

UNIVERSIDADE DE SÃO PAULO – ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS ELETRÔNICOS

SISTEMA DE RECONHECIMENTO DE PLACAS DE TRÂNSITO EM TEMPO REAL

Fábio de Albuquerque Spalding

São Paulo
2009

UNIVERSIDADE DE SÃO PAULO – ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA DE SISTEMAS ELETRÔNICOS

SISTEMA DE RECONHECIMENTO DE PLACAS DE TRÂNSITO EM TEMPO REAL

Relatório final do trabalho de formatura a ser apresentado à
Escola Politécnica da Universidade de São Paulo para
obtenção do título de Graduação em Engenharia.

Fábio de Albuquerque Spalding

Orientador: Hae Yong Kim

Área de Concentração:
Engenharia de Sistemas Eletrônicos

São Paulo
2009

RESUMO

Este projeto tem por objetivo o desenvolvimento de um sistema de reconhecimento de placas de trânsito em tempo real por técnicas de análise e processamento de imagens fazendo uso de uma câmera digital simples acoplada ao veículo em movimento e de um computador portátil para execução do algoritmo reconhecedor desenvolvido na linguagem C++.

A câmera realizará a aquisição das imagens em uma determinada taxa de quadros por segundo, e o grande desafio do projeto reside em assegurar que esses quadros consigam ser analisados suficientemente rápido, mas, mesmo assim, garantindo que, quadro a quadro, a identificação das placas de trânsito seja robusta.

No caso de ocorrer esta identificação, o sistema deverá alertar o motorista sobre a passagem por uma nova placa, e notificá-lo para ele saiba qual é a placa de trânsito pela qual ele acabou de passar, garantindo assim que ele não a deixe passar despercebida.

ABSTRACT

The goal of this present project is to develop a real-time road sign recognition system based on image processing and analysis techniques, and making use of a common digital camera for the image acquisition and a portable computer for the execution of the recognizer algorithm written in C++.

The digital camera will acquire the road images in a chosen frame rate, and the main challenge of the project resides in assuring that each frame could be analyzed fast enough in a way that the robustness of the road sign recognition results is guaranteed.

Once the identification of the road sign is achieved, the system is supposed to alert the unaware driver about the sign he has just passed by, and communicate to him which sign it was, keeping him safe in cases he has let it pass unnoticed.

SUMÁRIO

SUMÁRIO

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

ÍNDICE DE ABREVIATURAS

1.	INTRODUÇÃO	1
2.	OBJETIVOS	2
a.	Objetivos primários	2
b.	Objetivos secundários: expansões e melhorias	3
3.	JUSTIFICATIVAS	4
a.	Justificativas técnicas	4
b.	Justificativas práticas	4
4.	METODOLOGIA	5
a.	Primeira fase: Preparação	5
i.	Etapa 1: Aquisição de Vídeos para Teste	5
ii.	Etapa 2: Levantamento e Escolha das Técnicas	6
b.	Segunda fase: Primeiro Algoritmo Funcional	6
i.	Etapa 1: Pré-processamento	6
ii.	Etapa 2: Reconhecimento	7
c.	Terceira fase: Otimização	8
d.	Quarta fase: Funcionamento em Tempo Real	8
5.	DETALHAMENTO DO DESENVOLVIMENTO	9
a.	Reconhecimento dos pixels vermelhos	9
i.	Primeiro critério: Limiar de RGB.	9
ii.	Segundo critério: Limiar de HSI.	11
iii.	Terceiro critério: Faixa de Matiz e Excesso de vermelho	12

iv.	Quarto critério: Recorte no espaço RGB.	14
b.	Reconhecimento de padrões circulares	18
i.	Transformada de Hough circular.	18
ii.	Transformada de Hough circular com gradiente.	20
c.	Legibilidade do conteúdo da placa	22
d.	Delimitação da área para reconhecimento	25
i.	Identificação de elementos conexos	25
ii.	Análise das linhas e colunas	27
e.	Base de templates para reconhecimento	31
f.	Reconhecimento	33
i.	Template Matching	33
6.	DISCUSSÃO DO DESEMPENHO	42
a.	Sobre a captura do vídeo	43
b.	Etapa de detecção de pixels vermelhos	44
c.	Etapa de reconhecimento de padrões circulares	45
d.	Etapa de aumento da legibilidade da placa	46
e.	Etapa de segmentação da região por algarismos	48
f.	Etapa de reconhecimento dos algarismos	49
7.	ANDAMENTO DO PROJETO E METAS ALCANÇADAS	52
a.	Cronograma do projeto	54
8.	GESTÃO DO PROJETO	55
a.	Gestão dos materiais	56
b.	Gestão do tempo	57
9.	CONCLUSÕES	58
BIBLIOGRAFIA		
APÊNDICE A		
APÊNDICE B		

ÍNDICE DE FIGURAS

Fig. 1 - Input: quadro de video colorido.	7
Fig. 2 - Output: placa isolada pronta para a etapa de reconhecimento.	7
Fig. 3 - Fotografia original colorida.	10
Fig. 4 - Pixels vermelhos identificados pelo critério do limiar em RGB.	10
Fig. 5 - Operação de erosão	10
Fig. 6 - Operação de dilatação.	10
Fig. 7 - Diagrama ilustrativo do sistema de cores HSI.	11
Fig. 8 - Quadro original colorido.	12
Fig. 9 - Critério de limiar em HSI, seguida de detecção de borda.	12
Fig. 10 - Quadro original colorido.	13
Fig. 11 - Critério da faixa de matiz e excesso de vermelho.	13
Fig. 12 - Exemplo de placa pouco definida, em dia nublado, com luminosidade média.	14
Fig. 13 - Detalhe da placa observada na Fig. 12.	14
Fig. 14 - Exemplo de placa bem definida, em dia nublado, com boa luminosidade.	14
Fig. 15 - Detalhe da placa observada na Fig. 14.	14
Fig. 16 - Exemplo de placa pouco definida, em dia limpo, com baixa luminosidade.	15
Fig. 17 - Detalhe da placa observada na Fig. 15.	15
Fig. 18 - Exemplo de placa pouco definida, em dia limpo, com alta luminosidade.	15
Fig. 19 - Detalhe da placa observada na Fig. 18.	15
Fig. 20 - Gráfico de correspondência entre as componentes Red e Green	16
Fig. 21 - Gráfico de correspondência entre as componentes Red e Blue	16
Fig. 22 - Gráfico de correspondência entre as componentes Green e Blue.	17
Fig. 23 - Quadro original colorido.	17
Fig. 24 - Critério do recorte no espaço RGB.	17
Fig. 25 - Exemplo da transformada de Hough para círculos.	19

Fig. 26 – Pixels vermelhos identificados na imagem original.	20
Fig. 27 – Saída da Transformada de Hough.	20
Fig. 28 – Exemplo da transformada de Hough adaptada a círculos com o gradiente.	21
Fig. 29 – Exemplo de imagem de placa com baixa luminosidade.	22
Fig. 30 – Histograma correspondente à imagem predominantemente escura.	22
Fig. 31 – Exemplo de imagem de placa com alta luminosidade.	23
Fig. 32 – Histograma correspondente à imagem predominantemente clara.	23
Fig. 33 – Histograma original correspondente à imagem ao lado, da Fig. 34.	24
Fig. 34 – Placa sobre a qual serão efetuadas diferentes normalizações do histograma.	24
Fig. 35 – Normalização do histograma sem saturação.	24
Fig. 36 – Resultado da normalização do histograma, sem saturação.	24
Fig. 37 – Normalização do histograma com saturação.	24
Fig. 38 – Resultado da normalização do histograma, com saturação.	24
Fig. 39 – Imagem obtida após a normalização do histograma.	25
Fig. 40 – Exclusão de todos os pontos distantes a mais de um raio do centro da placa.	25
Fig. 41 – Exemplo da 4-conectividade.	26
Fig. 42 – Exemplo de malha conexa pela 4-conectividade.	26
Fig. 43 – Exemplo ilustrativo de busca de elementos conexos.	26
Fig. 44 – Imagem com a placa identificada e com o histograma normalizado.	27
Fig. 45 – Detalhe da região ao redor da placa identificada.	27
Fig. 46 – Detalhe de uma placa identificada e com o histograma normalizado.	28
Fig. 47 – Três gráficos. O gráfico mais abaixo mostra a soma S_j dos níveis de cinza de todos os pontos da coluna j . O gráfico ao centro mostra a primeira derivada da soma S . O gráfico acima mostra a segunda derivada de S .	29
Fig. 48 – Três exemplos de quadros originais coloridos, antes do processamento.	30
Fig. 49 – Algarismos são localizados e separados.	30
Fig. 50 – Detalhe da Fig. 48, mostrando mais claramente a separação dos algarismos.	30
Fig. 51 – Base de templates a serem reconhecidos.	31

Fig. 52 – Esquema ilustrativo do vetor de armazenamento da base de templates.	32
Fig. 53 – Proporção entre tamanho dos signos e tamanho da placa.	32
Fig. 54 – Exemplo ilustrativo de respostas para o Template Matching	34
Fig. 55 – Quadro original colorido	35
Fig. 56 - Detalhe da separação dos algarismos	35
Fig. 57 – Quadro original colorido	37
Fig. 58 – Detalhe da separação dos algarismos	37
Fig. 59 – Cinco quadros sucessivos referentes a uma mesma placa de 30 km/h	39
Fig. 60 – Cronograma do primeiro semestre de projeto.	55
Fig. 61 – Cronograma do segundo semestre de projeto	55

ÍNDICE DE TABELAS

Tabela 1 – Template matching aplicado ao primeiro algarismo da placa da Fig. 54.	36
Tabela 2 – Template matching aplicado ao segundo algarismo da placa da Fig. 54.	36
Tabela 3 – Template matching aplicado ao primeiro algarismo da placa da Fig. 56.	37
Tabela 4 – Template matching aplicado ao segundo algarismo da placa da Fig. 56.	38
Tabela 5 – Template matching no primeiro algarismo de cinco quadros sucessivos	39
Tabela 6 – Template matching no segundo algarismo de cinco quadros sucessivos.	40

ÍNDICE DE ABREVIATURAS

AVI	–	Audio Video Interleave
RGB	–	“Red, Green, Blue” color model
HSI	–	“Hue, Saturation, Intensity” color model
GPS	–	Global Positioning System
OpenCV	–	Open Source Computer Vision Library
JPEG	–	Joint Photographic Experts Group

1. INTRODUÇÃO

A maior parte das informações que recebemos do mundo exterior é por via visual. Assim sendo, não é de se espantar que tão logo a tecnologia tenha se tornado desenvolvida o bastante, técnicas de análise, processamento e síntese de imagens venham sendo desenvolvidas cada vez mais freqüentemente, numa variedade cada vez maior de áreas.

Estimulada inicialmente pela necessidade de áreas do conhecimento em que a imagem é um objeto de trabalho central, como a astronomia, a medicina e o cinema, as técnicas de processamento de imagens passaram por uma verdadeira revolução ao longo das últimas décadas.

Hoje, o processamento de imagens é uma técnica já amplamente difundida. É graças aos conhecimentos desenvolvidos nessa área e à sua grande flexibilidade de aplicações que hoje nós temos a possibilidade de aprimorar várias de nossas tarefas em diversos campos do conhecimento, como o controle de qualidade, o controle de processos industriais, o controle de tráfego, as previsões meteorológicas, a astronomia, a área de segurança, as imagens médicas, a indústria do cinema e entretenimento, e tantas outras áreas quanto se possa desejar.

É justamente por ser tão adaptável que se propõe nesse projeto o estudo e o desenvolvimento de algoritmos de processamento de imagens para facilitar uma tarefa quase obrigatória do dia-a-dia, que é a de dirigir. Quanto maior o conforto e a segurança com que se dirige, melhor acaba por se tornar a própria qualidade de vida do motorista. É com esse intuito que o projeto do reconhecimento de placas de trânsito em tempo real será desenvolvido ao longo dos próximos dois semestres.

2. OBJETIVOS

Os objetivos do projeto serão divididos em dois grupos: o dos objetivos primários, que definem o plano inicial mínimo traçado para o projeto, e os objetivos secundários, que compreendem expansões no escopo mínimo do projeto e melhorias no seu desempenho e apresentação.

a. Objetivos primários

O objetivo primário, que compõe, de fato, o sujeito deste projeto, é o de se obter ao seu término um algoritmo capaz de solucionar o problema proposto de reconhecimento de placas de trânsito em tempo real, com grande confiabilidade, pois se trata de uma solução que afeta diretamente a segurança de seu usuário, e performante o bastante para que sua utilização não requeira uma capacidade de processamento demasiado alta.

Para a satisfação desse objetivo primário, serão utilizados a princípio os conhecimentos existentes em manipulação de imagens digitais, e diversas técnicas computacionais que foram sendo criadas ao longo das últimas décadas. O objetivo do trabalho pretende que essas técnicas sejam criteriosamente escolhidas garantindo a compatibilidade entre si.

Além do emprego de técnicas de processamento de imagens, o projeto passará por uma fase de otimização para que a qualidade do produto final seja aceitável. Nessa fase, serão aplicadas medidas sucessivas de melhoria de algoritmo. O desempenho do algoritmo resultante será posto à prova durante a fase final do projeto, em que o algoritmo será testado em campo para verificar se os objetivos traçados para o projeto foram atendidos.

b. Objetivos secundários: expansões e melhorias

A depender do ritmo de evolução do projeto, objetivos secundários poderão ser implementados, como uma interface simples na qual o condutor do veículo poderá consultar os resultados das análises realizadas pelo algoritmo. Essa interface poderá ser realizada por meio de um aviso sonoro, ou de um display em que o limite de velocidade apresentado pela placa seja mostrado.

Outro objetivo secundário pensado é estender a escopo inicial de reconhecer apenas placas de limite de velocidade para placas quaisquer que possuam o mesmo formato das placas de velocidade (isto é, placas circulares vermelhas). Isso requereria a princípio apenas o enriquecimento da base de dados para *template matching*, e, dependendo do desempenho do algoritmo para o escopo expandido, possivelmente também seriam necessárias etapas suplementares de otimização.

O projeto também poderá se estender para qualquer placa de sinalização de trânsito, incluindo aquelas em que o modelo não seja o mesmo. No entanto, isso requereria um algoritmo muito mais complexo devido à multiplicidade de formas de placa existentes (triangulares, octogonais, losangonais), e uma robustez muito grande, pois os signos tornam-se cada vez mais complexos e detalhados, o que demandaria o uso de uma maior resolução para a aquisição do vídeo de modo a não comprometer o resultado do *template matching*, e uma capacidade de processamento muito maior.

Eventualmente, outras melhorias e expansões poderão ser pensadas no decorrer do projeto.

3. JUSTIFICATIVAS

a. Justificativas técnicas

Até muito recentemente, a única maneira de se obter informações sobre o espaço exterior ao carro era prestando grande atenção ao que se passava fora dele. Aos poucos foram se desenvolvendo acessórios capazes de proporcionar ao motorista informações cada vez mais confiáveis sobre a pista, sem que ele precise olhar para o exterior do veículo. É o caso de sistemas como radares de proximidade e o GPS. Em julho de 2008 a empresa Vauxhall, uma divisão da General Motors na Inglaterra, anunciou a inclusão de um reconhecedor de placas de trânsito para seu modelo Insignia já em 2009. Essa tecnologia no entanto permanece restrita a um artigo de luxo.

A intenção deste projeto é buscar uma maneira mais simples e mais barata de se conseguir um resultado de nível comparável, de forma a garantir um bom compromisso entre preço e segurança, tendo sempre em mente a busca de uma nova técnica que possa se adaptar a veículos populares.

b. Justificativas práticas

O projeto parte de uma necessidade real. Um motorista pode deixar de perceber a existência de uma placa de trânsito por diversos motivos, seja por distração, por condições de visibilidade comprometidas ou pelo simples fato de que o carro andava rápido demais para que houvesse tempo o bastante para reconhecer o que a placa de trânsito sinalizava. Como decorrência, várias situações indesejáveis podem ocorrer, desde ser surpreendido por uma lombada até transitar em velocidade acima da permitida sem ter consciência do erro. É nesse contexto prático que o projeto de um sistema de reconhecimento de placas de trânsito em tempo real se justifica, pois se propõe a trazer uma segurança a mais para o motorista.

4. METODOLOGIA

Para o desenvolvimento do projeto, serão utilizadas as seguintes ferramentas: na fase de aquisição dos vídeos, será utilizada a câmera Canon Powershot A95, que realizará a aquisição de vídeo no formato AVI (Audio Video Interleave) na resolução de 320x240 pixels, a uma taxa de 15 frames por segundo, o que resulta em uma taxa de dados de 2314kbps. Os vídeos adquiridos serão submetidos ao reconhecimento por meio de um programa desenvolvido na linguagem C++, que contará com o uso da quarta versão biblioteca ProEikon, gentilmente cedida pelo seu criador e orientador deste projeto, o professor Hae Yong Kim, voltada para o Processamento de Imagens e Visão Computacional. O programa também contará com o uso da biblioteca multiplataforma OpenCV (Open Source Computer Vision Library), criada pela Intel e livre para o uso acadêmico e comercial voltado ao desenvolvimento de aplicativos em Visão Computacional.

O projeto será dividido em quatro fases. Na primeira fase, que foi desenvolvida ao longo do primeiro semestre, é realizada a aquisição dos vídeos que serão utilizados como base para o desenvolvimento e teste do algoritmo de reconhecimento ainda offline, e é realizado o levantamento e estudo das possíveis técnicas de Processamento de Imagens capazes de solucionar o problema, e a escolha das técnicas a serem utilizadas na fase seguinte. Na segunda fase, os algoritmos escolhidos são implementados primeiramente para um único quadro do vídeo, e posteriormente para todo um vídeo. Na terceira fase o algoritmo é otimizado, para que seja rápido o bastante para reconhecer placas na taxa de 15 quadros por segundo adotada. Na quarta fase, o algoritmo será testado para vídeos adquiridos em tempo real.

a. Primeira fase: Preparação

i. Etapa 1: Aquisição de Vídeos para Teste

Nessa etapa, foram adquiridos os vídeos de 320 x 240 pixels a uma taxa de 15 quadros por segundo que servirão de base para o desenvolvimento do algoritmo de reconhecimento. A fim de contemplar a vasta multiplicidade de situações capazes de

afetar a aquisição dos vídeos, foram adquiridas amostras para diversas situações (vídeos noturnos e diurnos, dias nublados e ensolarados, placas velhas e em bom estado de conservação). Novas amostras serão adquiridas na medida em que se fizerem necessárias.

ii. Etapa 2: Levantamento e Escolha das Técnicas

Foram levantadas as técnicas existentes, avaliadas suas características, e, em parceria com o orientador deste projeto, foram escolhidas aquelas que melhor atendiam aos requisitos do problema proposto, a saber: velocidade, flexibilidade a mudanças, robustez à multiplicidade de entradas.

b. Segunda fase: Primeiro Algoritmo Funcional

i. Etapa 1: Pré-processamento

Cada quadro do vídeo passará pelo mesmo algoritmo de reconhecimento. Esse algoritmo realiza uma sequência de processamentos sobre o quadro original a fim de identificar a presença de uma placa, e seu conteúdo.

A primeira etapa do algoritmo isola da imagem original apenas os pixels que são percebidos ao olho humano como vermelhos. Além desse julgamento efetuado pixel a pixel, podem ser necessárias etapas suplementares para garantir que a saída a ser entregue para a etapa seguinte seja de boa qualidade segundo os critérios adotados para se garantir a velocidade e a robustez.

Em seguida, a imagem está pronta para a etapa de reconhecimento do padrão circular. Para este fim, utilizou-se uma generalização da transformada de Hough para círculos. A transformada de Hough pode ser generalizada para formas parametrizáveis [4], assim obtém-se a parametrização do círculo e a partir daí pode-se inferir com uma certa dose de certeza se a forma identificada era ou não um círculo na imagem original.

Tendo-se, com isso, conhecido o raio e o centro da placa, podemos partir para o reconhecimento. No entanto, outras etapas de processamento podem ser de nosso interesse, principalmente por possibilitarem um processamento mais ágil e um índice de acertos mais importante.

Assim, ao final da etapa de pré-processamento, obteríamos como saída a região da imagem original sobre a qual será efetuado o reconhecimento. As figuras a seguir ilustram a idéia pretendida com a etapa de pré-processamento, em que o input representa um quadro do vídeo analisado e o output representa a citada região para reconhecimento.



Fig. 1 - Input: quadro de vídeo colorido.

Pré
Processamento



Fig. 2 - Output: placa isolada pronta para a etapa de reconhecimento.

ii. Etapa 2: Reconhecimento

O reconhecimento será realizado por *template matching* simples. O objetivo primário do projeto contempla o reconhecimento de placas indicativas de limite de velocidade, o que limita o *template matching* aos dez algarismos (de 0 a 9). Dessa maneira, a imagem com a placa isolada obtida na etapa de pré-processamento será transformada em uma imagem binária, sobre a qual se fará uma convolução com os dez algarismos, ou uma simples comparação pixel a pixel, de modo a identificá-los. Na sequência, será determinado o posicionamento de cada um deles na imagem, para que se possa determinar qual o valor do limite de velocidade escrito na placa.

c. Terceira fase: Otimização

Primeiramente, o algoritmo foi desenvolvido e testado apenas para um quadro. A fase de otimização visa aplicá-lo para todo um vídeo, e verificar o tempo que o algoritmo leva para ser executado. Para garantir que o algoritmo é empregável em tempo real, deve-se garantir que seu tempo de execução não exceda a taxa de aquisição de quadros. Isso significa que o algoritmo deve ser executado em no máximo:

$$\frac{1 \text{ segundo}}{15 \text{ quadros}} = 0,066 \text{ segundos}$$

Para se alcançar tal resultado, será avaliada a necessidade de desenvolver uma pré-programação capaz de identificar prontamente se a aplicação do algoritmo é ou não necessária para um dado quadro. Além disso, as placas de trânsito aparecerão na grande maioria dos casos apenas nas laterais do quadro, o que restringe ainda mais o número de operações que o programa deverá executar.

d. Quarta fase: Funcionamento em Tempo Real

Nessa última fase, após haver-se garantido que o tempo de execução do algoritmo foi suficientemente otimizado, a câmera de vídeo será ligada a um notebook em um carro em movimento, de modo que o algoritmo seja executado na medida em que o vídeo vai sendo gravado. É nesse momento que será verificada a taxa de acerto do algoritmo e sua confiabilidade. Caso a taxa de acerto seja muito baixa, os parâmetros de calibração do algoritmo (índices-limite do componente matiz, grau de prevalência da componente vermelha sobre as componentes azul e verde, elemento realizador da erosão/dilatação, faixa de raios prováveis que uma placa poderá apresentar, critérios de determinação da probabilidade de ocorrência de uma placa em um quadro) deverão ser revistos, de modo a garantir a robustez que se pretende para o algoritmo.

5. DETALHAMENTO DO DESENVOLVIMENTO

a. Reconhecimento dos pixels vermelhos

Num primeiro momento, o algoritmo vinha sendo desenvolvido para um único quadro. Foram usadas imagens JPEG de resolução 640×480 durante essa fase, e foram obtidos bons resultados empregando-se o limiar de RGB como critério decisório. No entanto, ao se aplicar o mesmo para um vídeo, os resultados não foram apropriados, exigindo que um próximo critério fosse elaborado. Nesse momento decidiu-se por trabalhar em outro espaço de cores, o HSI, e dois possíveis critérios foram estudados: o limiar de HSI e a união de uma faixa permitida de matiz com o excesso de vermelho. Durante a fase de otimização, esse critério veio a ser revisto para se agilizar o processamento, e foi adotado como critério definitivo o recorte cuidadoso do espaço tridimensional RGB. Abaixo, são melhor explicados cada um dos critérios desenvolvidos ao longo do projeto.

i. Primeiro critério: Limiar de RGB.

Esse critério permite que sejam identificados na imagem original apenas os pixels em que a componente vermelha do espaço de cores RGB é preponderante em relação às componentes verde e azul. Para determinar qual é a quantidade de vermelho mínima e quais as quantidades de azul e verde máximas para identificarmos os vários tons vermelhos de uma placa, avaliou-se a resposta para vários valores, e visualmente inspecionou-se o resultado (v. Apêndice A), de modo a se obter:

$$R_{\min} = 145 \text{ (em uma escala que vai de 0 a 255)}$$

$$G_{\max} = 150 \text{ (em uma escala que vai de 0 a 255)}$$

$$B_{\max} = 150 \text{ (em uma escala que vai de 0 a 255)}$$

O resultado da primeira etapa de processamento com a calibração realizada de acordo com os valores acima resulta numa imagem binária em que os pixels pretos representam os pixels vermelhos da imagem original, e os pixels brancos representam todos os outros pixels. A citada imagem binária de resultado é apresentada na seguinte figura:



Fig. 3 - Fotografia original colorida.



Fig. 4 - Pixels vermelhos identificados pelo critério do limiar em RGB.

Em seguida, era interessante que a imagem resultante passasse por um processo de suavização, para eliminar os pontos espúrios e sem significado. Para isso, ela passa por processos que se baseiam em técnicas de Morfologia Matemática Binária. Os processos utilizados são os de erosão e dilatação com um elemento circular de 3×3 pixels, que, aplicados seguidamente um do outro (o que consiste em uma operação de abertura), resultam numa imagem "limpa". O resultado das etapas de erosão e dilatação são mostrados nas imagens binárias abaixo:



Fig. 5 - Operação de erosão aplicada à imagem binária correspondente aos pixels vermelhos.



Fig. 6 - Operação de dilatação aplicada à imagem binária resultante da operação de erosão.

O que se observou, no entanto, foi que a adoção dos critérios acima, que nos primeiros testes com as primeiras fotos haviam funcionado aceitavelmente, não era tão bem sucedida no processamento dos quadros em um vídeo. O critério foi avaliado sob a luz dessas novas constatações. Concluiu-se que pelo fato de os quadros de um vídeo apresentarem uma qualidade muito inferior àquela de uma fotografia seria preciso elaborar um segundo critério, mais severo, que se adequasse às restrições impostas pela aquisição de um vídeo.

ii. Segundo critério: Limiar de HSI.

À medida que a primeira fase veio sendo desenvolvida, testou-se sua eficiência, que já havia sido verificada para um único quadro, em um vídeo gravado. No entanto os resultados acima apresentados nem sempre foram suficientes para satisfazer às necessidades de um vídeo. Em um vídeo, o simples teste das componentes Red, Green e Blue do sistema RGB não se mostrou eficaz na detecção de pontos vermelhos. Assim, foi necessário o desenvolvimento de novos critérios complementares que possibilitassem essa identificação.

Assim, foram testados diferentes novos critérios. O primeiro deles foi a avaliação da cor de cada pixel individualmente, mas tomando uma nova base de cores, a HSI, cujas componentes Matiz (*Hue*), Saturação e Intensidade foram analisadas através da mesma metodologia aplicada à base RGB.

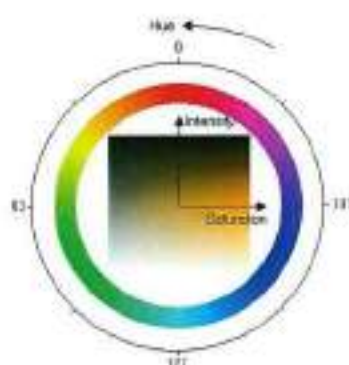


Fig. 7 – Diagrama ilustrativo do sistema de cores HSI.

Numa escala circular de 0 a 255, no sistema HSI, os pixels vermelhos são aqueles em torno do valor $0 = 256$. Procedendo de modo análogo ao critério anterior de busca de um limiar por inspeção visual, os limites encontrados para as componentes HSI foram:

$$\begin{cases} H_{\min} = 231 \\ \text{ou} \\ H_{\max} = 63 \end{cases}$$

$$S_{\max} = 55$$

$$I_{\max} = 60$$

Com esses critérios a placa era sempre muito bem identificada, porém muitos outros pixels, principalmente os muito próximos de cinza, eram identificados como vermelhos, o que exigiu uma etapa suplementar consistindo na passagem por um filtro de reconhecimento de borda.



Fig. 8 - Quadro original colorido.



Fig. 9 - Pixels vermelhos identificados pelo critério de limiar em HSI, seguida de detecção de borda.

O resultado final obtido é muito discernível para um ser humano, mas muito pouco robusto para ser manipulado com segurança pelas etapas seguintes. Além disso, a adição de um filtro suplementar ia contra a meta principal do trabalho que é a de um processamento veloz.

iii. Terceiro critério: Faixa de Matiz e Excesso de vermelho

Os critérios acima foram repensados. As conclusões foram que a componente Matiz era um bom critério de discriminação de pixels predominantemente vermelhos. Já as componentes Saturação e Intensidade pouco auxiliavam nessa detecção. Assim, os limites encontrados para as componentes HSI foram:

$$\begin{cases} H_{\min} = 231 \\ \text{ou} \\ H_{\max} = 63 \end{cases}$$

$$S_{\max} = \text{indiferente}$$

$$I_{\max} = \text{indiferente}$$

Da mesma maneira, foi pensado num novo critério que utilizasse a base de cores RGB, segundo o qual se determinaria que um pixel é vermelho pela comparação da intensidade da componente vermelha sobre as componentes verde e azul.

Assim, o critério final adotado para a detecção de um pixel vermelho foi:

$$\begin{cases} R > G + \Delta_{RG} \\ \text{e} \\ R > B + \Delta_{RB} \end{cases} \quad (1)$$

Onde o resultado ótimo foi encontrado para:

$$\Delta_{RG} = 1$$

$$\Delta_{RB} = 1$$

Abaixo, são mostrados exemplos da eficiência dos novos critérios quando aplicadas a um vídeo:



Fig. 10 - Quadro original colorido.



Fig. 11 - Pixels vermelhos identificados pelo critério da faixa de matiz e excesso de vermelho.

Os resultados obtidos para esses novos critérios se adequaram muito bem às condições de aquisição de um vídeo, e foram robustos o bastante para a etapa de identificação de padrões circulares que lhe seguiria. No entanto, durante a fase de otimização percebeu-se que a utilização de critérios baseados no espaço de cores HSI, se possível, deveriam ser evitados, pois a cada pixel analisado havia uma etapa de conversão do espaço RGB para termos HSI que tomava tempo. Diante disso, nas etapas de otimização, o critério baseado na estipulação de uma faixa de matiz foi abandonado, e foi um critério final que, em termos de RGB, fosse robusto o bastante.

iv. Quarto critério: Recorte no espaço RGB.

Este último critério para identificação de pixels vermelhos é um complemento ao primeiro critério testado, de aferição de um limiar das componentes R, G e B a ser respeitado.

Relembrando que o problema que levou ao abandono do primeiro critério foi a baixa performance para as imagens de vídeo, em que os resultados incluíam não apenas os pixels vermelhos como muitos outros, seguiu-se um método bastante rígido para o recorte do espaço de cores RGB.

Em primeiro lugar, foram estudados quatro diferentes quadros de diferentes vídeos, contemplando toda a variedade de circunstâncias ambientais. Os quadros estudados são mostrados abaixo:



Fig. 12 – Exemplo de placa pouco definida, em dia nublado, com luminosidade média.



Fig. 13 – Detalhe da placa observada na Fig. 12.



Fig. 14 – Exemplo de placa bem definida, em dia nublado, com boa luminosidade.



Fig. 15 – Detalhe da placa observada na Fig. 14.



Fig. 16 – Exemplo de placa pouco definida, em dia limpo, ao final da tarde, com baixa luminosidade.



Fig. 17 – Detalhe da placa observada na Fig. 15.



Fig. 18 – Exemplo de placa pouco definida, em dia limpo, com alta luminosidade.



Fig. 19 – Detalhe da placa observada na Fig. 18.

Foram escolhidos, dentre os quadros acima, 200 pixels que são percebidos pelo olho humano como vermelhos. As componentes Red, Green e Blue de cada um desses pixels foram anotadas. Abaixo, cada um desses pixels são representados em três projeções bidimensionais (RG, RB, GB) do espaço tridimensional RGB. Ao lado de cada projeção, estão listadas o conjunto de regras para o recorte do espaço de cores, obtido a partir de cálculos feitos com base nos próprios gráficos:

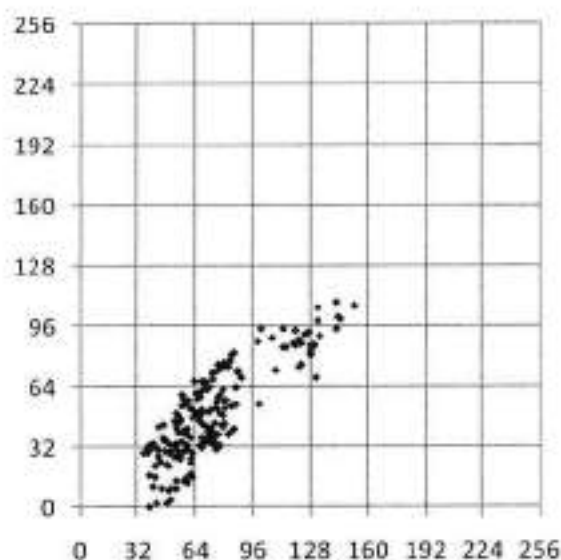


Fig. 20 - Gráfico de correspondência entre as componentes Red (eixo horizontal) e Green (eixo vertical) para os pixels percebidos como vermelhos pelo olho humano.

$$\left\{ \begin{array}{l} R > G - 6 \\ e \\ R < G + 52 \\ e \\ R > 35 \\ e \\ R < 150 \end{array} \right. \quad (2)$$

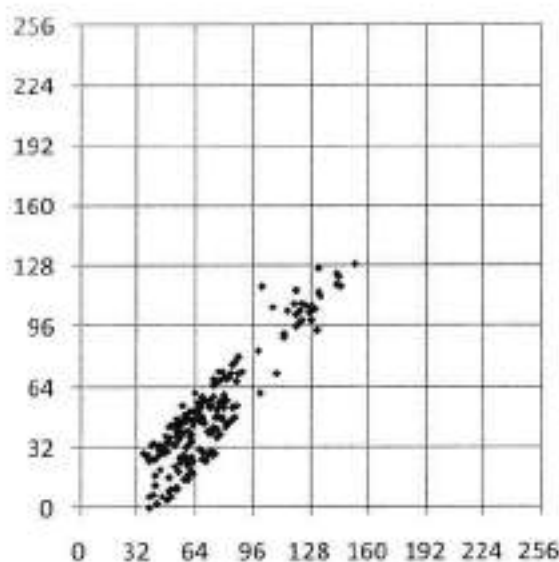


Fig. 21 - Gráfico de correspondência entre as componentes Red (eixo horizontal) e Blue (eixo vertical) para os pixels percebidos como vermelhos pelo olho humano.

$$\left\{ \begin{array}{l} 6R > 5B - 100 \\ e \\ R < B + 41 \\ e \\ R > 36 \\ e \\ R < 150 \end{array} \right. \quad (3)$$

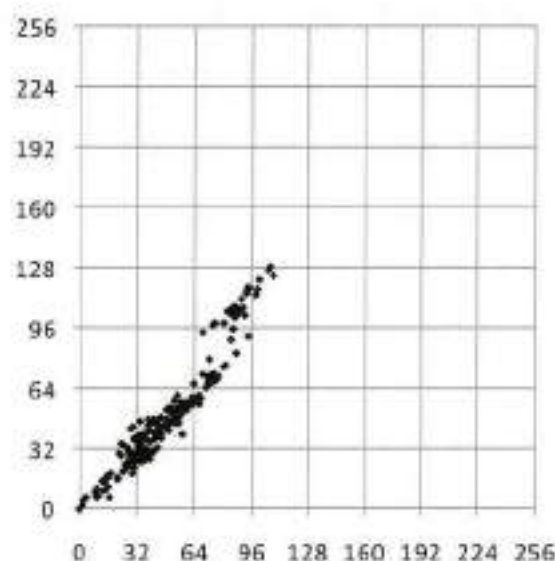


Fig. 22 - Gráfico de correspondência entre as componentes Green (eixo horizontal) e Blue (eixo vertical) para os pixels percebidos como vermelhos pelo olho humano.

$$\left\{ \begin{array}{l} B > G - 11 \\ 3B < 4G + 11 \end{array} \right. \quad (4)$$

Com os critérios de recorte enunciados acima, conseguiu-se um resultado bastante robusto, ainda que ainda falho para alguns tons muito próximos do cinza. Mesmo assim, foi o mais robusto e mais seguro resultado entre os estudados, e por isso foi o escolhido como definitivo. Abaixo, segue o exemplo do resultado do critério acima aplicado ao mesmo quadro dos exemplos anteriores.



Fig. 23 - Quadro original colorido.



Fig. 24 - Pixels vermelhos identificados pelo critério do recorte no espaço RGB.

b. Reconhecimento de padrões circulares

Como já foi citado, é possível generalizar a transformada de Hough para quaisquer outras formas parametrizáveis. No caso do forma circular a que nos pretendemos identificar, a parametrização é simples:

$$r = \sqrt{(l - l_0)^2 + (c - c_0)^2} \quad (5)$$

O espaço de saída da transformada de Hough para círculos possui três dimensões: raio do círculo, linha em que se encontra o centro, coluna em que se encontra o centro.

Sabemos que todas as posições de linha e coluna da imagem original devem ser analisadas. Assim, cada uma dessas dimensões está limitada às dimensões da imagem bidimensional de entrada, a saber: 240 pixels para a dimensão linha, 320 pixels para a dimensão coluna.

Observa-se, também, nos vários vídeos adquiridos, que os círculos também têm seus raios limitados a uma faixa. Assim, convém adotar uma faixa de valores prováveis para a dimensão raio. O limite superior pode ser determinado pela simples observação dos vídeos. Nota-se que, devido à distância entre a câmera e as placas, estas nunca aparecem na imagem com raio superior a 18 pixels. Quanto ao limite inferior, percebeu-se que para os círculos com raio inferior a 8 pixels a imagem interna da placa se tornava indiscernível até mesmo para o olho humano, o que impossibilitaria um reconhecimento seguro. Assim, a dimensão raio foi limitada ao intervalo [8, 18].

Duas abordagens diferentes para a transformada de Hough foram testadas. Elas são apresentadas a seguir.

i. Transformada de Hough circular.

A transformada de Hough adaptada a círculos funciona da seguinte maneira:

Para cada pixel aceso (i, c) da imagem analisada é “desenhado” um círculo de raio r na imagem de saída. A cada pixel da imagem de saída é associado, ao invés de uma cor, um valor *float*. Quando dois pontos se superpõem, eles têm seus valores somados. Dessa maneira, após se ter repetido o mesmo procedimento para todos os pixels acesos da imagem original, teremos como imagem de saída uma matriz de mesmas dimensões da imagem original com vários círculos “desenhados”. No caso de haver na imagem original um círculo de raio r , haverá uma superposição de vários círculos no pixel correspondente às posições do centro do círculo. Esse pixel terá um valor muito superior aos outros pixels da imagem, o que permite que seja identificado.

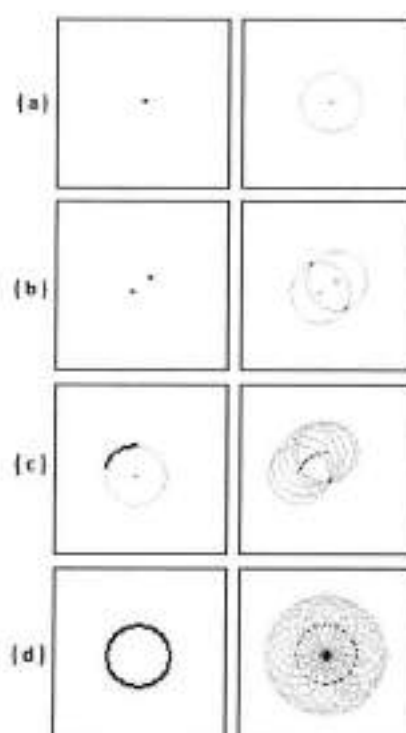


Fig. 25 – Exemplo ilustrativo de respostas da transformada de Hough para círculos aplicada à busca de um círculo de raio 16 pixels: (a) Resposta para um único ponto. (b) Resposta para dois pontos. (c) Resposta para um arco de círculo. (d) Resposta para um círculo. Observa-se, na resposta para este último caso, a grande concentração coincidindo exatamente com a posição do centro de um círculo de raio 16 pixels.

Para fins de visualizar o processo acima, geram-se figuras em escala de cinza para representar a imagem de saída, em que o ponto de maior valor recebe o valor 255 (representado em preto na figura abaixo), e o ponto de menor valor recebe o valor 0 (representado em branco). Aos valores intermediários são identificados valores correspondentes linearmente em níveis de cinza.

Um exemplo das respostas obtidas para a transformada de Hough generalizada para círculos aplicadas sobre uma imagem ilustrativa de seu funcionamento estão disponíveis no Apêndice B. Abaixo, obteve-se para o mesmo quadro que já foi mostrado acima que há um círculo de raio igual a 14 pixels com centro claramente identificado por uma concentração com coordenadas (85,295). A imagem a seguir mostra a identificação do círculo correspondente à placa de trânsito:



Fig. 26 - Imagem binária correspondente aos pixels vermelhos identificados na imagem original.



Fig. 27 - Saída da Transformada de Hough com uma concentração no pixel (85, 295), evidenciando a existência de um círculo de raio 14 ali centrado.

O algoritmo de identificação do centro adota que serão considerados como pontos identificadores de círculos de raio r todos aqueles que possuam ao menos 50% dos pixels à distância r acesos na imagem original. Essa taxa foi obtida a partir de vários testes com diferentes taxas. Observou-se que adotando taxas mínimas superiores a 50%, vários círculos deixavam de ser identificados. Por outro lado, para taxas mínimas inferiores a 50%, por vezes elementos da imagem original que não eram círculos passavam a ser identificados como tais.

À medida que eventuais aprimoramentos no reconhecimento dos pixels vermelhos forem feitos, é de se esperar que essa taxa de 50% possa ser aumentada, garantindo assim a robustez do algoritmo.

ii. Transformada de Hough circular com gradiente.

Uma segunda alternativa para o mesmo problema consiste na incorporação de uma pequena mudança à técnica apresentada anteriormente. Ao invés de cada pixel “aceso” da imagem original binária gerar um círculo completo na matriz de saída, leva-se em consideração o gradiente naquele ponto, que agora será de uma imagem em escala de cinza. Dessa maneira, ao invés de marcarmos $2\pi r$ pontos da matriz de saída, marcaremos apenas a região desse círculo para onde o gradiente daquele ponto original aponta.

Como o trabalho é todo realizado em pontos discretos, e não sobre uma curva contínua, o gradiente calculado corre o risco de ser uma aproximação ligeiramente grosseira. Desse modo, ao invés de marcarmos na matriz de saída apenas um ponto do círculo na direção para onde o gradiente aponta, marcaremos toda uma pequena região ao redor desse ponto.

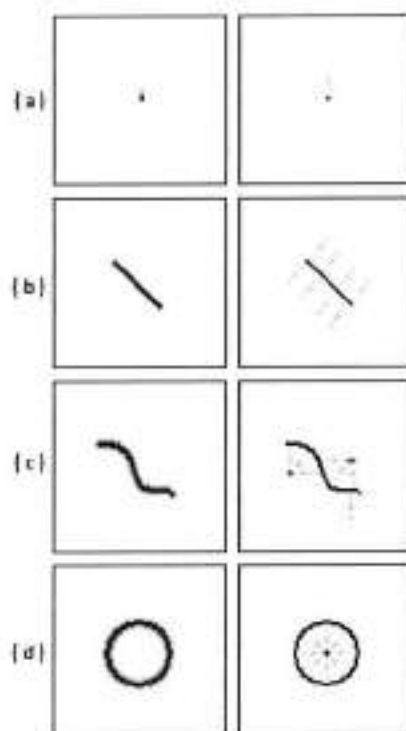


Fig. 28 – Exemplo ilustrativo da resposta da transformada de Hough adaptada a círculos quando se leva em conta o gradiente. (a) Resposta para um ponto. (b) Resposta para uma reta. (c) Resposta para uma curva sinuosa. (d) Resposta para um círculo.

Com essa pequena mudança se tem a quantidade de pontos marcados na saída drasticamente reduzida, o que reduz o ruído, as grandes quantidades de pontos inutilmente marcados.

c. Legibilidade do conteúdo da placa

A partir desse momento não há mais necessidade de trabalharmos com imagens coloridas. A necessidade de imagens coloridas se dá somente na primeira etapa, em que precisamos identificar os pixels vermelhos. Daqui em diante, é mais leve e simples trabalharmos com imagens em níveis de cinza.

No entanto, o que se nota é que dependendo da luminosidade do ambiente no momento da captura do vídeo, a placa pode estar mais ou menos nítida. Assim, julgou-se necessário uma etapa de processamento para normalizar a nitidez da placa. Isso será feito a partir da normalização do histograma da região onde a placa está localizada.

O histograma de uma imagem corresponde à distribuição estatística dos níveis de cinza presentes na placa, em que cada nível de cinza presente na placa possui uma taxa de ocorrência. A partir da análise do histograma pode-se saber, entre outras informações, se a imagem apresenta um alto ou baixo contraste, e se a imagem é predominantemente clara ou escura.

Vemos abaixo uma ilustração de sobre como o histograma de uma imagem clara se diferencia daquele de uma imagem escura:



Fig. 29 – Exemplo de imagem de placa com baixa luminosidade.

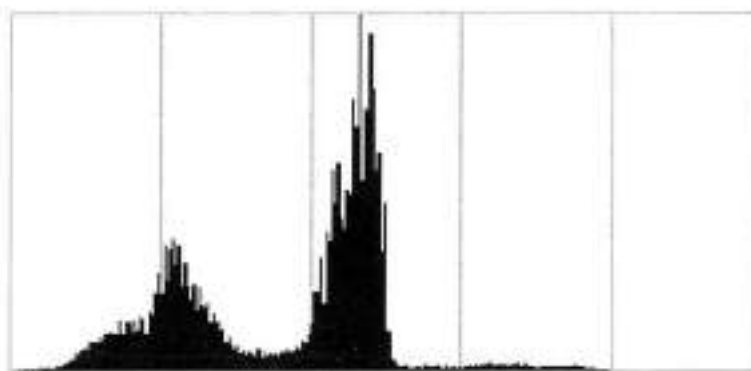


Fig. 30 – Histograma correspondente à imagem predominantemente escura exibida ao lado, na Fig. 29. No eixo horizontal os vários níveis de cinza, indo do preto ao branco. No eixo vertical a incidência de pixels com aquele nível de cinza. Nota-se a grande concentração de pontos deslocada à esquerda, na zona mais próxima ao preto.



Fig. 31 - Exemplo de imagem de placa com alta luminosidade.

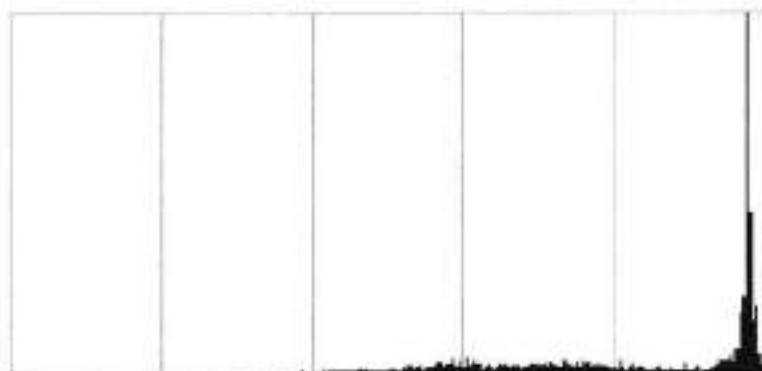


Fig. 32 - Histograma correspondente à imagem predominantemente clara exibida ao lado, na Fig. 31. Nota-se a grande concentração de pontos deslocada à direita, na zona mais próxima ao branco.

Diante desse exemplo fica clara a marcante diferença que pode existir entre duas imagens diferentes, e a razão pela qual se julgou necessário realizar a normalização do histograma.

O algoritmo para esta normalização se dará da seguinte forma. Primeiramente serão determinados os níveis de cinza mais claro e mais escuro presentes na imagem analisada. O intervalo entre esses dois níveis de cinza conterá todos os níveis de cinza existentes na imagem.

A normalização consiste em definir qual será o intervalo da imagem original a ser transportado para a imagem de histograma normalizado, na qual os tons de cinza variarão do preto ao branco. Desse intervalo original, todos os pontos cujo nível de cinza estiver abaixo do nível do cinza escolhido como limite inferior serão saturados em preto, e analogamente todos aqueles que estiverem acima do limite superior do intervalo serão saturados em branco. Aos pontos intermediários lhes são atribuídos seus novos valores por interpolação linear.

Abaixo, pode-se ver como mudanças no histograma podem afetar a legibilidade da placa. É adotado como primeiro exemplo a mudança em que se parte do intervalo original que compreende todos os pontos da placa, sem haver saturação; em seguida, é mostrado o exemplo de uma normalização do histograma em que há saturação. Os critérios mostrados no segundo exemplo são os escolhidos para o programa, após observação de vários outros critérios, com base na qualidade da saída.

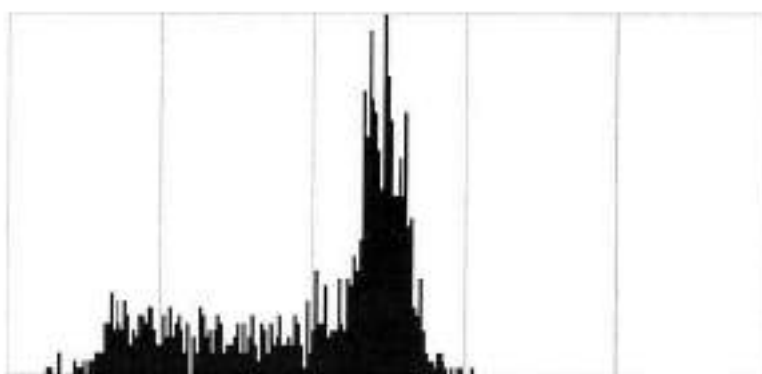


Fig. 33 - Histograma original correspondente à imagem ao lado, da Fig. 34.



Fig. 34 - Exemplo de imagem de placa sobre a qual serão efetuadas diferentes normalizações do histograma.

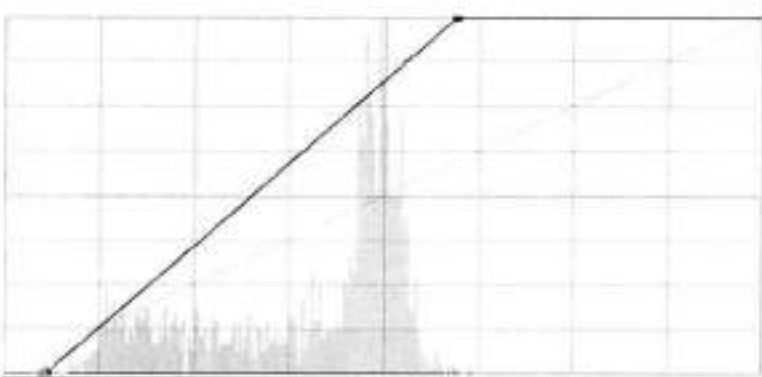


Fig. 35 - Normalização do histograma sem saturação. Ao fundo, na cor cinza claro, o histograma da imagem original. A reta em cinza claro que trespassa toda a imagem representa os limites do intervalo de níveis de cinza originais. A reta preta indica os novos limites do intervalo, em que o limite inferior corresponde ao ponto mais escuro da imagem original e o superior corresponde ao mais claro.



Fig. 36 - Resultado da normalização do histograma, sem saturação.

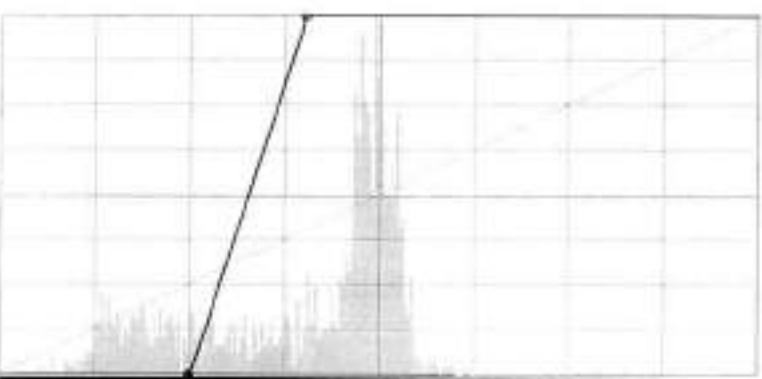


Fig. 37 - Normalização do histograma com saturação. A reta preta indica os novos limites do intervalo. O limite inferior foi escolhido acima do ponto mais escuro, a uma distância de 0,33 vezes o intervalo original. Abaixo dele, todos os pontos serão saturados em preto. O limite superior foi escolhido abaixo do ponto mais claro, a uma distância de 0,38 vezes o intervalo original. Acima desse ponto os pontos serão saturados em branco.



Fig. 38 - Resultado da normalização do histograma, com saturação e parâmetros: 0,33 e 0,38.

d. Delimitação da área para reconhecimento

Nessa etapa, pretende-se delimitar as áreas da imagem em que se encontram cada um dos algarismos que deverão ser reconhecidos.

Para esse fim, em primeiro lugar nos limitamos a analisar apenas os pixels internos da placa, excluindo assim a sua borda vermelha. Basta, para tanto, pintar de branco todos os pixels a uma distância superior ou igual ao raio r da placa encontrada:



Fig. 39 – Imagem obtida após a normalização do histograma.

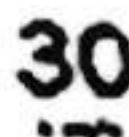


Fig. 40 – Imagem obtida após a exclusão de todos os pontos distantes a mais de um raio do centro da placa.

Na imagem resultante tem-se apenas o conteúdo da placa, com os algarismos referentes à velocidade máxima permitida. Precisa-se identificar, agora, cada um dos algarismos separadamente, para que o reconhecimento seja feito de forma mais veloz por estar restrito a uma área muito menor. Duas diferentes abordagens foram testadas, e são avaliadas criticamente a seguir.

I. Identificação de elementos conexos

A primeira abordagem adotada foi a identificação dos vários elementos conexos separadamente. Um elemento conexo é aquele cujos pontos são todos ligados um ao outro segundo um critério de conectividade. O critério de conectividade adotado foi a 4-conectividade, em que um ponto é dito conectado a outro caso se encontre em uma das seguintes quatro posições relativas a ele: acima, abaixo, à esquerda ou à direita:



Fig. 41 – Exemplo da 4-conectividade.

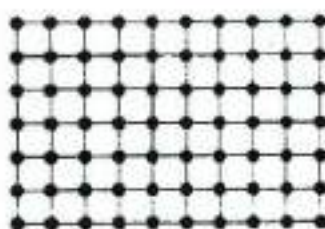


Fig. 42 – Exemplo de malha conexa pela 4-conectividade.

Assim, buscavam-se na imagem de entrada pontos cujo nível cinza fosse inferior a um determinado limite (estabeleceu-se *limite* < 36 numa escala de nível de cinza em que 0 é o valor dado aos pixels pretos e 255 aos brancos). Todos os pontos que estivessem ligados a esse ponto segundo o critério de conectividade adotado eram considerados como pixels de um mesmo elemento conexo.

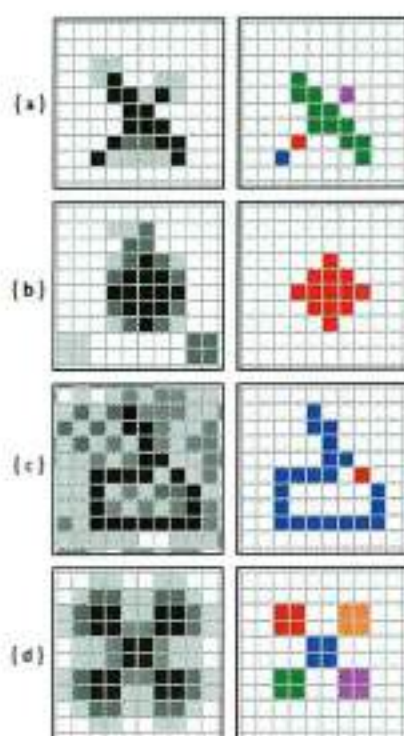


Fig. 43 – Exemplo ilustrativo de busca de elementos conexos, considerando os critérios de 4-conectividade e de um limite máximo de valor 36, em escala de cinza. Em cada exemplo (a), (b) (c) e (d), à esquerda se situa a imagem original, e à direita são coloridos distintamente cada um dos elementos conexos encontrados.

Desta maneira, varrendo toda a área analisada, podemos encontrar todos os elementos conexos ali presentes.

Os resultados, no entanto, apresentavam baixa taxa de satisfatoriedade. Nas imagens maiores, em placas de raios superiores a 12 pixels, os algarismos apresentavam-se claramente separados, o que não acontecia nas placas menores. Desta maneira, julgou-se que essa abordagem seria muito eficiente para vídeos capturados em resoluções superiores, mas para a resolução escolhida de 320×240 pixels seria preciso pensar numa segunda abordagem.

ii. Análise das linhas e colunas

A partir da mesma imagem de base, essa nova abordagem determina a localização da imagem na figura por meio de uma simples soma dos valores do nível de cinza dos pixels em uma mesma linha e uma mesma coluna.



Fig. 44 – Imagem com a placa identificada e com o histograma de sua região circunvizinha normalizado.

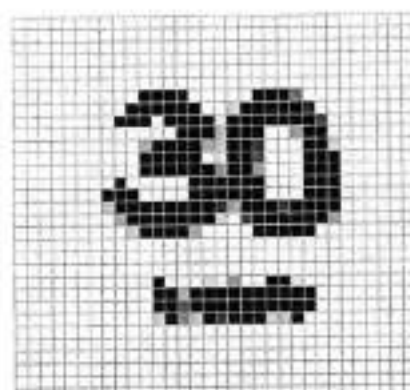


Fig. 45 – Detalhe da região ao redor da placa identificada.

Observando a imagem aumentada acima, podemos claramente deduzir que com esse artifício será possível localizar exatamente onde há uma mancha preta e onde não há. Abaixo seguem os valores das somas nas linhas e colunas para o exemplo citado:

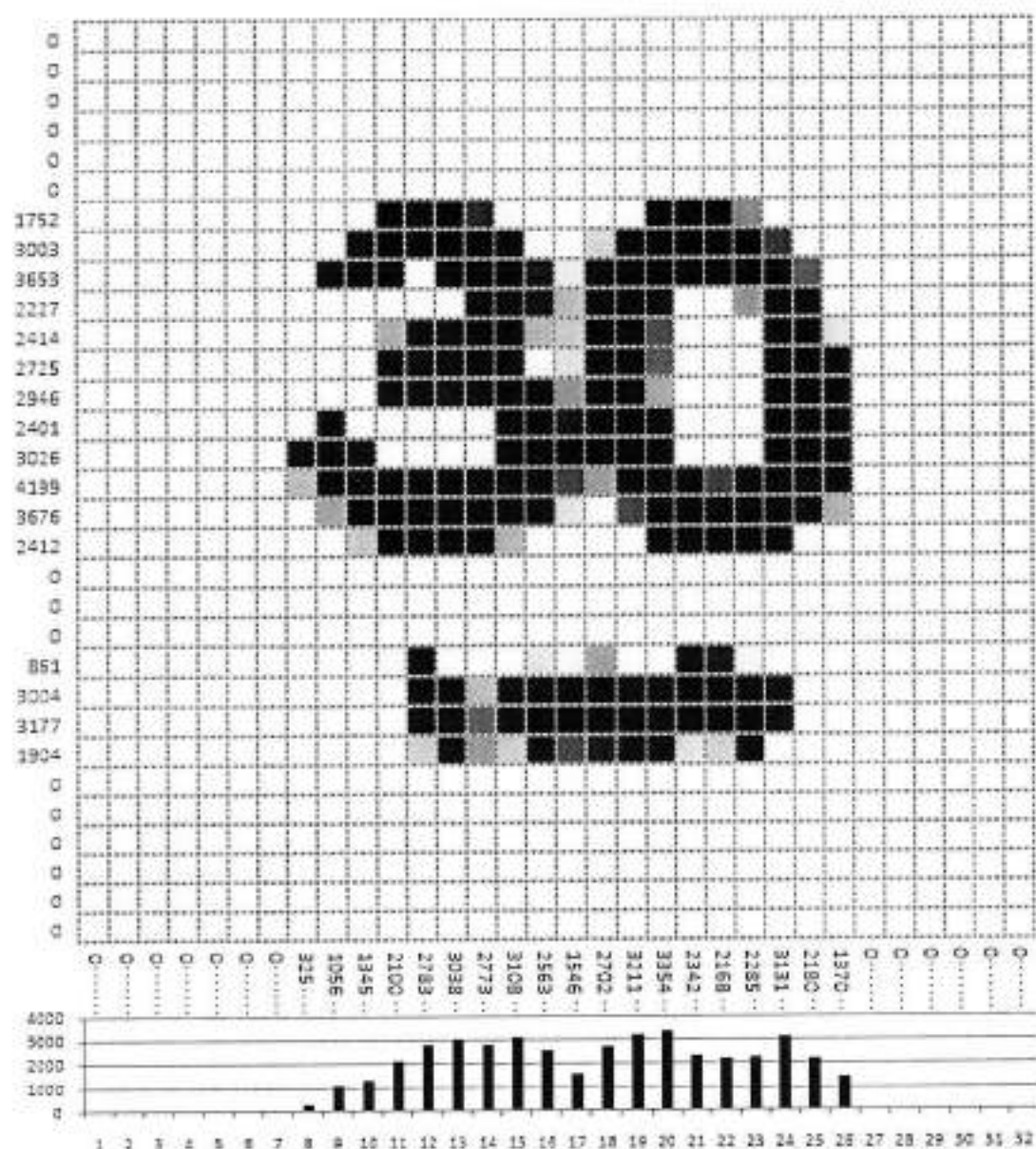


Fig. 46 – Detalhe da área em torno de uma placa identificada e com o histograma normalizado. À esquerda de cada uma das linhas da imagem, a soma dos níveis de cinza de todos os pontos daquela linha (considerando, para esse caso especificamente, que o branco se situa no 0 e o preto em 255). Abaixo de cada coluna, analogamente, a soma dos níveis de cinza de todos os pontos da coluna. Abaixo de todo o conjunto, a representação gráfica das somas dos níveis de cinza das colunas.

Com isso, fica evidente, por exemplo, que a imagem está localizada entre a 8ª e a 26ª colunas. Da mesma maneira, fica evidente que existe uma imagem entre a 7ª e a 18ª linhas, e uma outra imagem entre 22ª e a 25ª linhas.

Se, por um lado, já localizamos a região onde se encontram os algarismos, falta-nos separar um do outro. Se observarmos atentamente o gráfico colocado logo abaixo da grade, observamos que na 17ª coluna, que separa um algarismo do outro, há uma

queda na soma. Essa queda foi identificada em todas as outras imagens que foram analisadas. A partir dessa observação, podemos estabelecer um método para identificar a linha de separação entre os dois algarismos.

Observamos que na coluna de separação há uma súbita queda seguida imediatamente de um novo aumento nos valores das somas das colunas. Esse comportamento nos remete a uma concavidade acentuada, assim foram analisadas as derivadas primeira e segunda da soma S em relação à distância na horizontal x . Seja, então, a imagem em níveis de cinza im de dimensões l_{max} e c_{max} .

$$s_j = \sum_{i=1}^{l_{max}} 255 - im(i, j) \quad (6)$$

$$\frac{dS}{dx} = \frac{\Delta S_j}{\Delta j} = \frac{S_{j+1} - S_j}{j+1-1} = S_{j+1} - S_j \quad (7)$$

$$\frac{d^2 S}{dx^2} = (S_{j+2} - S_{j+1}) - (S_{j+1} - S_j) = S_{j+2} - 2S_{j+1} + S_j \quad (8)$$

Aplicadas ao caso analisado anteriormente, obtemos o seguinte perfil:

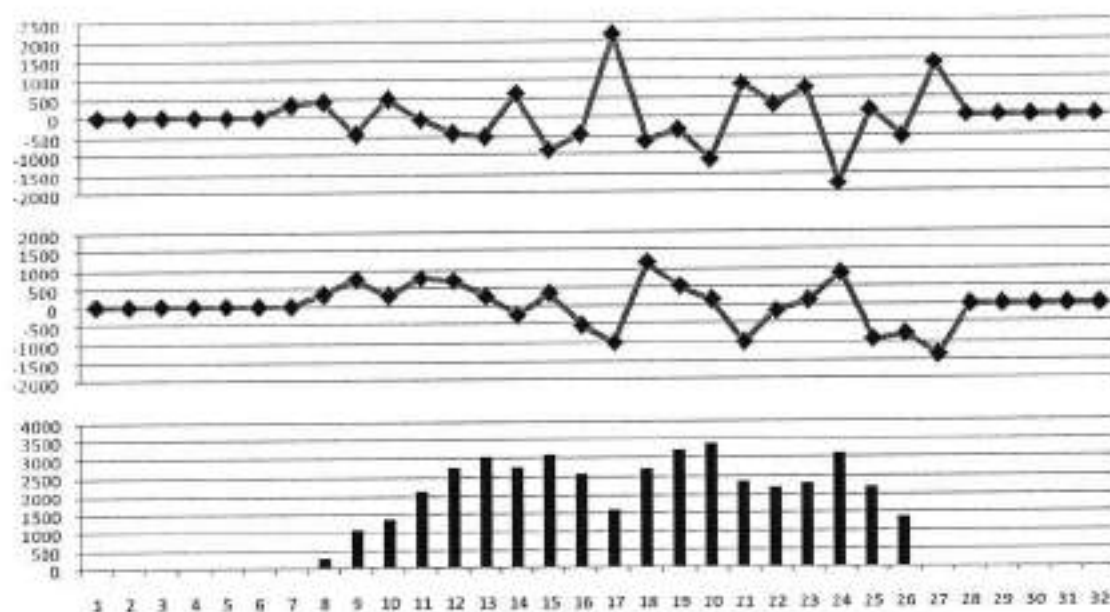


Fig. 47 – Três gráficos, relativos à Fig. 45, já previamente analisada na Fig. 46. O gráfico mais abaixo mostra a soma S_j dos níveis de cinza de todos os pontos da coluna j . O gráfico ao centro mostra a primeira derivada da soma S em relação à distância horizontal x . Finalmente, o gráfico acima mostra a segunda derivada de S em relação a x . Nota-se, no gráfico da segunda derivada, que o pico denuncia a linha de separação entre os algarismos 3 e 0 da imagem original.

Como esperado, o valor máximo da segunda derivada indica exatamente a 17ª coluna, que é aquela que separa os dois algarismos. Com base nessa observação, e também com base na análise empregada quanto às somatórias das linhas horizontais, que permitem separar os algarismos numéricos da imagem *km/h*, podemos então determinar com alta taxa de acerto as localizações de cada algarismo.

Abaixo, são mostrados alguns exemplos de identificação de algarismos em diferentes placas.



Fig. 48 – Três exemplos de quadros originais coloridos, antes de qualquer etapa de processamento.

Fig. 49 – Após a localização e normalização das placas, os algarismos são localizados e separados.

Fig. 50 – Detalhe da Fig. 48, mostrando mais claramente a separação dos algarismos.

e. Base de templates para reconhecimento

Antes de prosseguir com o reconhecimento, é preciso montar a base com todos os templates que buscamos reconhecer na imagem.

Conforme o escopo traçado pelo projeto, são dez os templates que buscamos reconhecer: 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9, que são exatamente os algarismos numéricos que aparecem nas placas de velocidade. Além deles, sabe-se que em todas elas aparece a imagem *km/h* sob a velocidade, mas justamente pelo fato de a imagem aparecer em todas elas de nada adianta, para o escopo definido para o projeto, reconhecer também esses signos, uma vez que eles não provêm auxílio algum na diferenciação entre duas placas distintas entre si.

Outro ponto que vale destacar é a precaução que se deve tomar contra a invariância de tamanho do template a ser reconhecido. Dependendo da distância da placa à câmera, os algarismos podem aparecer em tamanhos que variam dentro de um intervalo. Esse intervalo, verificado após análise de diversos vídeos gravados mantendo-se um mesmo posicionamento da câmera em relação ao automóvel, nos diz que os algarismos variam de 6 a 16 pixels de altura. Logicamente, para placas distantes, a imagem pode aparecer em tamanho inferior, porém para tamanhos muito pequenos a chance de erro é muito alta, e esses casos não serão avaliados.

A base de templates montada em níveis de cinza no formato JPEG é exposta abaixo:



Fig. 51 – Base de templates a serem reconhecidos, compreendendo os algarismos numéricos de 0 a 9 em tamanhos que variam de 6 a 16 pixels de altura. As imagens no formato JPEG estão em níveis de cinza.

Quando o programa for utilizado, ele começa por gravar a base de templates exibida a cima em um vetor de imagens tridimensionais em níveis de cinza. Cada índice do vetor corresponde à altura, em pixels, do algarismo da base de templates. Desta maneira às células de índices 6 a 16 serão atribuídas imagens tridimensionais. Cada imagem tridimensional equivalerá a um conjunto de dez imagens de mesmas dimensões, uma para cada algarismo a ser comparado com o fragmento de quadro extraído das etapas precedentes de pré-processamento.

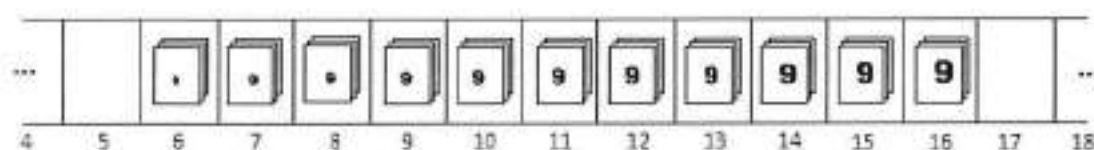


Fig. 52 – Esquema ilustrativo do vetor de armazenamento da base de templates.

Dependendo do raio do círculo encontrado, pode-se determinar a altura em pixels esperada para os algarismos, por haver sempre uma proporção entre os tamanhos da placa e dos signos ali inscritos. Com essa informação, sabemos em qual das imagens tridimensionais (i.e., conjuntos de templates de uma mesma altura) devemos realizar o template matching. A proporção, calculada em 36,4%, é ilustrada abaixo.

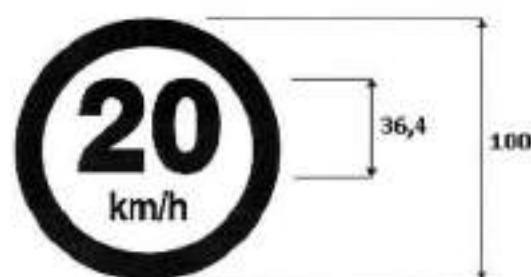


Fig. 53 – Proporção entre tamanho dos signos e tamanho da placa.

f. Reconhecimento

A etapa de reconhecimento consiste na comparação de uma região destacada do quadro original com o conjunto das imagens dos dez algarismos. Essa identificação pode ser feita de mais de uma maneira, cada uma apresentando vantagens e desvantagens.

Por exemplo, a comparação por *Template Matching* é bastante simples, mas ela exige que certos critérios sejam cumpridos pela área onde se realizará essa comparação. Por ser sensível a brilho e contraste, é necessário que a imagem original apresente um brilho comparável ao das imagens na base de template, e quanto maior o contraste entre o signo a ser identificado e o fundo da imagem, maior a segurança sobre a resposta.

Uma segunda técnica poderia ter sido utilizada, como, por exemplo, a técnica baseada na Correlação, que, apesar de mais complexa, não é sensível a brilho e contraste, e dispensa que se façam ajustes destas duas características.

i. Template Matching

A idéia do reconhecimento por *Template Matching* em níveis de cinza é bastante intuitiva. Como já se sabe, em uma imagem em nível de cinza, a cada pixel é associado um valor de intensidade I que pode variar de 0 a 255 em que o valor inferior corresponde à cor preta, o valor superior à cor branca, e os valores intermediários correspondem às diversas gradações de cinza. A idéia do *Template Matching* é comparar pixel a pixel duas imagens a e b de mesmo tamanho, medindo-se a diferença entre estes dois pixels de posição (l, c) , que será calculada por:

$$Dif_{a,b}(l, c) = \frac{1}{255} \cdot |I_a(l, c) - I_b(l, c)| \quad (9)$$

Os resultados da comparação entre dois pontos variarão no intervalo $[0, 1.0]$. A diferença máxima, de valor 1.0, ocorrerá na comparação entre um pixel preto e um pixel branco. Já a ocorrência da diferença mínima, 0, indicará que os dois pontos são idênticos, isto é, possuem exatamente a mesma intensidade.

Generalizando o conceito para uma imagem de dimensões $i \times j$, a distância entre essas duas imagens de mesmo tamanho será:

$$Dif_{a,b} = \frac{1}{i \cdot j} \cdot \sum_{l=1}^i \sum_{c=1}^j Dif_{a,b}(l, c) \quad (10)$$

Do mesmo modo, o resultado variará no intervalo $[0, 1.0]$, em que o valor 1.0 ocorrerá na comparação entre duas imagens binárias (em preto e branco) em que uma seja o exato oposto da outra, enquanto que o valor 0 ocorrerá na comparação de duas imagens idênticas. Quanto menor a diferença, maior a chance de essas duas imagens serem correspondentes. A idéia fica mais clara no exemplo abaixo:

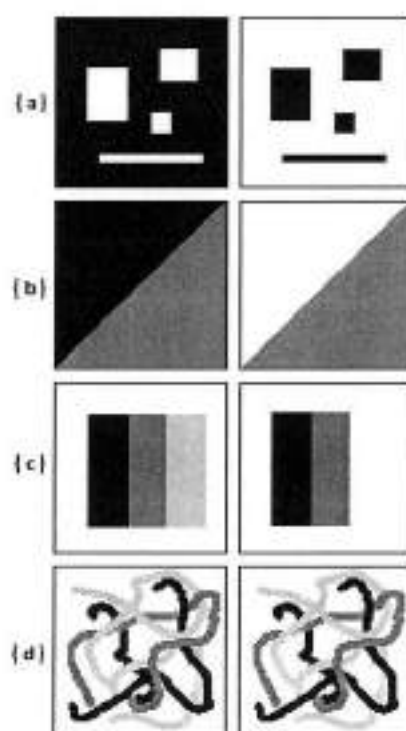


Fig. 54 – Exemplo ilustrativo de respostas para o Template Matching entre duas figuras. (a) Duas imagens binárias inversas, $Dif = 1.0$ (b) Duas imagens metade coincidentes, metade opostas, $Dif = 0.5$. (c) Duas imagens bastante semelhantes, $Dif = 0.1$. (d) Duas imagens idênticas, $Dif = 0.0$.

Para o caso deste projeto, estaremos trabalhando com duas imagens de tamanhos distintos. A primeira é a área do quadro original, que denotaremos por Im_a de

dimensões $l_a \times c_a$. A segunda imagem será a de um dos algarismos que compõe a base de templates, denotada por Im_b de dimensões $l_b \times c_b$. É imperativo que as dimensões de Im_a sejam superiores às de Im_b , ou seja, que tanto $l_a \geq l_b$ como $c_a \geq c_b$.

Assim, para duas imagens de tamanhos distintos, a operação de *template matching* será aplicada a cada região da imagem maior Im_a de mesmo tamanho de Im_b , resultando num total de $(l_a - l_b + 1) \cdot (c_a - c_b + 1)$ operações de *template matching*. O menor dos resultados dessas operações será escolhido como aquele em que houve uma maior semelhança entre as imagens.

Serão mostrados como exemplo, abaixo, os resultados do *template matching* aplicado ao quadro estudado anteriormente para cada um dos dez templates da base. O quadro ao qual o *template matching* fará referência é o logo abaixo:



Fig. 55 - Quadro original colorido



Fig. 56 - Detalhe da separação dos algarismos

Primeiramente, observemos os resultados obtidos para o primeiro algarismo. Nota-se que a menor diferença ocorre justamente na comparação com a imagem "3". A identificação pode ser julgada bastante segura, visto que a semelhança entre a imagem original e o a imagem do algarismo "3" presente na base de templates é consideravelmente maior que as outras. Na tabela abaixo, observamos que o fator de semelhança com o algarismo "3" é realmente superior, e que o segundo algarismo mais parecido seria o algarismo "8".

Tabela 1 – Resultados do *template matching* aplicado ao primeiro algarismo da placa da Fig. 55.

3

Template testado	Diferença entre imagens
Template Matching com 0:	0.414914
Template Matching com 1:	0.526738
Template Matching com 2:	0.367451
Template Matching com 3:	0.290802
Template Matching com 4:	0.481432
Template Matching com 5:	0.353595
Template Matching com 6:	0.404753
Template Matching com 7:	0.463886
Template Matching com 8:	0.337195
Template Matching com 9:	0.367879

Da mesma forma são apresentados na tabela abaixo os resultados do *template matching* para o segundo algarismo da placa. Mais uma vez o resultado encontrado esteve correto.

Tabela 2 – Resultados do *template matching* aplicado ao segundo algarismo da placa da Fig. 55.

0

Template testado	Diferença entre imagens
Template Matching com 0:	0.233298
Template Matching com 1:	0.557100
Template Matching com 2:	0.393155
Template Matching com 3:	0.358930
Template Matching com 4:	0.401218
Template Matching com 5:	0.355472
Template Matching com 6:	0.283779
Template Matching com 7:	0.550980
Template Matching com 8:	0.304884
Template Matching com 9:	0.298253

Desta forma, os algarismos “3” e “0” foram corretamente identificados, permitindo que a placa de limite máximo de velocidade estipulado em 30 km/h fosse reconhecida.

No entanto, certas vezes pode acontecer de o *template matching* indicar um resultado incorreto. Isso pode ocorrer por diversas razões.

- É possível encontrar placas que possuem manchas ou desgaste;
- a grande maioria das placas seguem uma mesma tipografia, no entanto há placas que fogem à regra, e as diferenças podem ser bastante significativas;
- em certos momentos objetos externos, como folhagem por exemplo, podem obstruir a placa;
- certos signos são muito semelhantes uns aos outros, como por exemplo o signo “3” e o signo “8”, de modo que por uma mínima diferença o signo errado pode ser identificado como se fosse o correto;
- dependendo da distância da câmera à placa a imagem dos signos fica tão reduzida que a diferenciação torna-se sutil demais para a técnica empregada.

Abaixo é mostrado um exemplo de um quadro em que a sutil diferença entre o signo “3” e o signo “9” não pôde ser bem percebida, de modo que o resultado do algoritmo de *template matching* para identificação do algarismo foi mal-sucedida. O quadro estudado é exibido abaixo:



Fig. 57 – Quadro original colorido



Fig. 58 – Detalhe da separação dos algarismos

Ao analisarmos os resultados exibidos na tabela abaixo, percebemos que a menor diferença entre a imagem da placa e uma imagem de template da base se deu para o algarismo “9”. Nota-se também que o fator de semelhança com o algarismo “3” ficou muito pouco atrás, numa diferença inferior a 2%.

Tabela 3 – Resultados do *template matching* aplicado ao primeiro algarismo da placa da Fig. 57.

Template testado	Diferença entre imagens
Template Matching com 0:	0.334259
Template Matching com 1:	0.402876
Template Matching com 2:	0.347495
Template Matching com 3:	0.325436
Template Matching com 4:	0.326928
Template Matching com 5:	0.352070
Template Matching com 6:	0.358660
Template Matching com 7:	0.438998
Template Matching com 8:	0.334913
Template Matching com 9:	0.319336

3

Já o segundo algarismo, "0", foi corretamente identificado, como vemos abaixo:

Tabela 4 – Resultados do *template matching* aplicado ao segundo algarismo da placa da Fig. 57.

Template testado	Diferença entre imagens
Template Matching com 0:	0.279521
Template Matching com 1:	0.475980
Template Matching com 2:	0.425436
Template Matching com 3:	0.430065
Template Matching com 4:	0.442919
Template Matching com 5:	0.392102
Template Matching com 6:	0.324945
Template Matching com 7:	0.485948
Template Matching com 8:	0.355011
Template Matching com 9:	0.321460

0

Por conta desta situação, ao invés de se determinar o algarismo que melhor corresponde à imagem para cada quadro individualmente, concluiu-se que a avaliação dos sucessivos quadros que correspondam a uma mesma placa resultará numa identificação mais segura.

Assim, a partir do momento em que se detecta a existência de um círculo vermelho em um quadro, todos os círculos que aparecerem nos quadros sucessivos serão considerados como novas imagens de uma mesma placa, até que não se identifiquem mais círculos, ou seja, até que a placa seja ultrapassada. Os resultados do *template matching* para cada um dos quadros com relação a um determinado algarismo da base de templates serão somados. Ao final, a soma que resultar menor será considerada como indicativo da melhor correspondência entre a placa e um dos algarismos da base de templates.

Para o exemplo anterior, observem-se os diversos quadros do vídeo que correspondem à mesma placa:



Fig. 59 – Cinco quadros sucessivos referentes a uma mesma placa de 30 km/h, ordenados cronologicamente da esquerda para a direita, de cima para baixo.

A esses quadros foi aplicado o *template matching* com relação a cada um dos algarismos numéricos, de acordo com o tamanho com que o signo aparece no quadro original. É de se esperar que quanto maior a proximidade da placa à câmera no momento da captura, maiores as chances de obtermos um resultado correto para o *template matching*. A seguir, os resultados dessa operação são compilados em duas tabelas, uma referente a cada um dos dois algarismos da placa.

Tabela 5 – Resultados do template matching aplicado ao primeiro algarismo de cinco quadros sucessivos. Em seguida, os fatores atribuídos aos quadros individualmente são somados todos, obtendo-se a resposta.

Template testado	Quadro 1	Quadro 2	Quadro 3	Quadro 4	Quadro 5	Todos os quadros
0	0.294826	0.337778	0.334259	0.342538	0.414914	1.724315
1	0.327996	0.391242	0.402876	0.314651	0.526738	1.963503
2	0.378105	0.385098	0.347495	0.318355	0.367451	1.796504
3	0.279272	0.310763	0.325436	0.335403	0.290802	1.541676
4	0.236874	0.323834	0.326928	0.335512	0.481432	1.704580
5	0.304202	0.299608	0.352070	0.371024	0.353595	1.680499
6	0.334259	0.351765	0.358660	0.361819	0.404753	1.811256
7	0.389924	0.394684	0.438998	0.385294	0.463886	2.072786
8	0.302233	0.313682	0.334913	0.325109	0.337195	1.613132
9	0.336220	0.324967	0.319336	0.362908	0.367879	1.711310
Escolha	4	5	9	1	3	3

Ao observarmos os resultados acima, notamos que a tentativa de reconhecimento do conteúdo de uma placa baseado na análise individual de um quadro oferece grande risco. Dos cinco quadros do vídeo analisado, quatro deles teriam sido erroneamente identificados caso a abordagem adotada fosse essa. Notamos que para cada um dos cinco casos a resposta correta, “3”, obteve sempre um índice de semelhança com a imagem original muito grande, mas nos quatro primeiros casos houve sempre algum outro template da base que era ligeiramente mais semelhante: 8.5% mais semelhante no primeiro quadro, 3.5% no segundo, 1.9% no terceiro e 6.2% no quarto.

No entanto, ao analisarmos a placa ao longo de toda a sequência de quadros que a compõem, atribuindo pesos iguais a cada um deles, notamos que a resposta correta se sobressai diante das outras. Na média das semelhanças, a resposta esperada “3” é 9.6% mais semelhante do que a segunda mais semelhante, “8”. É interessante notar que ao utilizarmos esse critério mais amplo, situações curiosas saltam aos olhos, como por exemplo o fato de o algarismo “1”, identificado como correto ao analisarmos o Quadro 4 individualmente, ser, na média, a segunda pior opção. Do mesmo modo, o algarismo “8”, que em nenhum dos quadros individuais foi apontado como o mais semelhante, aparece na média dos vários quadros como o segundo mais semelhante dos dez, a frente dos algarismos “4”, “5”, “9” e “1”, que foram apontados como as melhores opções para os quatro primeiros quadros.

Já a análise do segundo algarismo nos mostra uma situação muito mais regular, como observamos na tabela abaixo:

Tabela 6 – Resultados do template matching aplicado ao segundo algarismo de cinco quadros sucessivos. Em seguida, os fatores atribuídos aos quadros individualmente são somados todos, obtendo-se a resposta.

Template testado	Quadro 1	Quadro 2	Quadro 3	Quadro 4	Quadro 5	Soma dos quadros
0	0.215826	0.174782	0.279521	0.173943	0.233298	1.077370
1	0.485839	0.465468	0.475980	0.475163	0.557100	2.459550
2	0.405672	0.369826	0.425436	0.345360	0.393155	1.939449
3	0.376681	0.304412	0.430065	0.301699	0.358930	1.771787
4	0.390468	0.361057	0.442919	0.391198	0.401218	1.986860
5	0.342227	0.294771	0.392102	0.309368	0.355472	1.693940
6	0.290126	0.236656	0.324945	0.245272	0.283779	1.380778
7	0.522619	0.488181	0.485948	0.439597	0.550980	2.487325
8	0.281863	0.253758	0.355011	0.249368	0.304884	1.444884
9	0.282073	0.246351	0.321460	0.242004	0.298253	1.390141
Escolha	0	0	0	0	0	0

Nesse caso, em particular, tanto nas análises individuais de cada quadro quanto na média de todos os quadros estudados o resultado “0” apontado manteve-se constante e correto.

No entanto este é um caso especial. Em boa parte das vezes os resultados decorrentes de um quadro individual são equivocados. E por isso é mais seguro e confiável que se adote como regra a análise de todos os quadros componentes da placa como um todo.

Um possível aprimoramento para o critério acima consistiria na atribuição de pesos aos diferentes quadros. É de se esperar que os últimos quadros, por conterem imagens da placa mais próxima à câmera, sejam mais nítidos e, portanto, confiáveis. Com base nesta observação, seria possível aumentar o peso dos quadros em que a placa esteja mais próxima e diminuir o peso dos quadros em que a placa esteja mais distante. Essa aferição de pesos pode ser feita com base no raio, previamente calculado, da placa. Quanto maior o raio, mais nítida a imagem, e, portanto, maior o peso atribuído.

6. DISCUSSÃO DO DESEMPENHO

As várias etapas descritas anteriormente serão agora avaliadas individualmente com relação a seu desempenho. Para isso, serão consideradas a resposta obtida por cada uma delas em vista da expectativa que se tinha considerando uma determinada entrada a ela oferecida. Essa comparação entre resposta obtida e resposta esperada pode ser avaliada qualitativamente (como no caso da detecção de pixels vermelhos) ou quantitativamente (como no caso da taxa de acerto do *template matching*).

No caso da avaliação qualitativa, por ser complicado o estabelecimento de um critério técnico rígido de aferição, as avaliações serão feitas com base nas impressões visuais perceptíveis ao avaliador. No caso das etapas suscetíveis de uma avaliação quantitativa, busca-se um índice de desempenho capaz de mensurar a taxa de acerto da resposta.

Serão, também, avaliadas as possíveis causas de discrepâncias entre as respostas obtidas e esperadas, indicando os impactos prováveis de cada uma dessas causas e, quando possível, sugerindo métodos para minimizá-las para que uma eventual continuação do trabalho possa ser direcionada.

Além das análises de cada uma das etapas quanto ao seu desempenho individual, separado do conjunto, serão avaliados os impactos daquela etapa em sua integração com as etapas seguintes. O mau desempenho de uma determinada etapa não necessariamente influenciará o desempenho de uma etapa subsequente.

Toda a discussão a respeito do desempenho dos algoritmos desenvolvidos terá sempre em vista os dois critérios apontados no início deste relatório como os mais significativos para a finalidade proposta, a saber: a velocidade de execução do algoritmo (fator preponderante para a possibilidade da sua aplicação em tempo real) e sua robustez (fator que visa garantir que, por mais eficiente que o algoritmo se comporte do ponto de vista da velocidade de execução, as respostas indicadas por ele serão confiáveis – o que é muito importante, principalmente por se tratar de um projeto cuja finalidade prática tem envolvimento íntimo com a segurança de seu usuário).

a. Sobre a captura do vídeo

Antes da avaliação das etapas do algoritmo, é importante avaliar as circunstâncias que influenciam a aquisição do vídeo, bem como as escolhas dos parâmetros de aquisição que foram feitas.

Quanto ao posicionamento da câmera no carro, avaliou-se que ao colocarmos a câmera perpendicular ao eixo do carro – e, portanto, perpendicular ao eixo da via – a captura das placas será feita de modo a evitar deformações na maior parte das imagens, já que as placas de trânsito são posicionadas perpendicularmente às vias. Uma câmera inclinada produziria um grande número de placas circulares deformadas em formas elipsoidais, o que exigiria novas etapas de processamento com a finalidade de trazer de volta a imagem ao formato circular.

Quanto à taxa de aquisição, de 15 quadros/segundo, ela garante que mesmo as placas pelas quais se passa em alta velocidade sejam capturadas em mais de um quadro. O aumento dessa taxa exigiria uma redução ainda maior na velocidade de execução.

As dimensões de 320×240 pixels do vídeo, escolhidas para esse projeto, são talvez muito pequenas, dificultando em muitos casos o reconhecimento. Nessas dimensões, as placas que aparecem nos vídeos aparecem em tamanhos que variam de 1 a 40 pixels de diâmetro, e os signos internos aparecem em tamanhos que variam de 1 a 16 pixels de altura. Abaixo de 6 pixels de altura qualquer tentativa de reconhecimento gera respostas muito incertas. Essas pequenas dimensões influenciam negativamente sobre a segurança do reconhecimento.

No entanto, o aumento dessa dimensão, diga-se, para o dobro das dimensões 640×480 , levaria a um aumento em quatro vezes do número de pixels de cada quadro, e a um aumento da velocidade nessa mesma ordem.

Como atingir a velocidade ideal foi a dificuldade maior encontrado no desenvolvimento do projeto, optou-se, portanto, por uma dimensão do vídeo inferior. À medida que a velocidade seja diminuída (com a tradução do algoritmo para uma linguagem mais veloz, com o uso de processadores mais rápidos) pode-se, futuramente, aumentar o tamanho do vídeo.

b. Etapa de detecção de pixels vermelhos

A etapa de detecção de pontos vermelhos tem como entrada o quadro capturado colorido, e como saída uma imagem binária em que os pontos reconhecidos como vermelhos são brancos enquanto todos os outros são pretos.

A aferição do nível de acerto dessa etapa só pode ser feita qualitativamente, através da observação visual do quadro original em comparação com a resposta. Não há uma maneira de sistematizar a apuração da taxa de acerto, pois para que houvesse seria necessário justamente um detector de pixels vermelhos, que é exatamente a função do algoritmo que se testa aqui.

Diante disso, a partir da comparação visual de diversos vídeos, observa-se que o índice de acerto é muitíssimo alto, dentro de determinadas condições, em especial um nível de luminosidade intermediário e do estado de conservação das placas filmadas. Em casos de luminosidade muito baixa os pontos que deveriam ser reconhecidos fogem do recorte estipulado, bem como nos casos de luminosidade muito alta. Nos casos de placas muito desgastadas, o vermelho vivo pode adquirir tons de marrom, ferrugem ou roxo, o que pode fugir do recorte em alguns casos.

No entanto, o principal ponto de crítica sobre essa etapa não se refere aos pontos vermelhos que se deixa de reconhecer, mas sim aos pontos que não deveriam ser reconhecidos como vermelhos e são reconhecidos como tais. Isso pode acontecer principalmente com tons de cinza, laranja, marrom e roxo, cuja distinção do vermelho pode ser nítida aos olhos humanos, mas é muito sutil do ponto de vista da decomposição das cores no espaço RGB.

Além de pontos com tonalidades próximas ao vermelho, há outro problema. Na imagem original pode haver outros objetos que não a placa que também são vermelhos, como cartazes, outdoors, muros, ou mesmo o próprio solo, quando se anda por estradas rurais de chão de terra. Esses pontos aparecerão na saída como se fossem vermelhos.

A presença de um excesso de pontos reconhecidos como vermelhos na imagem de saída tem um impacto no tempo de processamento da etapa seguinte, de

reconhecimento de padrões circulares, que é linearmente proporcional ao número de pontos “acessos” na imagem de entrada.

Diante dessas observações, julga-se pertinente, como sugestão para continuidade do projeto, uma maior precisão no recorte do espaço de cores, sempre tendo em vista que dependendo do espaço de cores em que se trabalha o tempo de processamento pode ser impactado. Como já foi dito ao longo do detalhamento, trabalhar em RGB é rápido pois as cores dos pontos são definidas nesta base. Trabalhar em qualquer outra base exigiria que, para cada ponto analisado, uma etapa de transposição de bases fosse executada.

Além disso, estima-se que a inclusão de uma etapa suplementar de processamento que descartasse um excesso de pontos vermelhos possa ser de boa ajuda. Apesar de essa etapa suplementar aumentar o tempo de processamento, dependendo de seu sucesso, a redução dos pontos reconhecidos como vermelhos pode reduzir de tal modo o tempo de execução da etapa seguinte que pode chegar mesmo a compensá-lo.

c. Etapa de reconhecimento de padrões circulares

A etapa de reconhecimento de padrões circulares tem como entrada uma imagem binária com os pontos vermelhos identificados pela cor branca, tem como saída intermediária uma matriz tridimensional de números reais nas mesmas dimensões da imagem de entrada, e como produto final a localização do centro do círculo reconhecido e o tamanho, em pixels, do seu raio.

Seu índice de acerto depende, sobretudo, de uma constante de calibração que estipula qual a porcentagem da quantidade de pontos de uma circunferência suficientes para indicar um círculo na saída. Se esse critério for demasiado rígido, pode-se deixar de reconhecer círculos, e se não for rígido o bastante pode-se reconhecer como círculos um conjunto de pontos que são apenas ruído.

A garantia de um bom desempenho do reconhecimento de padrões circulares vem da qualidade da imagem binária produto da etapa de detecção de pixels vermelhos. Uma imagem binária livre de ruído permite que o critério de calibração seja mais rígido, evitando erros.

O critério de calibração encontrado como ideal para o projeto foi de 50%. Para este valor, garantiu-se que todas as placas que passavam pela filmagem fossem reconhecidas em pelo menos um dos quadros que a compunham.

A etapa de reconhecimento de padrões circulares pode ser bastante impactante sob o ponto de vista do tempo de processamento. Uma sugestão para que seja otimizada é que não se busque por círculos de todos os raios possíveis. Ao invés de se procurar por círculos de raio $\{6,7,8,9,\dots\}$ pode-se aumentar o passo, buscando por círculos de raios $\{6,8,10,12,\dots\}$ ou mesmo $\{6,9,12,15,\dots\}$. O fato de as circunferências possuírem uma certa espessura garante que o aumento do passo ainda permita o reconhecimento. Quanto maior a qualidade da imagem binária de entrada desta etapa, maior a flexibilidade que se tem para aumentar o passo. Com isso se garante que o tempo de processamento seja reduzido na mesma proporção do aumento do passo.

As etapas seguintes só acontecerão no caso de um círculo ter sido reconhecido. Para os quadros em que a resposta dessa etapa seja negativa para a existência de círculos, o processamento do quadro termina e inicia-se o processamento do próximo quadro. Portanto, as etapas seguintes não dependerão da qualidade do resultado dessas etapas até o presente momento.

d. Etapa de aumento da legibilidade da placa

A partir do momento em que um círculo é reconhecido, retorna-se ao quadro original colorido como entrada desta etapa. Além dele, serão também usados como etapa os três parâmetros indicadores do círculo reconhecido (linha, coluna, raio). Como saída desta etapa, será produzida uma imagem em níveis de cinza em que a região no entorno da placa, incluindo-a, terão sua legibilidade aumentada por uma normalização do histograma da região.

Há dois parâmetros de calibração para a normalização do histograma, referentes à qual porcentagem da faixa do histograma existente será saturada em branco e a qual porcentagem da faixa será saturada em preto.

A mudança desses dois parâmetros afeta a legibilidade final da placa. A saturação de porcentagens muito pequenas da faixa de histograma pode gerar uma imagem de saída em muitos tons de cinza, o que, para o método de reconhecimento de

algarismos escolhido (*template matching*) seria muito danoso, uma vez que este é sensível a contraste e a ruído. Por outro lado, a saturação de porcentagens muito extensas dessa faixa pode ser também igualmente danosa, podendo mesmo deformar os signos.

Sucessivos testes com vários valores para estes parâmetros foram realizados, chegando-se aos valores de 38% da faixa inferior do histograma sendo saturada em preto, e 48% da faixa superior sendo saturada em branco, deixando, portanto, 14% da faixa central do histograma para ter seus novos valores determinados por interpolação.

Um possível problema, não verificado nos vídeos testados, mas possível em teoria, é o de a escolha dos limites de saturação da faixa surtir efeitos pouco perceptíveis. Isso aconteceria, por exemplo, no caso da presença, na área analisada, de um ponto com intensidade luminosa muito superior ou muito inferior aos outros, o que distorceria o histograma. Para evitar esse problema, um método de cálculo sistematizado dos parâmetros seria sugerido. Esse método poderia se basear, por exemplo, na determinação dos limites das faixas de saturação com base na porcentagem de pontos acima ou abaixo daquele limite. Uma segunda sugestão para essa sistematização seria um método de desconsiderar esses pontos discrepantes.

Por se tratar de um algoritmo bastante simples aplicado a uma área muito pequena (levando-se em conta que o maior círculo encontrado nas filmagens tinha raio igual a 17 pixels, a área sobre a qual a normalização do histograma ocorreria se restringiria a um trecho quadrado da imagem de dimensões 34×34), o tempo de processamento dedicado a esta etapa tem pouco impacto no tempo de processamento total.

Vale ressaltar também o fato de que esta etapa do algoritmo se faz necessária em vista de uma restrição do método de reconhecimento de padrões escolhido. Caso o reconhecimento fosse feito baseado na correlação entre duas imagens, por exemplo, ao invés do *template matching*, a etapa de aumento da legibilidade seria dispensável.

Quanto ao impacto da qualidade da saída desta etapa na etapa seguinte de segmentação da região analisada em áreas correspondentes a diferentes algarismos, é importante notar que a boa escolha do parâmetro de saturação do histograma, sobretudo do parâmetro que determina a saturação em preto, é determinante. Uma

má escolha levaria a uma saturação em preto dos pixels situados entre os algarismos de uma placa, o que comprometeria seriamente o acerto da segmentação.

e. Etapa de segmentação da região por algarismos

A etapa de segmentação tem como entrada uma imagem em níveis de cinza com a legibilidade da região da placa adaptada, e gera como saída uma lista de “retângulos” (posições de linha e coluna superiores e inferiores, totalizando quatro parâmetros) que delimitam a região onde se encontra cada algarismo separadamente.

A aferição de uma taxa de acertos é possível, ao se checar visualmente uma grande quantidade de segmentações e verificar se a segmentação foi bem procedida. Com base nisso foram checadadas as segmentação de 300 diferentes quadros, todos eles contendo placas devidamente identificadas e com o histograma normalizado. A taxa de acerto foi de 251 num universo de 300, ou seja, temos uma taxa de acerto na segmentação de 83.7%.

Cabe, portanto, questionar as razões que levariam 16.3% das segmentações a falharem. As causas verificadas, por ordem de frequência com que ocorrem, são as seguintes:

- A segmentação aconteceu em um dos extremos (direita ou esquerda) da placa. Isso pode ser facilmente explicável retomando o critério de segmentação, que buscava coluna cuja soma de todos seus pontos possuísse o a maior derivada segunda (ou seja, a maior variação *preto-branco-preto* num intervalo de três colunas). Isso ocorre porque nas extremidades essa variação também é muito importante, e por vezes chega a ser superior àquela da região que separa os dois algarismos. Para solucionar esse problema sugere-se que se excluam as extremidades das possibilidades de segmentação.
- Por vezes a imagem perde o foco, e embora isso não seja tão determinante no sucesso da etapa de identificação e localização do círculo, isso pode impedir qualquer possibilidade de reconhecimento.
- Em terceiro lugar, porém em menor frequência, é possível que a saída da etapa de aumento da legibilidade tenha saturado em preto os pontos que deveriam separar os dois algarismos, impedindo a correta segmentação.

O impacto dos erros nesta etapa na etapa seguinte, de *template matching*, é muito grande. Se retomarmos, como exemplo, o resultado do reconhecimento de uma placa de 30 km/h documentado previamente no tópico de detalhamento deste mesmo relatório, perceberemos que o reconhecimento se dá por uma pequena margem. Assim, qualquer erro nesta etapa de segmentação acrescentaria uma perturbação capaz de alterar o resultado do reconhecimento. Essa constatação recomenda grande prudência ao se avaliar o desempenho desta etapa, e sugere-se que quaisquer erros ali presentes sejam corrigidos.

Quanto ao impacto no tempo de processamento, analogamente à observação feita para a etapa de aumento da legibilidade da região da placa, por se tratar de uma região muito pequena da imagem sobre a qual estamos trabalhando estima-se que o impacto sobre o tempo total seja muito pequeno.

Outra observação que vale ser feita é a de que o método como foi concebido é eficiente apenas para imagens de tamanho reduzido, uma vez que os algarismos que aparecem nos vídeos não superam os 16 pixels de altura. Isso se deve ao fato de que a segmentação depende de um abrupto decréscimo na soma das intensidades dos pixels de uma coluna. Em uma imagem de grandes dimensões, a separação entre os dois algarismos seria de várias colunas, e não só de uma. Portanto, a idéia só se torna aplicável a vídeos de dimensões superiores tendo em vista que adaptações serão necessárias.

f. Etapa de reconhecimento dos algarismos

A etapa de reconhecimento dos algarismos por *template matching* tem como entradas uma imagem em níveis de cinza com a região da placa destacada e legível, e um conjunto de retângulos indicando as regiões correspondentes a cada algarismo que deverá ser reconhecido. Como saída, é gerada uma tabela indicativa do nível de semelhança entre a imagem de entrada e cada um dos templates previamente integrados à base de templates. A partir desta tabela, que é gerada para cada quadro, é possível determinar como saída final desta etapa qual é o template que mais se assemelha a cada um dos algarismos da entrada, determinando, assim, qual é a provável placa filmada.

É possível determinar uma taxa de acerto para esta etapa por meio da comparação entre uma entrada e a resposta obtida, da mesma maneira como se procedeu durante a aferição da taxa de acerto da etapa anterior. Para garantir a credibilidade da resposta, é preciso que as entradas oferecidas ao *template matching* garantam as condições de legibilidade dos algarismos, correta segmentação e correta correspondência entre a escala da imagem e a escala do template comparado a ela.

Retomando a advertência feita durante o tópico anterior, qualquer quadro não obedecendo aos critérios mencionados pode alterar os resultados de toda a sequência de quadros que integra, pois o *template matching* é bastante sensível. Diante disso, para fins da aferição dos resultados do *template matching* foram excluídos os quadros que não atendiam às especificações.

Para que o levantamento dessa taxa de acertos fosse confiável seria necessário um extenso número de vídeos de entrada, todos eles funcionais. Diante da dificuldade de gravar tal quantidade de vídeos, o teste foi feito usando-se, além dos vídeos gravados na rua, vídeos gravados em casa em que placas desenhadas simulavam a passagem de uma placa de trânsito pela câmera.

Desse modo, em um universo de 200 amostras (ou seja, 400 algarismos a serem reconhecidos), a taxa de acertos do *template matching* obtida considerando os algarismos dos quadros individualmente foi de 57.4% (o que significa que, caso a taxa de acerto se mantenha estável para todos os algarismos, a probabilidade de acerto dos dois algarismos de uma placa simultaneamente pode ser estimada em 32.9%).

O levantamento da taxa de acerto quando se consideravam os vários quadros que compõem uma placa no vídeo aumenta a taxa de acerto para 81.9% dos algarismos, desde que não haja nenhum quadro errado. Com base nesse valor, estima-se que a chance de acerto de dois algarismos simultaneamente é de 67.1%, evidenciando que a tomada em consideração de vários quadros aumenta significativamente a chance de acerto.

Os fatores preponderantes para que taxa de acerto não tenha sido superior são a grande quantidade de placas em tamanho pequeno, o que dificulta a diferenciação, a possível variação tipográfica da placa, e a sutileza nas diferenças entre os algarismos. Certos algarismos possuem uma taxa de acerto superior aos outros, como é o caso do

algarismo "0" ou do algarismo "1", cujos formatos são muito particulares. É interessante notar, por exemplo, que a taxa de acerto do algarismo "0" é de 92.6%, medidos num universo de mais de 100 amostras, o que é muito superior aos 57.4% de expectativa de acerto quando a entrada é desconhecida. Já no caso do reconhecimento de algarismos como "6" e "5", que se assemelham muito entre si, essa taxa situa-se abaixo dos 57.4% (considerando, em ambos exemplos, o reconhecimento de um algarismo individualmente).

Uma possível melhora para a robustez do algoritmo seria conseguida trabalhando-se com vídeos em dimensões superiores aos 320×240 pixels adotados. No entanto, como já se alertou, o aumento nas dimensões acarretaria uma grave perda de velocidade.

No entanto, na medida em que se consiga reduzir o tempo de execução do algoritmo como um todo, é aconselhável que, para fins de garantia de um resultado acertado, dedique-se boa dose de atenção a aprimoramentos nesta etapa, como na adoção de novos critérios auxiliares de aferição do resultado possível a partir do conjunto de tabelas obtido. Uma vez que a segurança do usuário de um equipamento desenvolvido com base neste projeto está envolvida, é especialmente importante, para um eventual prosseguimento, que se aumente a robustez do sistema.

7. ANDAMENTO DO PROJETO E METAS ALCANÇADAS

Para a primeira fase do projeto, já iniciada em de seu primeiro semestre e concluída no início do segundo semestre, esperava-se como resultado o conhecimento das características de diversas técnicas aplicadas à Análise e Processamento de Imagens. Buscava-se a compreensão de como essas técnicas poderiam ser aplicadas na solução do problema proposto, e da maneira como a sua utilização afetaria o tempo de execução do programa e sua robustez. Como produto final, foi traçada a estratégia de abordagem da metodologia que será usada para a criação do algoritmo.

À medida que se fazia necessário um retorno à pesquisa de novas alternativas para a resolução de novos problemas que ocorriam no decorrer do projeto era feito o retorno às tarefas executadas durante essa primeira fase. Além disso, em certos momentos, quando as técnicas e métodos estudados não ofereciam as melhores soluções para o problema a ser resolvido, a solução era buscada por meio da criação de uma nova abordagem para o problema.

Para a segunda fase do projeto, a idéia inicialmente era de buscar como resultado um programa que resolvesse o problema proposto para um quadro fixo. A partir do momento em que o algoritmo desenvolvido estivesse funcional, ele seria aplicado para um vídeo, e seu desempenho no tocante à velocidade de execução e à taxa de acerto seria avaliada. O que se verificou ao longo do desenvolvimento do projeto é que havia uma grande perda na eficiência do algoritmo na comparação entre como ele funcionava com um quadro fixo e, posteriormente, com os sucessivos quadros de um vídeo. Isso deriva principalmente da qualidade da imagem com que se trabalha. Na aquisição de uma imagem fixa, sua qualidade é muito superior à dos quadros de um vídeo. Isso exigiu que o algoritmo fosse recalibrado, e que até mesmo a metodologia traçada na fase de planejamento fosse reformulada. Diante disso foi necessário rever a abordagem pensada anteriormente, e chegou-se a conclusão de que não valeria a pena do ponto de vista do tempo disponível para a conclusão do projeto continuar trabalhando com quadros fixos antes de iniciar o trabalho com vídeo. O produto final da segunda fase continuou sendo o mesmo, um algoritmo funcional para vídeos ainda por ser otimizado, mas as atividades planejadas foram alteradas diante das dificuldades experimentadas na prática.

O que se notou durante o desenvolvimento desta segunda fase do projeto foi que a inexperiência em projetos similares acabou por se refletir na elaboração de um planejamento que não contribuía para o desenvolvimento do trabalho em tempo hábil. Parte dessa expectativa positiva advém do fato de, ainda durante o primeiro semestre do desenvolvimento do projeto, testes terem sido feitos em fotografias de dimensões superiores às das imagens dos vídeos que viriam a ser usados. Diversos problemas que pareceram simples quando se trabalhava com fotografias de dimensões maiores revelavam-se extremamente sutis e complexos ao se trabalhar com vídeos. Esse primeiro contato produziu uma impressão otimista sobre o desenvolvimento do projeto, que acabou por não se concretizar.

Na terceira fase, prevista para ser desenvolvida após a conclusão da segunda, o algoritmo funcional resultante da segunda fase deveria ser otimizado até que se atendessem a todos os requisitos propostos nos objetivos. Nesse momento do projeto, esperava-se que fosse necessário abandonar etapas de processamento de imagem julgadas periféricas e suplementares na medida em que os resultados pretendidos quanto a rapidez não fossem atingidos, ou, inversamente, cogitava-se acrescentar etapas de processamento complementares na medida em que o algoritmo não se mostrasse robusto o bastante. Como resultado, esperava-se um algoritmo com boa relação entre velocidade e robustez, pronto para ser testado para um vídeo adquirido em tempo real.

No entanto, à medida que as dificuldades da segunda fase acumulavam-se, e que se buscavam soluções para resolvê-las, os questionamentos próprios à terceira fase, sobre o impacto de cada etapa do algoritmo no tempo de execução e na taxa de acerto do reconhecimento, passaram a ser pensados desde antes do início previsto para esta terceira fase. De fato, ao invés de se desenvolverem diversas diferentes técnicas simples para a resolução de um mesmo problema, permitindo que se pudesse escolher quais delas fossem convenientes apenas ao final de acordo com seu impacto, esse impacto já era avaliado durante a segunda fase, evitando assim a perda de tempo com a implementação de um algoritmo que não viesse a ser aproveitado.

É bem verdade que, por vezes, nem mesmo com as precauções tomadas se foi capaz de antever certos resultados negativos, mas o fato de exercer sobre cada novo passo implementado na segunda fase um olhar crítico num nível de detalhe que só se planejava dispensar durante a terceira fase evitou muita perda de tempo.

À quarta fase do projeto competiria utilizar o algoritmo desenvolvido na prática, em um carro em movimento, e avaliar se este é de fato capaz de reconhecer as placas de trânsito na medida em que aparecessem. Na ocorrência de não-conformidades, o algoritmo deveria ser revisto segundo a mesma abordagem tomada durante a terceira fase do projeto, até que se atingisse um nível mínimo de qualidade capaz de garantir que a solução seria segura para uso. Como resultado final dessa quarta e última fase, esperava-se um algoritmo que, enfim, reconheça placas de trânsito em tempo real.

Porém, não se pôde alcançar a quarta fase do projeto a tempo. A grande quantidade de pequenas dificuldades aparecidas durante a segunda fase⁰ consumiu muito mais tempo do que se contava *a priori*.

Como metas alcançadas do projeto, tem-se um algoritmo funcional para vídeos *offline*, capaz de reconhecer placas de trânsito indicativas de limite de velocidade. No entanto, os critérios de velocidade e de taxa de acerto ainda não satisfazem as expectativas que se tinha ao início do projeto.

Quanto às metas por cumprir, temos o trabalho de otimização do algoritmo para garantir uma maior taxa de acertos, mediante a correção dos diversos pequenos detalhes problemáticos de cada etapa de processamento que, ao se acumularem, geram discrepâncias ao final – sobretudo em se tratado a última etapa de um *template matching*, que é particularmente sensível a pequenas variações.

Além dele, uma meta ainda por cumprir seria a redução da velocidade a níveis que permitissem o emprego do algoritmo para reconhecimento em tempo real. Esse objetivo poderia ser obtido mediante a melhora na qualidade das várias etapas de processamento, a avaliação de cada operação de acordo com sua real importância para o resultado final e, por fim, mediante a tradução do algoritmo escrito em C++ fazendo uso da biblioteca ProEikon para outra linguagem mais veloz.

Os cronogramas que resumem as atividades cumpridas ao longo dos dois semestres em comparação com as atividades previstas podem ser vistos a seguir.

a. Cronograma do projeto

O cronograma que resume as atividades desenvolvidas ao longo do primeiro semestre do projeto pode ser visto abaixo:

Semana	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18
Definição do tema	✓	✓	✓	✓	✓													
Fase 1: Aquisição do Vídeos para Teste						✓											✓	✓
Fase 1: Estudo das técnicas existentes						✓	✓	✓										
Fase 1: Avaliação do desempenho das técnicas									✓	✓	✓	✓						
Fase 1: Definição das técnicas a serem usadas												✓	✓	✓	✓	✓		
Fase 2: Algoritmo Funcional para um Quadro															✓	✓	✓	✓
Revisão do Cronograma para o 2º semestre															✓	✓		
Preparação da apresentação final																	✓	✓

Fig. 60 – Cronograma das atividades desenvolvidas ao longo do primeiro semestre de projeto, com as atividades previstas indicadas em cinza, e as atividades cumpridas indicadas com um sinal afirmativo (✓).

O cronograma referente ao segundo semestre do projeto é visto a seguir:

Semana	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Fase 2: Criação da Base para Template Matching	✓	✓														
Fase 2: Algoritmo Funcional para um Quadro	✓	✓														
Fase 2: Algoritmo Funcional para um Vídeo			✓	✓	✓	✓	✓									
Fase 3: Otimização da Velocidade							✓	✓	✓	✓						
Fase 3: Otimização da Robustez							✓	✓	✓							
Fase 4: Algoritmo Funcional em Tempo Real																
Fase 4: Algoritmo Otimizado em Tempo Real																
Preparação da Monografia Final												✓	✓	✓	✓	
Entrega da Monografia Final															✓	
Apresentação Final																✓
Protótipo Funcional e Pôster Final																✓

Fig. 61 – Cronograma das atividades desenvolvidas ao longo do segundo semestre de projeto, com as atividades previstas indicadas em cinza, e as atividades cumpridas indicadas com um sinal afirmativo (✓).

8. GESTÃO DO PROJETO

A partir do momento em que se iniciou a etapa prática do projeto aqui descrito, notou-se desde o princípio a existência de dificuldades não previstas (como, por exemplo, a dificuldade acima assinalada na adaptação de um algoritmo já desenvolvido que para uma imagem fixa era funcional e para um vídeo se mostrou insuficiente, a ponto de ser em grande parte revisto).

Diante dessa primeira não-conformidade entre a estratégia elaborada na teoria e sua execução na prática, é necessário fazer uma análise das possíveis falhas que podem vir a aparecer no projeto no decorrer das semanas de sua conclusão. Além de citar algumas delas, é preciso estimar o impacto no desenvolvimento do projeto, bem como possíveis abordagens para mitigar os riscos.

a. Gestão dos materiais

Do ponto de vista material e técnico o projeto cumpriu a previsão de que seria viável. Para sua realização foram usados softwares e bibliotecas gratuitos, o que não representa um risco do ponto de vista financeiro. Além deles, foi usada a biblioteca particular gentilmente cedida pelo orientador desse projeto, o prof. Hae Yong Kim. O projeto passaria a apresentar gastos apenas em seus objetivos secundários, como, por exemplo, na elaboração de uma interface com o usuário. No entanto, esses objetivos secundários não constituem o centro de interesse do projeto, que é obter um algoritmo capaz de realizar a identificação de placas de trânsito a partir de métodos de processamento de imagens.

b. Gestão do tempo

Quanto à gestão do tempo, havia-se previsto que o projeto pudesse, sim, enfrentar obstáculos que comprometessem seu cumprimento pleno – como de fato ocorreu. Nem sempre as estratégias traçadas foram aplicáveis ao objeto a que se destinavam, pois é possível que se tenha partido de hipóteses incondizentes com a realidade. Para mitigar o impacto dessas situações que poderiam demandar um longo tempo de reestruturação estratégica contou-se com o auxílio do orientador do projeto, capaz de prover esclarecimentos em situações de impasse.

Num primeiro momento não apareceram situações de grande impacto, e estas puderam ser resolvidas mediante reestudo. No entanto, nas grandes transições de fase do projeto (ida de imagem para vídeo, tentativa de ida de vídeo para tempo real, otimização do algoritmo) percebeu-se que aquilo que era uma possibilidade de ocorrência de imprevistos se realizou. E isto se deu por vezes de forma tão impactante que nem mesmo a revisão que se havia feito do cronograma do projeto adiantando algumas de suas atividades, numa abordagem conservadora, conseguiu garantir que seus objetivos propostos sejam alcançados.

9. CONCLUSÕES

Os objetivos traçados para o projeto contemplaram uma variedade de técnicas que foram desenvolvidas ao longo das últimas décadas para a Análise e Processamento de Imagens, tenham elas sido usadas como foram concebidas ou como idéia-base de uma variação. As dificuldades previstas para o projeto se concretizaram, e apresentaram-se até mesmo mais desafiadoras do que se havia imaginado na fase de planejamento das atividades. Essas dificuldades dizem respeito principalmente à qualidade e à confiabilidade das respostas obtidas por essas técnicas, principalmente por se posicionarem diante de problemas que exigem um alto grau de segurança para a resposta. O grande desafio do projeto residiu em tentar superar essas dificuldades, que de fato foram vencidas em grande número, embora muitos pontos ainda tenham sido deixados em aberto para eventuais continuações. O algoritmo ainda apresenta-se lento para ser utilizado em tempo real e insuficientemente seguro para servir, no estado atual, a um propósito prático. Tendo em vista a necessidade de se garantir uma grande qualidade no que diz respeito à velocidade e à robustez do algoritmo, novos passos de otimização do algoritmo seriam necessários. O trabalho produzido ao longo dos dois semestres de duração deste projeto resultou em um algoritmo capaz de reconhecer, com boa dose de acerto, uma placa de trânsito indicativa de limite máximo de velocidade. Conseguindo-se, dentro do contexto de um prosseguimento deste projeto, realizar as otimizações necessárias a satisfazer os objetivos traçados para o projeto, teremos em mãos um algoritmo funcional, que, num segundo momento, poderia muito bem ser empregado em pequenos dispositivos acopláveis a um veículo.

BIBLIOGRAFIA

- [1] R. C. Gonzalez and R.E. Woods, "Digital Image Processing", Reading, MA, 1992 (reprinted 1993).
- [2] Duda, R. O. and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures" Comm. ACM, Vol. 15, pp. 11-15 (January, 1972)
- [3] Lowe, David G. (1999). "Object recognition from local scale-invariant features", Proceedings of the International Conference on Computer Vision 2: 1150-1157.
- [4] [Ballard1981] D. H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," Pattern Recognition, vol. 13, no. 2, pp. 111-122, 1981.

APÊNDICE A

Exemplo de resposta de um filtro que aponta a presença de pontos predominantemente vermelhos da figura original, variando-se o limite mínimo aceitável para a componente vermelha R (considerando o espaço de cores RGB) para um dado pixel.



Fig. 62 - Imagem Original



Fig. 63 - Rmin = 100



Fig. 64 - Rmin = 105



Fig. 65 - Rmin = 110



Fig. 66 - Rmin = 115



Fig. 67 - Rmin = 120



Fig. 68 - Rmin = 125



Fig. 69 - Rmin = 130

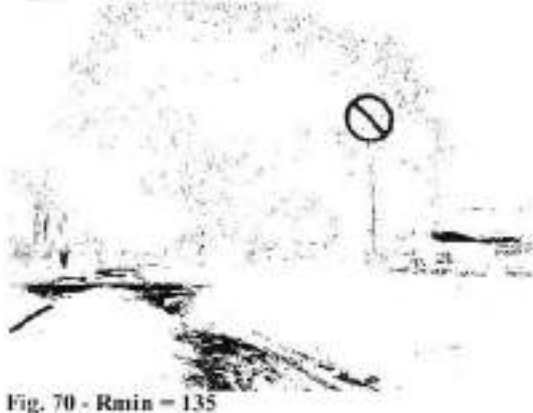


Fig. 70 - Rmin = 135

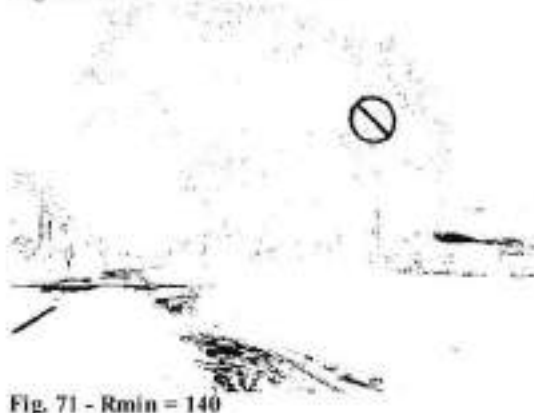


Fig. 71 - Rmin = 140



Fig. 72 - Rmin = 145



Fig. 73 - Rmin = 150



Fig. 74 - Rmin = 155

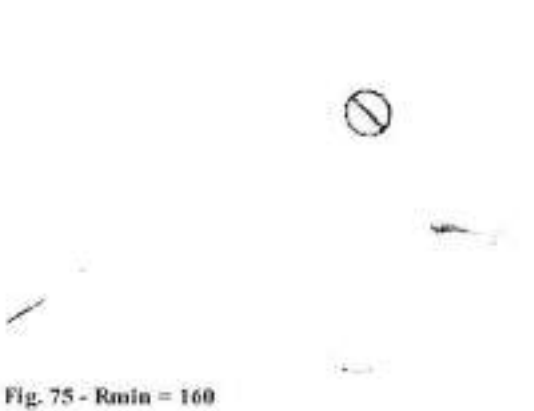


Fig. 75 - Rmin = 160

APÊNDICE B

Exemplo de detecção de círculos por meio de uma generalização da transformada de Hough, para raios variando de 33 a 27 pixels. Notamos nas últimas três imagens que foram detectados círculos de raios 27, 28 e 29 pixels. A detecção de um círculo da imagem original é mostrada na forma de um ponto escuro concentrado na figura de resposta da transformada de Hough.



Fig. 76 - Imagem Original



Fig. 77 - Hough para Raio = 33



Fig. 78 - Hough para Raio = 32



Fig. 79 - Hough para Raio = 31



Fig. 80 - Hough para Raio = 30



Fig. 81 - Hough para Raio = 29



Fig. 82 - Hough para Raio = 28



Fig. 83 - Hough para Raio = 27

