o filtro baseado no tamanho médio de todas as placas que compõem a base de desenvolvimento (Filtro 1), aquele baseado no tamanho das placas maiores que compõem a base (Filtro 2) e por último o filtro baseado no tamanho das placas menores (Filtro 3).

6.2.2.1 Filtro 1

A tabela 6.3 apresenta as eficiências alcançadas para um conjunto de parâmetros de pré-processamento²⁰, usando a normalização de energia na entrada do filtro (proposta neste trabalho):

Tabela 6.3: Eficiências de localização do Filtro 1.

pré-proces	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Não	-	-	-	-	-	-	0,552
Sim	0,02	0,08	20	3	30	50	0,552
Sim	0,02	0,08	20	2	30	50	0,608
Sim	0,02	0,08	23	2	30	50	0,44
Sim	0,02	0,08	23	3	30	50	0,312
Sim	0,02	0,08	23	4	30	50	0,264
Sim	0,03	0,05	20	2	30	50	0,632
Sim	0,03	0,05	20	3	30	50	0,608
Sim	0,03	0,06	17	2	30	50	0,672
Sim	0,03	0,06	18	2	25	50	0,688
Sim	0,03	0,06	18	2	30	50	0,712
Sim	0,03	0,06	18	2	30	60	0,792
Sim	0,03	0,06	18	2	35	50	0,704
Sim	0,03	0,06	18	2	35	60	0,800
Sim	0,03	0,06	18	2	35	65	0,768
Sim	0,03	0,06	18	3	35	60	0,808
Sim	0,03	0,06	18	4	35	60	0,800
Sim	0,03	0,06	18	3	30	50	0,68
Sim	0,03	0,06	19	2	30	50	0,616
Sim	0,03	0,06	20	1	30	50	0,584
Sim	0,03	0,06	20	2	30	50	0,664
Sim	0,03	0,06	20	3	30	50	0,6
Sim	0,03	0,07	20	1	30	50	0,528
Sim	0,03	0,07	20	2	30	50	0,648
Sim	0,03	0,07	20	3	30	50	0,576

Neste primeiro filtro, conseguimos alcançar uma eficiência de 80,8%, resultado bem acima dos apresentados pelos filtros baseados no traço. Com os resultados aqui

²⁰ Os resultados da tabela apresentam incerteza de 0,008.

obtidos, podemos descartar aqueles que apresentam eficiências mais baixas nos testes dos próximos filtros.

6.2.2.2 Filtro 2

A tabela 6.4 apresenta as eficiências alcançadas para um conjunto de parâmetros de pré-processamento²¹, usando a normalização de energia na entrada do filtro (proposta neste trabalho):

Tabela 6.4: Eficiências de localização do Filtro 2.

pré-process	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Não	-	-	-	-	-	-	0,644
Sim	0,03	0,06	18	2	30	55	0,816
Sim	0,03	0,06	18	2	35	60	0,816
Sim	0,03	0,06	18	2	35	65	0,792
Sim	0,03	0,06	18	3	35	60	0,808
Sim	0,03	0,06	18	4	35	60	0,800
Sim	0,03	0,06	18	2	35	65	0,800
Sim	0,03	0,06	18	3	35	60	0,800

Com o Filtro 2, foram atingidas eficiências de 81,6%, superando as eficiências alcançadas pelo Filtro 1. Vale notar nos dois casos como o pré-processamento tem um papel importante, fazendo eficiência aumentar notadamente.

Para efeitos comparativos, aplicamos a normalização de energia clássica ao filtro 2, com o conjunto de entrada pré-processamento correspondente à 2ª linha da tabela 6.4. Neste caso, a eficiência alcançada foi de 41,6%. A aproximação determinística utilizada para o problema não possibilita a utilização desta normalização, dado a

²¹ Os resultados da tabela apresentam incerteza de 0,008.

grande flutuação entre as amostras usadas para o desenvolvimento do filtro, o que distancia o problema de uma abordagem determinística clássica.

6.2.2.3 Filtro 3

A tabela 6.5 apresenta as eficiências alcançadas para um conjunto de parâmetros de pré-processamento²², usando a normalização de energia na entrada do filtro (proposta neste trabalho):

Tabela 6.5: Eficiências de localização do Filtro 3.

pré-process	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Não	-	-	-	-	-	-	0,220
Sim	0,03	0,06	18	3	35	60	0,296

O Filtro 3 apresentou uma eficiência muito abaixo dos anteriores, principalmente porque no redimensionamento durante sua construção há uma perda de informação inerente a essa operação, uma vez que a maioria das placas utilizadas no desenvolvimento do filtro foram redimensionadas para ficar com tamanhos menores que os originais.

Desta forma, fica claro que os filtros que apresentam melhores resultados são aqueles que foram construídos a partir do tamanho médio das placas grandes (Filtro 2), apresentando eficiências de até 81,6%. A análise qualitativa, apresentada anteriormente, ajudará a escolher aquele que possui resposta mais próxima da ideal.

²² Os resultados da tabela apresentam incerteza de 0,008.

6.2.3 Filtro baseado nos caracteres

Os filtros baseados nos caracteres foram testados de uma forma um pouco diferente dos demais. Ao invés de varrer a imagem com todos os 36 filtros, a imagem era varrida apenas pelos filtros correspondentes aos caracteres da placa. Exemplificando: para uma placa LJD8430, apenas os filtros baseados em “L”, “J”, “D”, “8”, “4”, “3” e “0” eram usados.

Além disso, foi tomado cuidado com placas que possuíam caracteres repetidos (como JXZ3220). Uma vez encontrado um caractere por um filtro, o mesmo filtro não teria como resposta a mesma posição, uma vez que não mais procuraria na mesma região.

A tabela 6.6 mostra a eficiência de localização para esses filtros em imagens sem a etapa de pré-processamento.

Tabela 6.6: Eficiências de localização dos filtros de caracteres. Alguns caracteres não ocorreram durante os testes, por isso não há eficiências associadas a eles.

A	B	C	D	E	F	G	H	I
66,7%	77,8%	88,9%	0%	100%	-	66.7%	100%	100%
J	K	L	M	N	O	P	Q	R
100%	86,7%	85,2%	80,0%	58,3%	60%	100%	100%	75%
S	T	U	V	W	X	Y	Z	1
-	100%	-	100%	100%	100%	100%	33,3%	86,7%
2	3	4	5	6	7	8	9	0
73,4%	89,4%	100%	90,9%	90,9%	100%	71,4%	93,3%	86,7%

Além disso, outra medida de eficiência foi feita. Foi contado o número de placas em que todos os caracteres foram encontrados. Para este caso, foi atingida uma eficiência de 67,4% das placas, inferior às eficiências atingidas pelos filtros baseados na placa inteira.

É interessante notar que os erros que produzidos na maioria das vezes foram causados por ruídos no fundo da imagem e não por caracteres da placa, isto é, o filtro não escolheu um outro caractere qualquer em detrimento ao caractere correto, salvo poucas exceções.

6.2.4 Escolha do melhor filtro

Baseado na análise qualitativa, escolheremos o filtro de melhor desempenho. A análise será feita sobre o Filtro 2 (filtro baseado na placa inteira), com os parâmetros de

entrada de pré-processamento que produziram maior eficiência²³, os quais denominaremos de Filtro2.1 e Filtro 2.2:

Tabela 6.7: Filtros de maior eficiência de localização.

Filtro	ll	ul	T	numResp	distResp	tamRecorte	eficiência
Filtro 2.1	0,03	0,06	18	2	30	55	0,816
Filtro 2.2	0,03	0,06	18	2	35	60	0,816

A distribuição da distância dos acertos (análise qualitativa) dos filtros da tabela 6.7 pode ser vista nas figuras 6.5, 6.6, 6.7 e 6.8, para cada coordenada:

²³ Essa análise poderia ser feita a partir de até 6 conjuntos de parâmetros de entrada, uma vez que essa é a quantidade de conjuntos que apresentaram eficiências dentro da mesma margem de incerteza dos resultados.

(a) Filtro 2.1

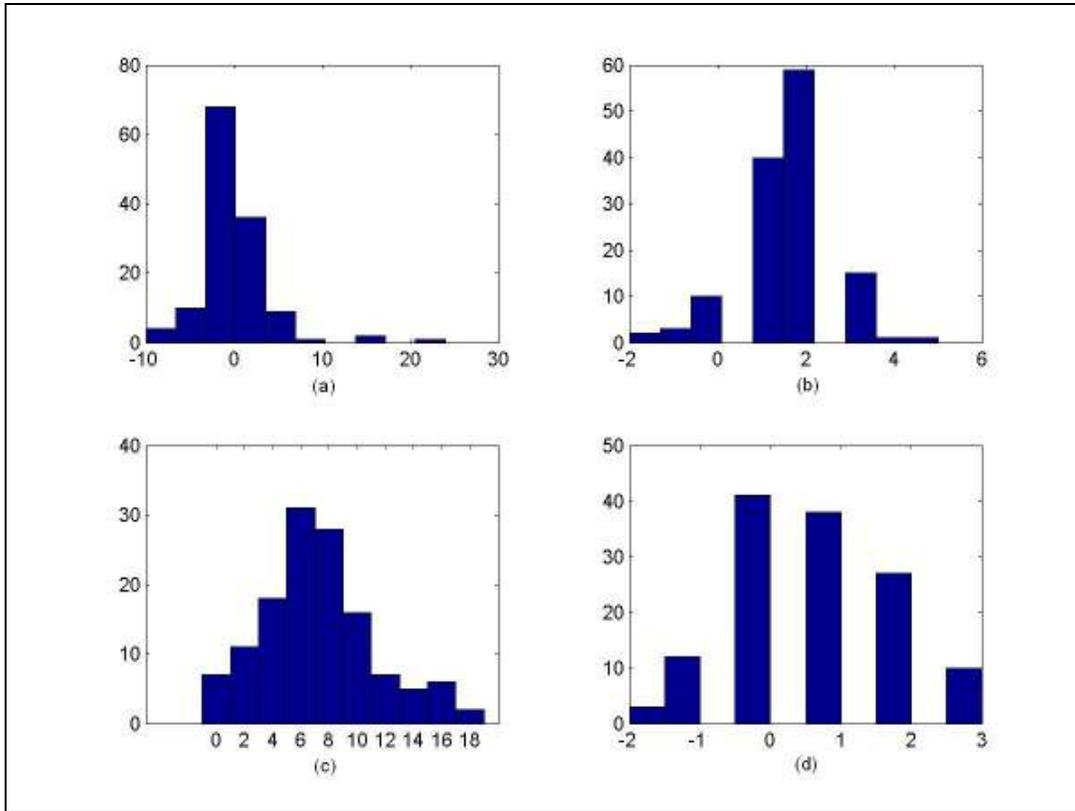


Figura 6.5: (a) $d_{XS} = CSXPlaca - CXRS$; (b) $d_{YS} = CSYPlaca - CYRS$; (c) $d_{XE} = CEXPlaca - CXRE$;
(d) $d_{YE} = CEYPlaca - CYRE$.

(a) $d_{XS} = 0 \pm 4$ pixels

(b) $d_{YS} = 2 \pm 1$ pixels

(c) $d_{XE} = 8 \pm 4$ pixels

(d) $d_{YE} = 1 \pm 1$ pixels

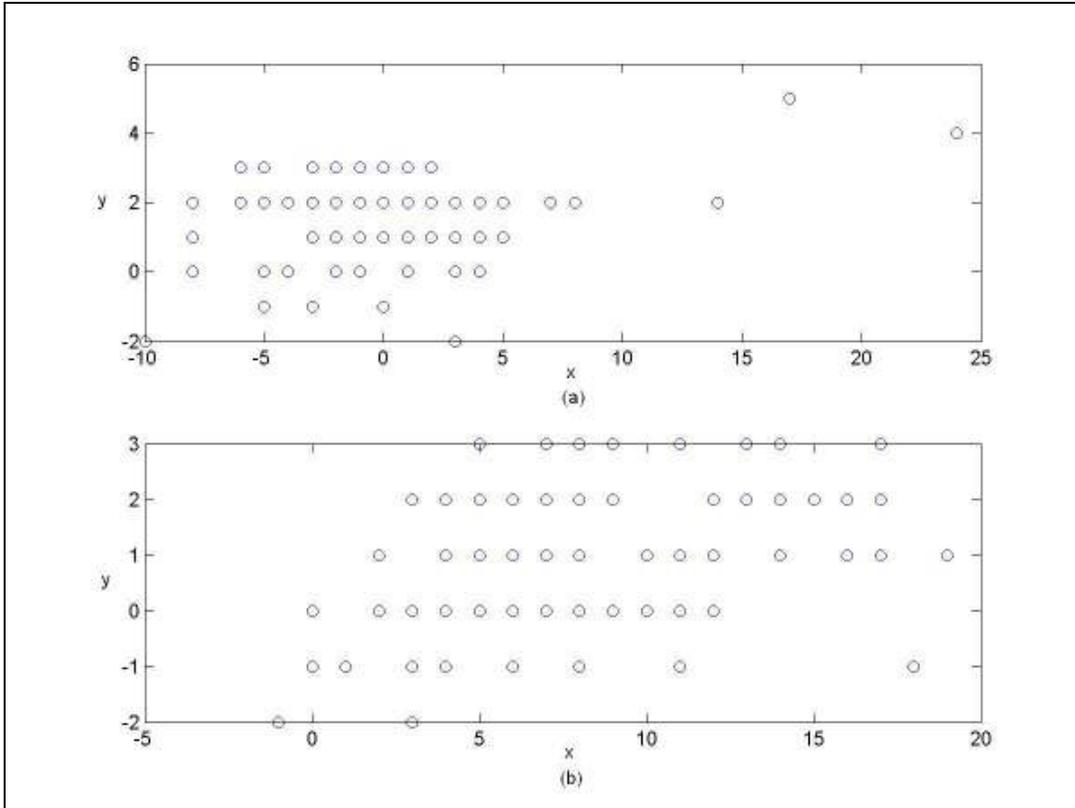


Figura 6.6: (a) Indica o quanto o ponto inicial (CXRS,CYRS) se distanciou da posição (ideal) contida na base de dados (CSXPlaca, CSYPlaca); (b) Indica o quanto o ponto final (CXRE,CYRE) se distanciou da posição (ideal) contida na base de dados (CEXPlaca, CEYPlaca).

(b) Filtro 2.2

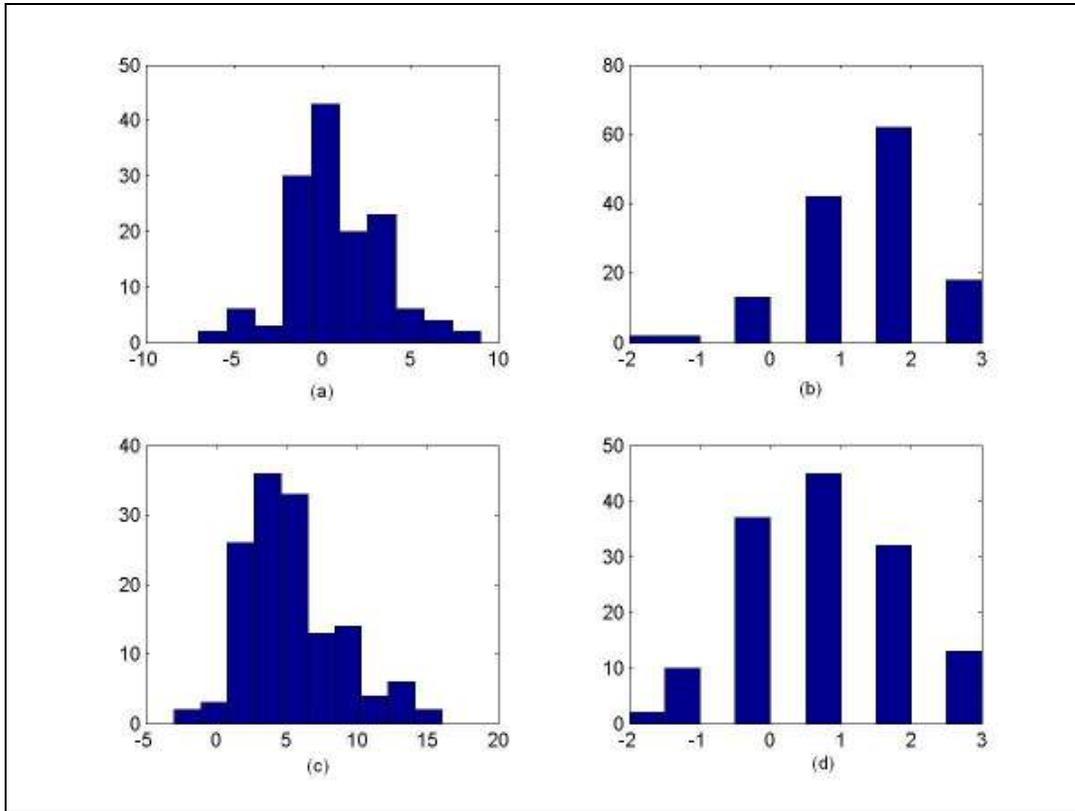


Figura 6.7: (a) $d_{XS} = CSXPlaca - CXRS$; (b) $d_{YS} = CSYPlaca - CYRS$; (c) $d_{XE} = CEXPlaca - CXRE$;
(d) $d_{YE} = CEYPlaca - CYRE$.

(a) $d_{XS} = 1 \pm 3$ pixels

(b) $d_{YS} = 2 \pm 1$ pixels

(c) $d_{XE} = 5 \pm 4$ pixels

(d) $d_{YE} = 1 \pm 1$ pixels

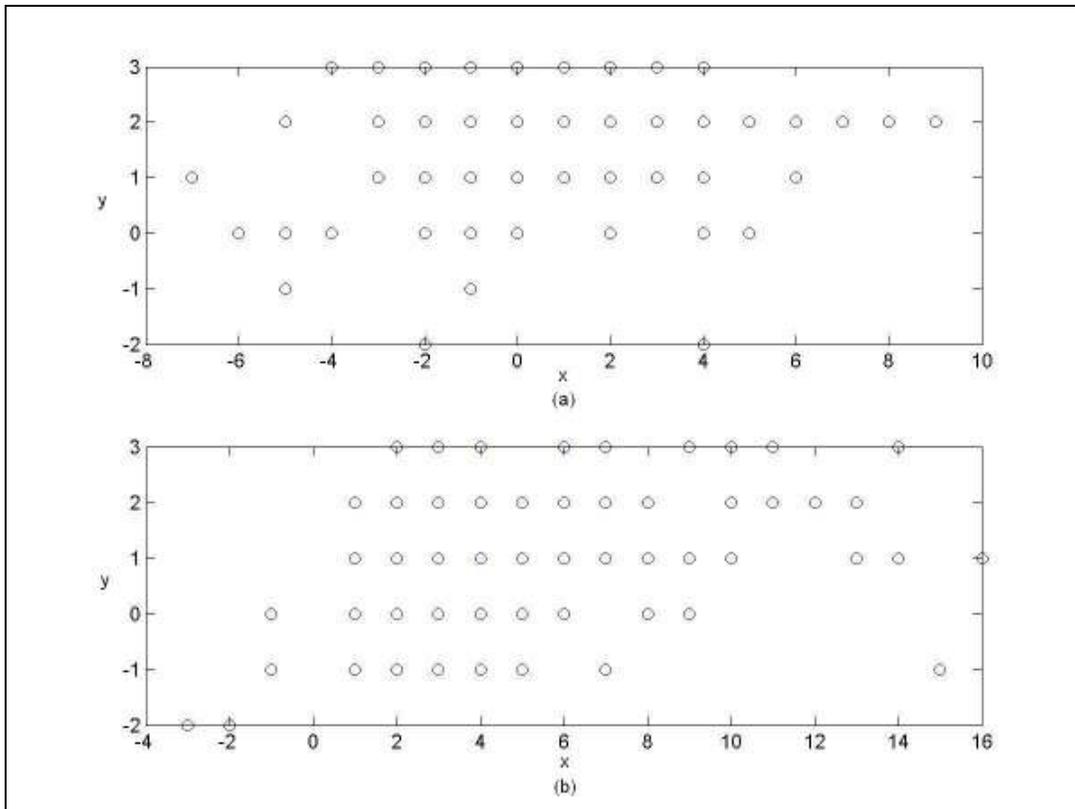


Figura 6.8: (a) Indica o quanto o ponto inicial (CXRS,CYRS) se distanciou da posição (ideal) contida na base de dados (CSXPlaca, CSYPlaca); (b) Indica o quanto o ponto final (CXRE,CYRE) se distanciou da posição (ideal) contida na base de dados (CEXPlaca, CEYPlaca).

A partir dos resultados obtidos, fica difícil escolher o melhor conjunto de parâmetros de entrada do pré-processamento, uma vez que a análise qualitativa ficou semelhante para ambas as análises. Por isso, escolheremos ambos os conjuntos de parâmetros como os melhores conjuntos de entrada do pré-processamento, utilizando o filtro baseado na placa do tipo 2.

Ao final dos cálculos de eficiência cabe uma observação: a eficiência da etapa dos filtros casados, aplicados sobre as imagens pré-processadas pode ser considerada alta. O pré-processamento utilizado pelo Filtro 2.2 tem uma eficiência de 84%. Sabendo

que a eficiência do sistema após o processamento do Filtro 2.2 é de 81,6%, é possível se concluir que a eficiência do Filtro 2.2, exclusivamente sobre as imagens pré-processadas é de 97,1%. Daí, é fácil notar a importância do pré-processamento, uma vez que a eficiência do Filtro 2, sem pré-processamento, é de 64,4%.

6.3 Limiares de decisão

O sistema proposto neste trabalho decide por uma área da imagem e assume que a área escolhida contém a placa. Todavia não há qualquer mecanismo inteligente que confirme a decisão do sistema. Pensando nisso, foi feito um estudo para estabelecer limiares de decisão para os filtros, isto é, baseado no valor da resposta do filtro seria possível saber se aquela resposta corresponderia à placa ou não. Essa realimentação poderia trabalhar como um pós-processamento.

O estudo aqui desenvolvido consiste em medir o valor de respostas corretas e respostas erradas de alguns filtros propostos e em seguida tentar traçar um limiar de decisão entre respostas corretas e respostas erradas. Iniciaremos a discussão com o filtro do tipo 2 baseado na placa, que alcançou a maior eficiência, e em seguida passaremos para os filtros dos caracteres.

6.3.1 Filtro baseado na placa

Para esse filtro, usaremos os valores de correlação obtidos pelo Filtro 2 para algumas imagens. Neste caso, o filtro varre as imagens por completo. As figuras 6.9 e 6.10

mostram os gráficos de distribuição das respostas para áreas que contêm a placa e para áreas que não contêm a placa respectivamente:

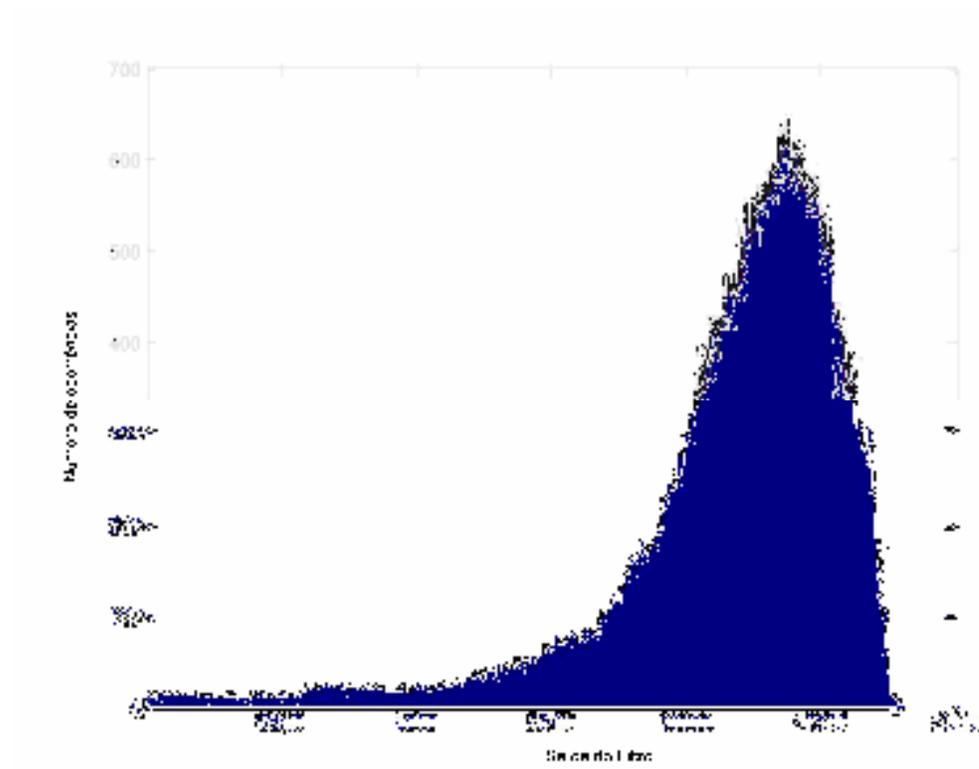


Figura 6.9: Resposta da correlação entre o filtro e áreas da imagem que contêm a placa.

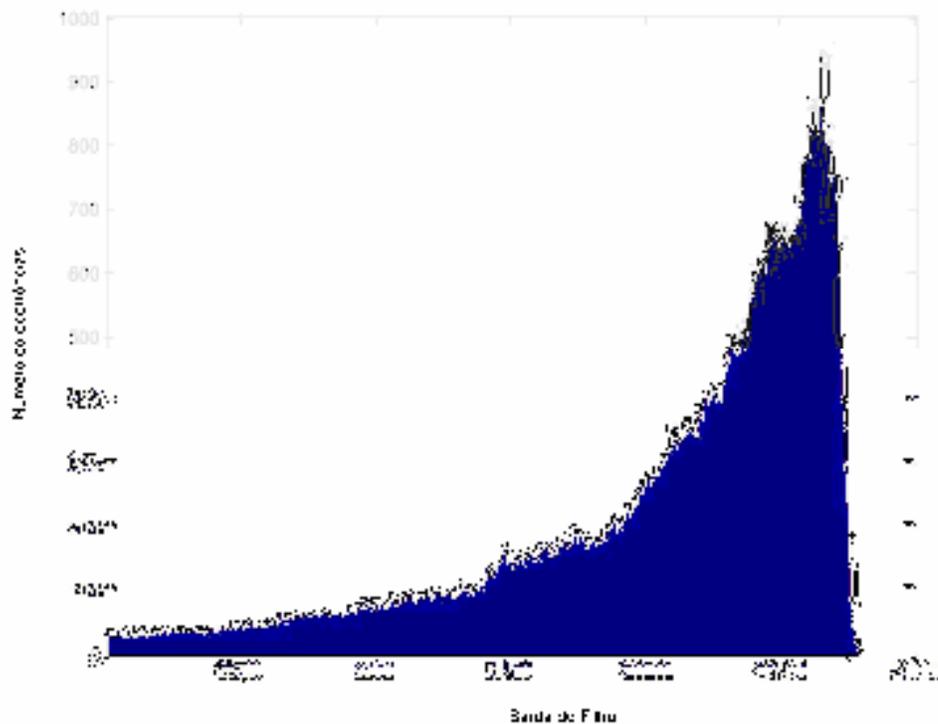


Figura 6.10: Resposta da correlação entre o filtro e áreas da imagem que não contêm a placa (ruído).

Através desses gráficos é possível verificar que não é possível estabelecer um limiar de decisão para a resposta da placa, dado que ambas as distribuições estão sobrepostas.

6.3.2 Filtro baseado nos caracteres

No caso dos caracteres, o método utilizado para o levantamento de um limiar de decisão foi um pouco diferente do utilizado para a placa. Ao invés de varrer as imagens com os 36 filtros, optou-se por varrê-las apenas com os filtros cujos caracteres pertenciam às placas e guardar os maiores valores de correlação de cada filtro (resposta do sistema) e classificá-los como acerto ou erro. Desta forma,

chegamos à tabela 6.8, onde são mostrados os valores médios e desvio padrão de correlação no caso de acerto para cada um dos filtros, bem como no caso de erros.

Tabela 6.8: Valores médios e desvios padrões da resposta a acertos e erros dos filtros de caracteres.

	A	B	C	D	E	F
Acerto	788,0±1,5	647,8±0,8	657,1±0,7	-	655,7±1,6	-
Erro	782±0	644,0±0	654±0	696,0±0,5	-	-
	G	H	I	J	K	L
Acerto	700,0±0,5	787,0±0,5	263,5±0,5	543,5±0,5	882,6±1,7	778,9±1,3
Erro	695,5±0,5	-	-	-	879,0±1,0	777,0±0,5
	M	N	O	P	Q	R
Acerto	854,0±1,0	745,6±1,0	751,5±1,0	679,2±0,3	786±0	753,7±1,1
Erro	849±0	738,8±0,4	749,0±1,0	-	-	749,5±0,0
	S	T	U	V	W	X
Acerto	-	702,0±0,5	-	668,5±2,5	709,5±3,0	711,5±0,5
Erro	-	-	-	-	-	-
	Y	Z	1	2	3	4
Acerto	658,0±0,0	772±0	435,3±0,5	569,3±1,0	633,9±1,2	613,0±1,3
Erro	-	717,5±1,0	433,5±1,5	566,0±1,0	630,0±0,0	-
	5	6	7	8	9	0
Acerto	590,7±0,9	569,2±0,9	613,6±0,4	593,0±1,2	612,6±1,0	567,7±1,3
Erro	587±0	566,0±0,0	-	591,5±1,5	609±0	564,5±0,0

Pela tabela acima já é possível estabelecer limiares para alguns filtros de caracteres, como no caso do “N”, onde os valores médios de acerto e de erro estão bem separados. Com isso, é possível concluir que os filtros de caracteres, se aplicados à área de resposta do filtro baseado na placa pode servir como validação do sistema e servi-lo como realimentação, um pós-processamento.

Capítulo 7

Conclusão

Neste trabalho, apresentamos um sistema de localização de placas de veículos particulares brasileiros baseado em análises freqüenciais e filtros casados. Vimos que o sistema de localização é constituído de três partes: aquisição da imagem do veículo, criação da base de dados e localização da placa na imagem e o respectivo recorte. Neste estudo foi utilizado, consideravelmente, procedimentos estatísticos que serviram para dar uma base sólida ao trabalho realizado.

O algoritmo de localização de placas de veículos particulares brasileiros é apenas uma das etapas do sistema de reconhecimento de placas de veículos particulares brasileiros.

Não foram usadas as técnicas de processamento de imagens clássicas, como: erosão, dilatação, binarização, entre outras. Ao invés disso, foram utilizadas técnicas geralmente aplicadas no processamento de sinais 1D clássico, como filtragem freqüencial e filtros casados.

7.1 O Banco de dados

Foi especificado um banco de dados, formado por imagens reais de veículos em trânsito, com a finalidade de desenvolver e avaliar o sistema de localização. Este banco foi dividido em duas partes: amostras de desenvolvimento (90%) e amostras de teste

(10%). Nele foram inseridas informações relevantes sobre a imagem e sobre a placa, como a categoria da imagem e da placa, bem como as coordenadas de interesse para esse estudo.

Aproximadamente 25.000 imagens foram obtidas na praça de pedágio da Ponte Rio-Niterói, das quais 5.000 haviam sido cadastradas, dentre elas, 1250 que atendiam o objeto de estudo deste trabalho. As imagens foram adquiridas tentando respeitar a resolução mínima apresentada em [2], ponto de partida deste projeto.

A utilização de uma base de dados reais torna o resultado alcançado bastante expressivo face o estado da arte no mundo.

7.2 Pré-processamento

O sistema de pré-processamento apresentado em [1] foi aplicado e acabou sendo validado. Todavia, adaptações foram necessárias a fim de melhorá-lo, tornando-o mais robusto. Essas alterações fizeram com que alcançássemos eficiências de 92,3% na localização da região da placa.

7.3 Filtro casado

Neste trabalho, foram utilizados filtros casados determinísticos. É de se esperar que a evolução do mesmo seja utilizar filtros estocásticos, uma vez que o problema apresenta estocacidade bastante forte.

Mesmo utilizando filtros casados determinísticos (filtros mais simples de serem implementados), as eficiências do sistema, em torno de 82%, foram satisfatórias. Se formos levar em consideração a eficiência apenas da etapa dos filtros casados, nas imagens já pré-processadas, essas chegam a 97,1%, uma eficiência considerada alta. Por isso, diante deste dado, seria interessante, além do desenvolvimento do filtro estocástico, buscar uma melhora no sistema de pré-processamento.

7.4 Pós-processamento

Como foi mostrado anteriormente, é possível encontrar um limiar de separação para respostas corretas e erradas da maioria dos caracteres. Com isso, seria possível implementar uma etapa de pós-processamento do sistema que serviria como validação da resposta do filtro baseado na placa inteira, bem como, em caso de erro, realimentação do mesmo.

Todavia, para definir os limiares de forma mais precisa, seria necessário fazer a análise em um maior número de placas.

7.5 Implementações

No início do projeto, o pré-processamento estava sendo feito em Matlab, enquanto que o filtro casado era desenvolvido em LabView. A fim de padronizar a linguagem de todos os programas desenvolvidos, optou-se por utilizar apenas Matlab, ferramenta mais poderosa e de domínio de todos envolvidos no projeto.

É sabido que Matlab não é uma linguagem voltada para fins comerciais, uma vez que sua licença é bastante cara e seu processamento, apesar de complexo e poderoso, é lento. Todavia, para validação das técnicas, ele atende perfeitamente.

Para esse projeto, usando um computador Pentium III – 800MHz, uma imagem leva cerca de 4 minutos para ser processada. Esse tempo de decisão inviabiliza uma aplicação comercial.

Com o intuito de tornar essa pesquisa um produto tecnológico, uma vez atingidas eficiências aceitáveis para comercialização (em todas as etapas do projeto), o sistema pode ser implementado em um DSP, a fim de torná-lo barato e rápido.

7.6 Trabalhos futuros

Os resultados apresentados neste trabalho precisam ser melhorados para que seja possível implementar a técnica em um projeto prático. A primeira barreira encontrada é o pré-processamento, que já limita a eficiência do projeto. A utilização de filtros morfológicos é uma das possíveis saídas para a melhoria das eficiências desta primeira etapa.

Os resultados atingidos com os filtros casados com caracteres apontam para o desenvolvimento de um banco de filtros de caracteres, que além de localizar os caracteres, seria capaz de identificá-los. Os limiares de decisão entre as respostas

correspondentes a acertos e erros dos caracteres ficaram bem definidos e isso pode ajudar na validação do resultado, através de realimentação do resultado obtido.

Além disso, o sistema precisa evoluir para a identificação de todos os tipos de placas existentes nas vias brasileiras e ser capaz de identificar ausência de placa na imagem.

Com essas melhorias, o sistema pode se tornar uma aplicação prática.

Referências Bibliográficas

1. Leite, B.B., “Localização Automática de Placas de Veículos Automotores Particulares em Imagens Digitalizadas”, Projeto Final, DEL/UFRJ, junho, 2003.
2. Gesualdi, A.R., “Sistema Neuronal de Reconhecimento de Placas em Veículos Automotores Particulares”, Dissertação de Ms.C., COPPE/UFRJ e CBPF, agosto, 2001.
3. Cinsdikici, M. & Tunali, T, “License Plate Segmentation for Intelligent Transportation Systems”, International Computer Institute Ege University, Turkey.
4. Gonzalez, R.C., Woods, R.E., “Digital Image Processing”, Addison-Wesley Publishing Company, 1993.
5. Bussador, A & Facon, J, “Método Granulométrico para Segmentação de Placas de Veículos”, Dissertação de Ms.C., PUC-PR, Brasil.
6. de Souza, F.P.C. & Susin, A., “SIAV – Um Sistema de Identificação Automática de Veículos”, In: “Anais do XIII Congresso Brasileiro de Automática”, 2000.
7. Marques, R.C.P, Laprano, C.M., Silva, J.L., Medeiros, F.N.S., “License Vehicle Plates in Localization Using Maximum Correlation”, In: “Joint IAPR International Workshops, SSPR 2004 and SPR 2004”, pp. 470, 2004.
8. Martín, F., Borges, D., “Automatic Car Plate Recognition Using a Partial Segmentation Algorithm”, Signal Theory and Communications Department, Vigo University, Spain.

9. Otsu, N., "A Threshold Selection Method for Gray Level Histograms", IEEE Transactions on System, Man and Cybernetics, 1979.
10. <http://www.htsol.com/Products/SeeCar.html>.
11. Albuquerque, Marcelo P., Caner, E.S., Gesualdi, A., Albuquerque, M.P., Leite, B.B., Vasconcelos, C.M., "Image Database of Brazilian Vehicles for Validate License Plate Recognition Systems", Proceedings of XVII SIBGRAPI, 2004.
12. Stearns, S.D. & David, R.A., "Signal Processing Algorithms In Matlab", Prentice Hall, 1996.
13. Kailath, T., "From Matched Filters to Martingales", ISIT'98 Plenary Lecture Report, Proceedings of 1998 IEEE International Symposium on I.T., 1998.
14. Haykin, S., "Communication Systems", John Wiley & Sons, Inc., 1994.
15. Proakis, J.G. & Manolakis, D.G., "Digital Signal Processing – Principles, Algorithms, And Applications", Prentice Hall, 1996.
16. Diniz, P.S.R., da Silva, E.A.B. & Netto, S.,L., "Digital Signal Processing – System Analysis and Design". Cambridge University Press, 2002.
17. Mitra, S.K., "Digital Signal Processing – A Computer-Based Approach". McGraw-Hill, 1998.
18. Castleman, K.R., "Digital Image Processing". Prentice Hall, 1996.

19. Gesualdi, A.R., et al, “Um Sistema Neural de Reconhecimento de placas de Veículos Automotores”, In: “Anais do XIII Congresso Brasileiro de Automática”, pp. 333, 2000.
20. Filho, O.M. & Neto, H.V., “Processamento Digital De Imagens”, Brasport, 1999.
21. Etter, D.M., “Engineering Problem Solving With Matlab”, Prentice Hall, 1997.
22. Oppenheim, A.V. & Schafer, R.W., “Digital Signal Processing”, Prentice Hall, 1975.
23. Shanmugan, K.S. & Breipohl, A.M., “Random Signals”, Wiley, 1988.
24. Knoll, G.F., “Radiation Detection and Measurement”, Wiley, 1989.
25. Van Trees, H.L., “Detection, Estimation and Modulation, part I, II”, Wiley, 1968.
26. de Souza, F.P.C. & Susin, A., “SIAV – Um Sistema de Identificação Automática de Veículos”, In: “Anais do XIII Congresso Brasileiro de Automática”, pp. 1377-1380, 2000.

27. Barroso, J., Rafael, A., Dagless, E.L. & Bulas-Cruz, J., “Number Plate Reading Using Computer Vision”,
<http://www.utad.pt/~jbarroso/html/isie97.html>.

28. Mathworks, <http://www.mathworks.com>

29. National Instruments, <http://www.ni.com>