

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS,
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
APLICADA - PIPCA
MESTRADO

MÁRCIO JUNIOR CAPPELLARI

**Detecção de Tráfego Rodoviário Utilizando Visão
Computacional: Um Algoritmo de Apoio ao Motorista**

SÃO LEOPOLDO
2010

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

xxx Cappellari, Márcio Junior

Detecção de Tráfego Rodoviário Utilizando Visão Computacional: Um Algoritmo de Apoio ao Motorista / Márcio Junior Cappellari. — 2010.

87 f.: il. ; 30cm.

Proposta (mestrado) — Universidade do Vale do Rio dos Sinos - UNISINOS. Ciências Exatas e Tecnológicas, Programa Interdisciplinar de Pós-Graduação em Computação Aplicada - PIPCA, São Leopoldo, BR-RS, 2010. Orientador: de Oliveira, Luiz Paulo Luna.

1. Visão Computacional. 2. Processamento de Imagens. 3. Detecção de Obstáculos. 4. Sistema de Apoio ao Motorista. I. Título.

CDU xxx

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS

Reitor: Dr. Marcelo Fernandes de Aquino

Vice-Reitor: Dr. Jos Ivo Follmann

Pró-Reitor Acadêmico: Dr. Pedro Gilberto Gomes

Diretor de Pós-Graduação e Pesquisa: Dr. Alsones Balestrin

Coordenador do PIPCA: Prof. Dr. Arthur Tórgo Gómez

AGRADECIMENTOS

Faço estes agradecimentos não somente para registro daquelas tantas pessoas que colaboraram na realização deste trabalho, mas o faço como quem faz uma singela oração, desejando poder retribuir em algum tempo a ajuda recebida e aprendendo muito com cada um. Muitas pessoas fizeram parte desta caminhada, agradeço especialmente ao Professor Dr. Cláudio Rosito Jung, que me apresentou uma linha de pesquisa, totalmente nova e fascinante, diferente de tudo que havia visto. Foi ele quem apoiou, orientou e que em momentos difíceis da minha jornada foi compreensivo e sempre trouxe uma palavra de confiança. Ao Professor Dr. Luis Paulo Luna de Oliveira, agradeço por aceitar assumir a orientação deste trabalho em fase final e que trouxe consigo o direcionamento reto, objetivo e positivo para a conclusão deste. Que soube fazer as perguntas certas e que dividindo o trabalho em pequenas tarefas formou este resultado final. Agradeço ao meu colega e amigo Dario Scott, a quem admiro muito pela espírito de luta e cujo exemplo me motiva. Aos meus colegas de Laboratório, em especial ao Leandro Dihl, pelo caminho já trilhado, José Carlos Bins e Alessandro Parolin pelo convívio e pelas dicas em C/C++ e OpenCV. Obrigado aos professores do Programa de Pós-Graduação em Computação Aplicada da Unisinos, em especial aos professores Drs. João Carlos Gluz e João Francisco Valiati pelas sugestões e considerações em meu relatório de andamento do mestrado. Obrigado ao professor Dr. Arthur Tórgo Gómez, coordenador do programa, por permitir a extensão do prazo deste trabalho em um momento que os problemas pessoais não permitiam a concentração e clareza necessária para o bom andamento desta dissertação. Agradeço ao SICREDI na pessoa de Juliano Murlick, que incentivou a realização deste mestrado e que flexibilizou a jornada de trabalho sempre que necessário para as atividades acadêmicas. Por fim, agradeço ao colega de trabalho Tiago Minuzzi, pela ajuda incondicional na construção deste documento, pelas inúmeras sugestões e contribuições que foram sempre bem vindas.

Dedico este trabalho à minha companheira Patricia, sempre a meu lado e aos meus pais
Gládis e Doralino, exemplos perpétuos em minha vida.

“A diferença entre VENCER e PERDER é quase sempre ... PERSISTIR.”

(Walt Disney)

“O melhor meio de prever o futuro é inventá-lo.”

(Alan Kay)

RESUMO

A preocupação com a segurança no trânsito é tão antiga quanto a história do automóvel e muitos são os esforços das montadoras, dos órgãos públicos e de pesquisa, visando diminuir o número de acidentes e de vítimas do trânsito. Muitos dos acidentes que acontecem são atribuídos a falha humana dos motoristas, que por imprudência e/ou imperícia, não conseguem perceber obstáculos a tempo de evitar uma colisão. Entenda-se por obstáculo outro veículo, um pedestre na pista, e até mesmo uma árvore, animal ou qualquer objeto que obstrua a passagem do condutor e que poderá causar um acidente. De fato, este trabalho está focado na identificação de outros veículos.

O presente trabalho apresenta um algoritmo capaz de detectar obstáculos na pista por visão computacional. Trata-se de um veículo equipado com uma câmera monocular embarcada, com processamento e identificação de obstáculos em tempo real, apoiando o motorista sobre a presença destes obstáculos no campo de visão da câmera, e sobre a aproximação destes com risco de colisão. Outros sensores, como radar, infra-vermelho, sonar poderiam apoiar na detecção de obstáculos, porém, é premissa deste estudo, desenvolver o algoritmo utilizando recursos de baixo custo e focada no processamento de imagens.

Inicialmente, procurar-se-á a delimitação da região de busca por obstáculos, também chamada de região de interesse, através da detecção das bordas da pista. Na sequência o detector trabalhará na geração de hipóteses (HG), com a identificação de candidatos a obstáculos, para sobre eles processar a etapa de verificação da hipótese e assim confirmar ou negar a presença de obstáculos reais. São considerados atributos da imagem como cor/intensidade, simetria, quinas, bordas, linhas horizontais e verticais, e calibração de câmera. Além disso, treinou-se um classificador de cascata considerando um conjunto de características Haar.

Palavras-chave: Visão Computacional, Processamento de Imagens, Detecção de Obstáculos, Sistema de Apoio ao Motorista.

TITLE: “Real-Time Traffic Detection using Computer Vision: an algorithm to support the driver”

ABSTRACT

The concern for traffic safety is as old as the automobile history, and many are the efforts of carmakers, public agencies and research in order to decrease the number of accidents and victims of traffic accidents. Many of the accidents that happen are attributed to human failure. Because of reckless driving and / or malpractice, they can not see obstacles with enough time to avoid a collision. There are many types of obstacles: a vehicle, a pedestrian, a tree, or even an animal. Any object that obstructs the passage of the driver can cause an accident. This work is focused on identifying only other vehicles as obstacles.

This work presents an algorithm capable of detecting an obstacle on the track by computer vision. The project uses a vehicle equipped with a monocular camera, for processing and identification of obstacles in real time, supporting the driver about the presence of them on the road, and alerting him about collision risks. Other sensors, such as radar, infrared, or sonar could assist in obstacle detecting, however, the premise of this study is to develop an algorithm using low-cost resources and focused on image processing.

Initially, we will start with the delineation of the region for obstacles search, also called the region of interest (ROI), by detecting the runway lanes. Next, the detector will work on hypothesis generation (HG), identifying candidates for obstacles, and then processing them on the hypothesis verification stage, to confirm or deny the presence of real obstacles. The main attributes considered are image color / intensity, symmetry, edges, borders, horizontal and vertical lines and camera calibration. Also, using Haar Like Features, the classifier cascade was trained.

Keywords: Image Processing, Computer Vision, Obstacle Detection.

SUMÁRIO

RESUMO	6
ABSTRACT	7
LISTA DE FIGURAS	10
LISTA DE ABREVIATURAS	12
1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	13
1.2 OBJETIVOS	15
1.3 ESTRUTURA DO TRABALHO	15
2 CONCEITOS BÁSICOS	16
2.1 VISÃO COMPUTACIONAL	16
2.2 OPERADORES DE BORDAS	19
2.3 IDENTIFICADOR DE RETA	22
2.4 IDENTIFICADOR DE QUINAS	24
2.5 PADRÕES DO TIPO HAAR	25
2.6 COMPONENTES CONEXOS	29
2.7 OPERADORES MORFOLÓGICOS	30
2.8 CALIBRAGEM DE CÂMERA	32
3 REVISÃO BIBLIOGRÁFICA	34
3.1 GERAÇÃO DE HIPÓTESES	34
3.1.1 Base de conhecimento	34
3.1.1.1 Simetria	34
3.1.1.2 Cor	34
3.1.1.3 Bordas Horizontais e Verticais	35
3.1.1.4 Quina	35
3.1.2 Fluxo Óptico	36
3.2 VALIDAÇÃO DE HIPÓTESES	36
3.2.1 Validação Baseada em Modelo	36
3.2.2 Validação Baseada em Aparência	39
4 METODOLOGIA PROPOSTA	41
4.1 MÓDULO 1: DELIMITAÇÃO DA REGIÃO DE INTERESSE	42
4.2 MÓDULO 2: IDENTIFICAÇÃO DO PADRÃO DE PISTA	48
4.3 MÓDULO 3: GERAÇÃO DE HIPÓTESES	52
4.4 MÓDULO 4: VALIDAÇÃO DAS HIPÓTESES	54
4.5 PERCORRENDO O ALGORITMO	56
5 RESULTADOS	59
5.1 MÓDULO 1: DELIMITAÇÃO DA REGIÃO DE INTERESSE	59

5.2	MÓDULO 2: IDENTIFICAÇÃO DO PADRÃO DE PISTA	61
5.3	MÓDULO 3: GERAÇÃO DE HIPÓTESES	65
5.4	MÓDULO 4: VALIDAÇÃO DAS HIPÓTESES	66
5.5	APLICAÇÃO DE <i>HAAR LIKE FEATURES</i>	68
6	ESTADO DA ARTE	75
7	CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	79
7.1	CONSIDERAÇÕES FINAIS	79
7.2	TRABALHOS FUTUROS	80
	BIBLIOGRAFIA	82

LISTA DE FIGURAS

Figura 1.1	Exemplo Sensores (JONES, 2002)	14
Figura 2.1	Olho Humano	17
Figura 2.2	Analogia entre visão humana e computacional	17
Figura 2.3	Representação da imagem no Computador (BRADSKIE ADRIAN, 2008)	18
Figura 2.4	A mesma imagem 3D muda radicalmente na representação 2D (BRADSKIE ADRIAN, 2008)	19
Figura 2.5	Resultados do detector de bordas Canny: (a) Imagem original. (b)(c)(d) Resultados obtidos pela variação de parâmetros	23
Figura 2.6	Exemplos de quinas	24
Figura 2.7	Estágios do Classificador	25
Figura 2.8	Tipos de Haar Like Features (FACE... ,)	26
Figura 2.9	Calculo da Imagem Integral	27
Figura 2.10	Componentes Básicos do Modelo de Viola e Jones	27
Figura 2.11	Identificação de Componente Conexo	30
Figura 2.12	Operadores Morfológico: (a) Erosão. (b)Dilatação (MORPHOLOGY,)	31
Figura 2.13	Esquerda: Estrada em coordenadas de Mundo. Direita: Estrada em coordenadas de imagem.	32
Figura 3.1	Transformada Hough evidenciando as retas de uma imagem	35
Figura 3.2	Fluxo Óptico da Imagem	37
Figura 3.3	Proporção entre largura da pista e do veículo	38
Figura 3.4	Distância entre faróis e outros veículos (a) imagem destacando os faróis (b) Componentes conexos identificados	39
Figura 4.1	Calibragem da Câmera	42
Figura 4.2	Fluxo Lógico de Desenvolvimento	43
Figura 4.3	Região de interesse da imagem	44
Figura 4.4	Divisão da pista em campos para detecção de bordas da pista	45
Figura 4.5	Relação entre sistemas de coordenadas	47
Figura 4.6	Etapas ROI: (a) Imagem original. (b) Etapa de identificação das Bordas da Pista (c) Delimitando a região de interesse	48
Figura 4.7	Marcações da pista sendo classificadas como candidato a obstáculo .	52
Figura 4.8	Etapas da identificação do padrão de pista: (a) Identificação inicial de pista e não pista (b) Filtro das marcações da pista	53
Figura 4.9	Avaliação do tamanho do componente conexo: (a) <i>Blob</i> selecionado (b) Identificação de retas	55

Figura 4.10	Projeção de sobras sobre a pista	55
Figura 4.11	Projeção de sobras sobre a pista	56
Figura 4.12	Limite do Componente Conexo	57
Figura 4.13	Parte do veículo fora da região de interesse	58
Figura 5.1	Identificação da bordas da Pista	60
Figura 5.2	Resultado da Calibragem	61
Figura 5.3	Identificação da região de interesse	61
Figura 5.4	Identificação de não pista nos canais RGB e <i>Grayscale</i>	63
Figura 5.5	Identificação de Não Pista	63
Figura 5.6	Divisão da pista em blocos horizontais	64
Figura 5.7	Diferente classificação de pixel para blocos vizinhos	65
Figura 5.8	Classificação de pista e não pista	65
Figura 5.9	Classificação de pixels como não pista - falso positivo	66
Figura 5.10	Delimitação do componente conexo	66
Figura 5.11	Teste de descarte de estruturas pequenas na imagem	67
Figura 5.12	Detecção de Retas: (a) Componente Conexo (b) Detector de Bordas de Canny (c) Detector de linhas de Hough	68
Figura 5.13	Descarte de sombra: (a) Componente Conexo (b) Detector de Bordas de Canny (c) Nenhuma reta detectada	68
Figura 5.14	Amostras positivas para o treinamento: (a)Imagem inteira (b) Metade inferior da Imagem	69
Figura 5.15	Teste com árvore de decisão treinada	70
Figura 5.16	Teste em imagem manipulada	72
Figura 5.17	Detecções em Pista	72
Figura 5.18	Sequencia de cenas	74
Figura 6.1	Controle de Cruzeiro Adaptativo da Audi	76

LISTA DE ABREVIATURAS

RGB	Sistema de cores aditivas formado por Vermelho (<i>Red</i>), Verde (<i>Green</i>) e Azul (<i>Blue</i>)
FPS	Quadros por segundo (<i>Frames per second</i>)
HG	Hypothesis Generation
HV	Hypothesis Verification
SISAMO	Sistema de Apoio ao Motorista
MAD	Median Absolute Deviation
PIPCA	Programa de Computação Aplicada da Universidade do Vale do Rio dos Sinos
Daer	Departamento Autônomo de Estradas de Rodagem
Renaest	Registro Nacional de Acidentes e Estatística de Trânsito
Denatran	Departamento Nacional de Trânsito
GM	General Motors
CES	Consumer Electronics Show
LBP	Local Binary Pattern
PCA	Principal Component Analysis
LOC	Local Orientation Code
ROI	Region of Interest

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Desde sua concepção, o automóvel traz internalizado o objetivo de proporcionar aos ocupantes um meio de transporte que, agregando tecnologias, visa ser rápido, confortável e seguro. A popularização do automóvel trouxe consigo grande avanço e, em contrapartida, gerou novos problemas.

No Brasil em 2007, ocorreram 116.962 acidentes com vítimas, gerando um universo de 151.186 vítimas não fatais e 6.885 vítimas fatais (RENAEST, 2009). Em dezembro de 2007, a frota brasileira totalizava em torno de 50 milhões de veículos e mais de 42 milhões de condutores habilitados.

Somente no estado do RS o número de acidentes passou de 4.588 em 1991 para 10.090 em 2008 (DAER, 2009). Quanto aos tipos de acidentes apurados em 2008, os principais tipos são abalroamento, choque e colisão, que representam 77 % dos acidentes registrados conforme a Tabela 1.1.

Tabela 1.1: Principais tipos de acidentes

Tipos de Acidentes	2008
Abalroamento	3.375
Choque	2.572
Colisão	1.849
Capotamento	682
Tombamento	643
Não Apurado	373
Atropelamento de Pessoa	341
Atropelamento de Animal	255
Total	10.090

A condução autônoma significa que, algum dia, você poderia revisar seu e-mail, tomar café da manhã, maquiarse e assistir um vídeo enquanto dirige-se para o trabalho, explicou Rick Wagoner diretor da GM (*General Motors*). Segundo Wagoner, em apresentação no CES (*Consumer Electronics Show*), em Las Vegas, em janeiro de 2008, o futuro dos automóveis é brilhante e cada vez mais eletrônico. Wagoner acrescenta que uma das principais metas é que os automóveis se comuniquem entre si. Os carros poderiam assim receber sinais de outros veículos e utilizar controles computadorizados para adotar ações como desacelerar para evitar uma colisão.

Enquanto a comunicação entre veículos não se torna uma realidade, deve-se adotar a detecção de obstáculos através de sensores embarcados. Em termos práticos, tem-se um exemplo de um Chevrolet Tahoe transformado, capaz de se dirigir sozinho, vencendo

um desafio do Departamento de Defesa americano, manobrando em meio ao tráfego de uma cidade. Em termos acadêmicos, é possível identificar uma série de propostas de identificação de obstáculos, incluindo-se pedestres e outros veículos, a partir do uso de sensores passivos ou ativos (ver Capítulo 3). Enquanto um sensor passivo não atua de forma intrusiva para a aquisição de dados, um sensor ativo é intrusivo, pela emissão de algum sinal que é refletido pelo objeto (SUN et al., 2006). Sensores como radares, lasers, infra-vermelhos são exemplos de sensores ativos, enquanto que câmeras são classificados como sensores passivos. Exemplo de um veículo com vários sensores pode ser observado na Figura 1.1. A presença destes sensores encarecem o valor total da solução.

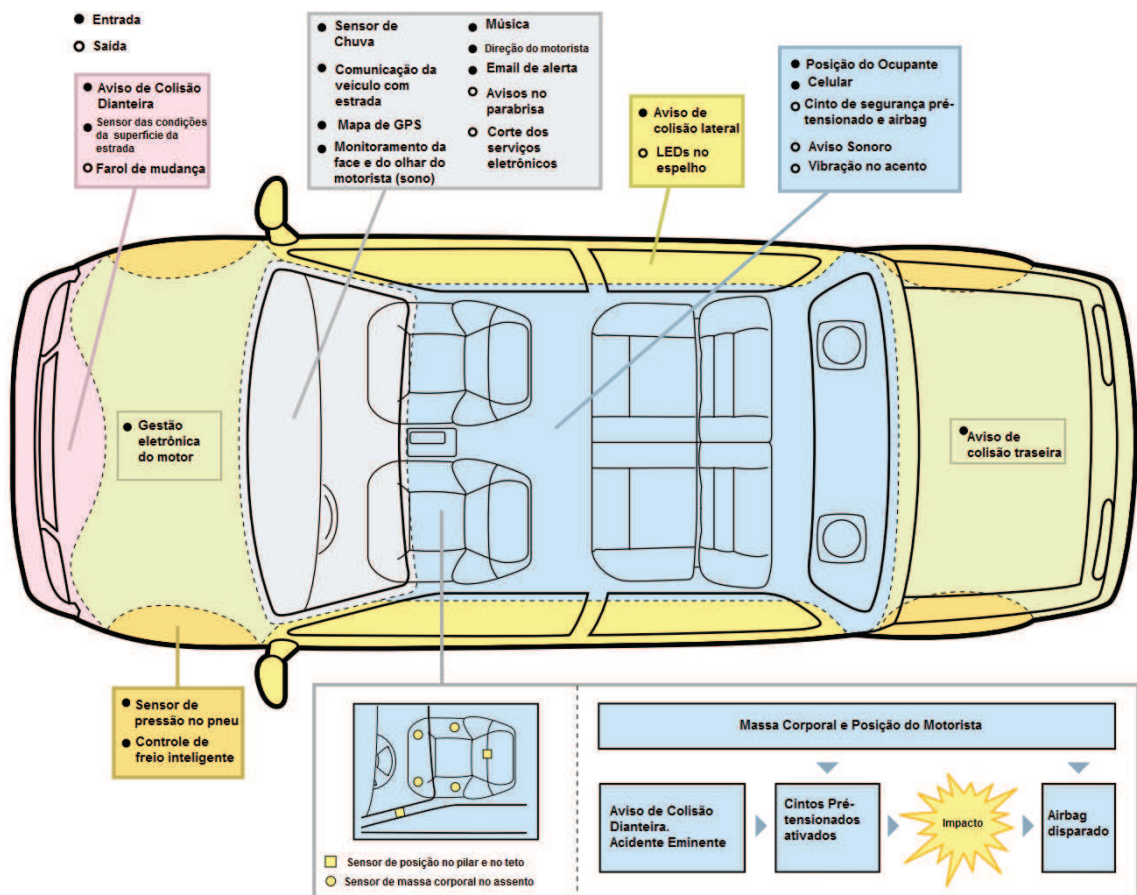


Figura 1.1: Exemplo Sensores (JONES, 2002)

A concepção de um sistema automotivo embarcado de sensores, como os referidos acima, capaz de identificar o ambiente e as possibilidades de colisão como um todo, é um grande desafio e ainda não é suficientemente simples para se tornar de uso comum. Além disso, encareceria demasiadamente o veículo. Entretanto, grande parte dos acidentes graves ocorre em tráfego não urbano, devido aos choques entre automóveis em auto estradas. Desta forma, desenvolver um sistema mais simples, restritamente capaz de detectar a presença de outros veículos na pista, interagindo com o condutor, alertando-o, ajudaria a diminuir as estatísticas de acidentes graves, aumentando a segurança em

viagens de uma maneira geral. Por isto, o objetivo deste trabalho é propor um sistema de detecção de tráfego em auto estradas, usando visão computacional, isto é, a partir de identificação de padrões em imagens obtidas de uma simples câmera embarcada. Para um automóvel trafegando numa pista, o tipo mais comum e mais perigoso de obstáculo, é justamente outro automóvel na pista (na mão ou na contra-mão). Portanto, neste trabalho, os obstáculos considerados são os automóveis numa mesma pista, que doravante, serão referidos simplesmente como obstáculos.

1.2 OBJETIVOS

O objetivo principal deste trabalho é desenvolver um algoritmo de visão computacional para a detecção de tráfego através de uma única câmera monocular embarcada em um veículo.

Dentre os objetivos específicos estão:

- Simplicidade de funcionamento;
- Alta sensibilidade de detecção;
- Baixo número de falsos positivos;
- Identificação de obstáculos em tempo real.

1.3 ESTRUTURA DO TRABALHO

No Capítulo 2 são referenciadas as informações básicas sobre a área de estudo em que o projeto foi realizado. O Capítulo 3 é a visita realizada a estudos relacionados a este trabalho. Traz o embasamento teórico e faz apontamentos sobre técnicas e algoritmos que foram aplicados no trabalho para atingir o objetivo traçado. O Capítulo 4 apresenta o modelo do algoritmo desenvolvido. É detalhado o fluxo do algoritmo, bem como, os aspectos relevantes que compuseram a solução. O capítulo 5 relata os resultados obtidos após a implementação do algoritmo, detalhando os experimentos realizados e avaliando a performance geral da solução. Uma avaliação sobre o estudo da arte, comparando as abordagens e os resultados obtidos por este trabalho com o mercado e com outras pesquisas serão tratados no capítulo 6 Finalizando no capítulo 7, em que são descritas as conclusões e recomendações para trabalhos futuros.

2 CONCEITOS BÁSICOS

Sun et al. (2006) destacam alguns aspectos que representam os grandes desafios para a identificação de veículos a partir do processamento de imagens. Afinal, o veículo pode assumir diversas formas, tamanhos e cores. Além disso, a aparência de um veículo depende do ângulo de visão e é afetada por outros objetos próximos a este veículo. Como se não bastassem as dificuldades inerentes à própria identificação dos veículos, Sun como? cita os elementos complicadores em um ambiente externo, como iluminação, tráfego e a necessidade de um processamento rápido. É necessário identificar um obstáculo a tempo para a tomada de uma ação preventiva, ou seja, antes da possível intersecção com o obstáculo. Uma solução confiável, deverá alertar o motorista sobre a presença de pedestres, sobre obstáculos, e até mesmo alertar sobre uma possível saída da pista. Este tipo de informação só gerará segurança se for robusta, real e em tempo para evitar um acidente.

Alguns conceitos básicos são importantes para a contextualização e entendimento do trabalho. Este capítulo trata dos principais. Na Seção 2.1 serão discutidos os conceitos de Visão Computacional, na Seção 2.2 serão tratados os conceitos de Operador de Borda, na Seção 2.3 o assunto é Identificador de Reta. Na Seção 2.4 serão discutidos identificadores de quina, a Seção 2.5 abordará os padrões do tipo Haar e por fim, as Seções 2.6, 2.7, 2.8 abordam os temas de componentes conexos, operadores morfológicos e calibragem de câmera respectivamente.

2.1 VISÃO COMPUTACIONAL

Segundo Pedrini e Schwartz (2007), “a *visão computacional* procura auxiliar a resolução de problemas altamente complexos, buscando imitar a cognição humana e a habilidade do ser humano em tomar decisões de acordo com as informações contidas na imagem”. Nixon e Aguado (NIXON e AGUADO, 2008) fazem a analogia de que, enquanto o cérebro processa imagens obtidas através do olho no sistema de visão humano, um sistema de visão computacional processa imagens obtidas a partir de uma câmera eletrônica.

A Figura 2.1 ilustra as partes que compõem o olho humano.

Com efeito, a luz é capturada pelo sistema ocular humano e transmitida ao cérebro pelo nervo óptico. Este, por sua vez, reconhece os padrões geométricos das imagens capturadas, primeiramente selecionando os aspectos mais relevantes da imagem, para depois comparar com padrões conhecidos. O foco da visão é utilizado para restringir parte do todo da imagem vista, para focarmos naquilo que é de real interesse.

O desafio está no desenvolvimento de soluções computacionais autônomas capazes de reagir a estímulos visuais. Para Pedrini e Schwartz (2007), o objetivo principal de um sistema de visão computacional é obter uma descrição que contenha informações

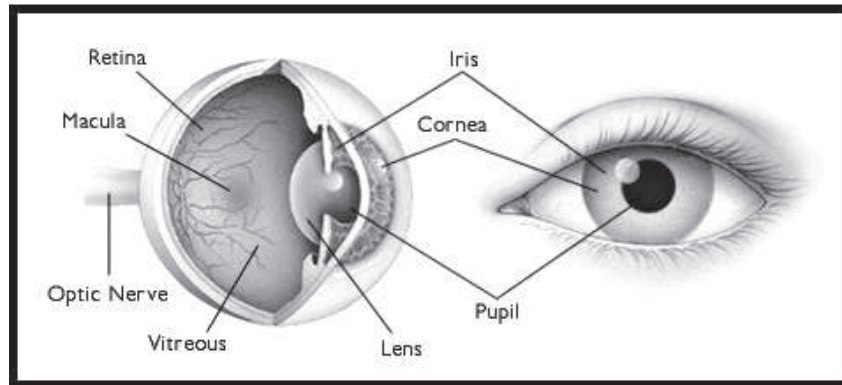


Figura 2.1: Olho Humano

suficientes para distinguir entre diferentes objetos de interesse, de forma confiável e requerendo o mínimo de intervenção humana.

Visão computacional é a transformação de dados, obtidos de uma foto ou vídeo digital, em uma decisão (exemplo: há um obstáculo na pista) ou nova representação (por exemplo: a conversão de uma imagem colorida em tons de cinza) (BRADSKI e ADRIAN, 2008). Cada transformação tem um objetivo específico. No contexto deste trabalho, o objetivo é identificar a presença de obstáculos na pista.

A Figura 2.2 faz um paralelo entre a visão humana e computacional. Na visão humana a imagem é capturada pelo olho, interpretada pelo cérebro que decide e comanda o corpo. No caso da visão computacional, a imagem é capturada por uma câmera, processada e interpretada pelo computador, que comanda um braço mecânico.



Figura 2.2: Analogia entre visão humana e computacional

Processamento de imagem é qualquer forma de processamento de dados no qual a entrada e saída são imagens, tais como fotografias ou quadros de vídeo. Ao contrário do tratamento de imagens, que se preocupa somente na manipulação de figuras para sua representação final, o processamento de imagens é um estágio para novos processamentos de dados, tais como aprendizagem de máquina ou reconhecimento de padrões. A maioria das técnicas envolve o tratamento da imagem como um sinal bi-dimensional, no qual são aplicados padrões de processamento de sinal (PEDRINI e SCHWARTZ, 2007).

O presente trabalho considera, como entrada, a imagem obtida a partir de uma câmera monocular. Seu processamento é o ponto focal do algoritmo, visando a identificação dos obstáculos. O objeto de saída deste processamento é o alerta ao motorista e a manipulação da imagem com o realce do obstáculo.

Tanto a visão humana quanto a computacional, convergem para o objetivo de interpretar um dado espacial, dado este que é indexado por mais que uma dimensão. A visão humana não foi totalmente compreendida em como o sinal obtido pelo olho é passado ao cérebro e como é processado por este. Como ilustrado na Figura 2.3, para o computador, a imagem não passa de uma grade de números de uma grande matriz, entretanto existe relação entre estes números e o mundo real.

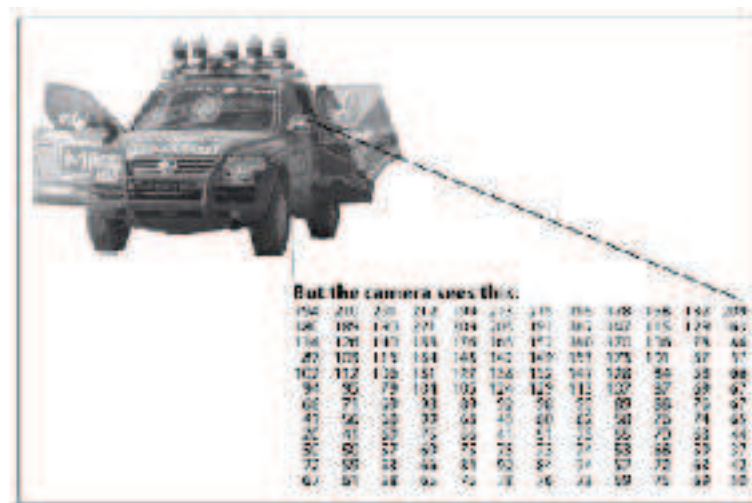


Figura 2.3: Representação da imagem no Computador (BRADSKI e ADRIAN, 2008)

Uma imagem é uma maneira de representar algo tridimensional em duas dimensões por uma matriz. Para Bradski e Adrian (2008), esta representação numérica em duas dimensões “*pior que difícil; isto é formalmente impossível de resolver*”. De fato, dada uma informação bidimensional (2D), não há apenas um meio de reconstruir uma informação tridimensional (3D) validamente associada. Em outras palavras, a mesma informação 2D de uma imagem pode representar uma de infinitas combinações de cenas em 3D conforme ilustrado na Figura 2.4. Ou seja, apesar do objeto capturado ser exatamente o mesmo, a mudança do ângulo de visão modifica totalmente o matriz numérica que a representa.

A *imagem digital* é uma representação da imagem através de um sinal digital. Pedrini e Schwartz (2007) descrevem o processo de digitalização em dois passos, chamados de amostragem e digitalização. O primeiro passo consiste em discretizar o domínio de definição da imagem nas direções x e y , gerando uma matriz de $M \times N$ amostras. Cada elemento desta matriz é chamado de pixel. No passo da quantização, cada um dos pixels recebe um valor inteiro que representa a imagem em L níveis de tons de cinza, no caso de imagem monocromática.

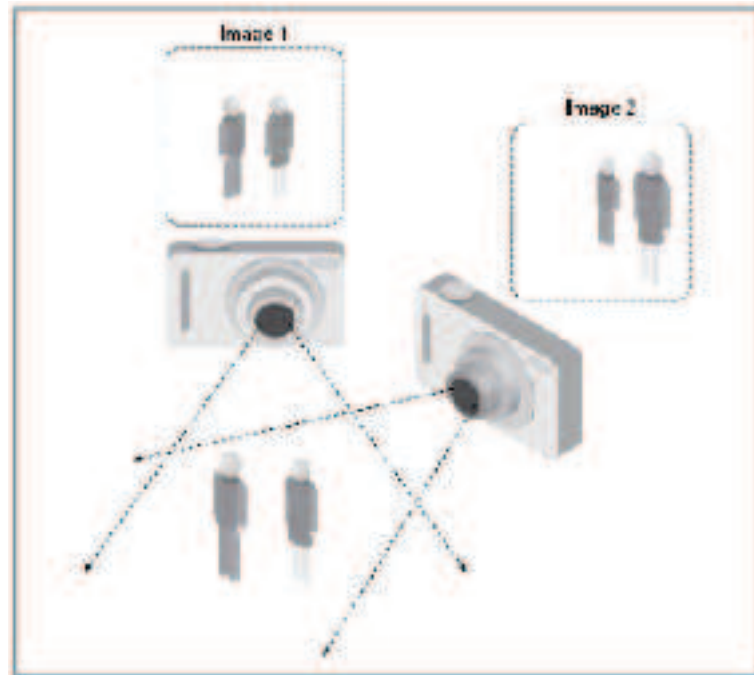


Figura 2.4: A mesma imagem 3D muda radicalmente na representação 2D (BRADSKIE ADRIAN, 2008)

A partir de uma imagem digital, é possível extrair uma série de características, que podem ser úteis para a detecção de estruturas específicas. A seguir, algumas dessas características que são comumente exploradas em processamento de imagens, serão brevemente descritas. Também serão abordadas algumas técnicas que exploram essas características para a detecção de obstáculos a partir de imagens.

2.2 OPERADORES DE BORDAS

Pedrini e Schwartz (PEDRINI e SCHWARTZ, 2007) definem borda como o limite ou a fronteira entre duas regiões com propriedades relativamente distintas de nível de cinza.

A detecção de bordas é, essencialmente, a operação de identificação de locais com alteração significativas nos níveis de cinza. É o processo de localização e realce dos pixels de borda, aumentando o contraste. Como uma borda é definida por uma mudança no nível de cinza, quando ocorre uma descontinuidade na intensidade, ou quando o gradiente da imagem tem uma variação abrupta, um operador que é sensível a estas mudanças operará como um detector de bordas. Pedrini e Schwartz (2007) destacam que uma mudança em intensidade pode ser detectada pela diferença entre valores de pixels adjacentes. Bordas *verticais* podem ser detectadas pela diferença *horizontal* entre pontos, enquanto bordas *horizontais* são percebidas pela diferença *vertical* entre pontos adjacentes da imagem.

Para Vale e Poz (2002), computacionalmente, define-se uma borda como sendo uma região que possui um gradiente de alta magnitude. O detector de Sobel é um dos mais

populares e uma interessante forma de se realizar a detecção de bordas, visto que este operador é baseado no cálculo do gradiente e sua magnitude. Segundo Seara (1998), o cálculo do gradiente é dado por:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.1)$$

E sua magnitude é dada por:

$$mag(\nabla f) = [G_x^2 + G_y^2]^{\frac{1}{2}} \quad (2.2)$$

Onde:

$$G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{ e } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Além de Sobel há uma série de operadores e algoritmos já conhecidos para a detecção de bordas: Operador de Prewitt (1970), Operador de Kirsch (1971), Operador de Canny (1986).

Para o desenvolvimento do algoritmo deste trabalho foi utilizado o Operador de Canny, considerado por muitos o detector de bordas ótimo e que tem comportamento destacado no tratamento de imagens com ruído, que é muito frequente em imagens em movimento. Tal operador foi desenvolvido por John F. Canny em 1986. É um operador gaussiano de primeira derivada que suaviza os ruídos e localiza as bordas. Este processo de localização/detecção de bordas deve atender a três critérios básicos especificados (CANNY, 1986):

1. Taxa de Erro ou boa Detecção: é a maximização da razão sinal/ruído (RSR). Representa a eficiência do filtro em relação a falsos positivos. Canny define o calculo de RSR pela Equação:

$$RSR = \left(\frac{\left| \int_{-w}^w G(-x)f(x)dx \right|}{n_0 \sqrt{\int_{-w}^w f^2(x)dx}} \right), \quad (2.3)$$

onde $f(x)$ é a resposta de impulso do filtro definido no intervalo $[-w; w]$, $G(x)$ é uma borda unidimensional n_0 a quantificação do ruído da imagem. Assume-se que

a borda está centrada em $x = 0$. Para (CANNY, 1986), quanto maior for o RSR, maior será a probabilidade de detectar as bordas da imagem e melhor a detecção;

2. Boa Localização: é a premissa de que a borda detectada deve estar o mais próxima possível das posições verdadeiras. É mensurado como sendo o inverso da distância entre um ponto detectado e a posição física da borda. Desta forma, quanto maior for a função Localização (L), mais próximos das posições reais estarão os pontos detectados;

Canny propôs a seguinte Equação:

$$L = \left(\frac{\left| \int_{-w}^w G'(-x) f'(x) dx \right|}{n_0 \sqrt{\int_{-w}^w f'^2(x) dx}} \right), \quad (2.4)$$

Para atender ao critério da localização, é utilizado o processo de histerese ou limiarização adaptativa (VALE et al., 2004). Este processo consiste em aplicar duas vezes o processo de limiarização em uma mesma imagem, com a utilização de dois limiares, T_1 e T_2 , onde $T_1 = 2T_2$ ou $T_1 = 3T_2$. A limiarização se dá da seguinte forma:

- (a) Verificar se a luminância do pixel é maior que T_1 . Se for, ele é automaticamente aceito como borda;
 - (b) Se a luminância do pixel for menor que T_2 , este pixel será automaticamente rejeitado como borda;
 - (c) Se o valor da luminância estiver entre os dois limiares, este será aceito somente se os seus pixels imediatamente vizinhos, forem pixels com alto valor de luminância.
3. Resposta Mínima: O último dos critérios corresponde à quantidade de respostas geradas pelo detector em relação a uma única borda. Para cada borda existente na imagem, o filtro deve apontar somente uma resposta. Considerando a Equação:

$$x_{max} = 2\pi \left(\frac{\int_{-\infty}^{\infty} f'^2(x) dx}{\int_{-\infty}^{\infty} f''^2(x) dx} \right)^{1/2}, \quad (2.5)$$

a expressão matemática para a distância (x_{max}) entre máximos adjacentes na resposta do filtro $f(x)$ devido ao ruído. Assim, ao maximizar as condições dadas pelas Equações 2.3 e 2.4, deve-se também garantir que x_{max} seja o maior possível, aumentando a possibilidade de separação de máximos verdadeiros dos falsos na saída do filtro $f(x)$. Via de regra, as bordas são representadas por pixels que possuem gradiente de alta magnitude, mas que sofrem com a presença de ruído na imagem.

Através do terceiro critério, os pixels de magnitude não máxima são eliminados, restando 1 pixel que corresponde a 1 ponto de borda.

Em síntese, o detector de bordas de Canny é dado pelo encadeamento das seguintes etapas:

1. Redução de ruído: através da convolução da imagem com uma máscara gaussiana é produzida uma versão ligeiramente desfocada da imagem original, porém, isto reduz o efeito dos pixels ruidosos para os passos seguintes;
2. Cálculo dos gradientes da intensidade da imagem: são consideradas quatro máscaras para detectar bordas verticais, horizontais e diagonais (45 e 135 graus). É executada a convolução da imagem original com cada uma das máscaras;
3. Criação de Mapas: para cada pixel, é identificado o maior resultado do gradiente nesse pixel, e o sentido da máscara que produziu essa borda. Assim, são criados dois mapas, um para a intensidade e o outro para o sentido do gradiente;
4. Detecção de bordas: apesar de os gradientes com maior intensidade terem probabilidade maior de serem bordas, não há um valor absoluto que confirme esta hipótese. Para isto, são utilizados dois limiares, um superior e outro inferior. O limiar superior visa mapear as bordas mais destacadas e, a partir destas, o limiar inferior acompanha o sentido das bordas previamente identificadas. O objetivo é seguir linhas mesmo que de baixa intensidade, pois as bordas tendem a ser linhas contínuas.

O resultado é uma imagem binária que consiste de borda e não borda.

A variação dos parâmetros influencia diretamente no tempo de processamento e na qualidade da detecção da borda, devendo ser ajustada aos objetivos da aplicação e ao padrão de imagem. A Figura 2.5 demonstra a variação do resultado do detector com diferentes parâmetros (SIMULADOR..., 2010).

2.3 IDENTIFICADOR DE RETA

A partir do momento em que se tem as bordas da imagem definidas, algumas características são passíveis de serem investigadas. Um problema comum na análise de imagens é a detecção de um conjunto de pontos de uma imagem, que pertencem a uma curva específica, tal como a segmentação de circunferência, elipse e reta. Pedrini e Schwartz (2007) enfatizam que estas curvas constituem uma família representada assim:

$$f(v, p) = 0 \tag{2.6}$$

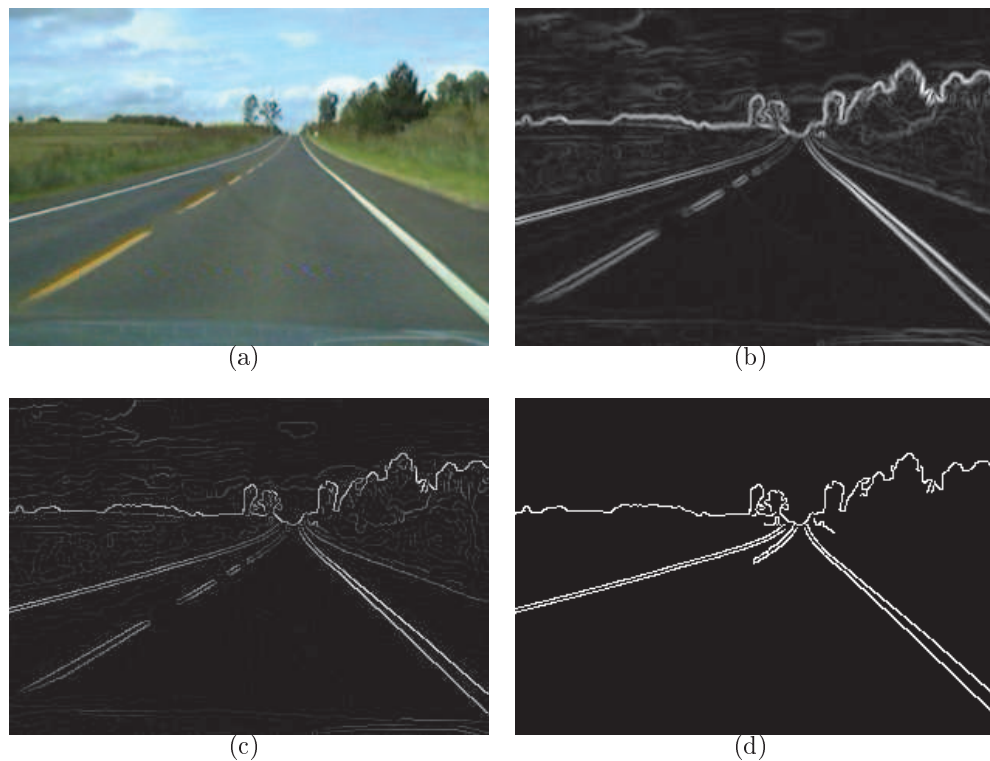


Figura 2.5: Resultados do detector de bordas Canny: (a) Imagem original. (b)(c)(d) Resultados obtidos pela variação de parâmetros

em que v é um vetor de coordenadas e p é um vetor de parâmetros característicos da curva. De posse de um conjunto de pontos $p_i(x_i, y_i)$, o próximo passo é identificar se este conjunto pertence a uma determinada curva. Em se tratando de uma reta, o desafio é encontrar um subconjunto de pontos que sejam colineares. Pedrini e Schwartz (2007) citam que, uma possível solução seria encontrar todos os segmentos de reta formados entre cada par de pontos e buscar os subconjuntos de pontos que estejam próximos dos segmentos de retas. Entretanto, o custo computacional para esta abordagem é elevado.

Uma alternativa é o uso da Transformada de Hough, desenvolvida por Paul Hough (1962). O método desenvolvido por Hough consiste em detectar pontos colineares ou quase colineares numa imagem. Cada borda de uma imagem é transformada pelo mapeamento para determinar células no espaço de parâmetros, indicadas pelas primitivas definidas através do ponto analisado. Essas células são incrementadas, e indicarão, no final do processo, através da máxima local do acumulador, quais os parâmetros correspondentes à forma especificada (exemplo reta).

Bones e Bruno (2007) explicam que uma reta pode ser definida por dois parâmetros, utilizando-se coordenadas polares (p, θ) , onde p indica a distância mínima da reta à origem do plano cartesiano e, θ o coeficiente angular que o segmento de reta r faz com o eixo x

das ordenadas. Os pontos desta reta podem ser representados por:

$$p = x\cos(\theta) + y\sin(\theta) \quad (2.7)$$

2.4 IDENTIFICADOR DE QUINAS

Também chamados de junção ou canto, a quina pode ser definida com um ponto de intersecção entre dois ou mais segmentos da borda de objetos (PEDRINI e SCHWARTZ, 2007). A sua identificação fornece informações importantes para várias aplicações envolvendo análise de imagens, como a localização de pontos homólogos em visão estereoscópicas, detecção de movimento, rastreamento de objetos e restauração de imagens. As junções podem indicar também, a presença de objetos, facilitando a tarefa de interpretação da imagem.

A Figura 2.6 representa os tipos mais comuns de quintas que são: L, Y, T, Seta, e X.

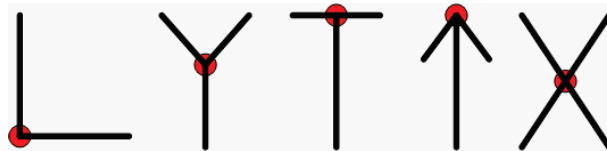


Figura 2.6: Exemplos de quinas

Quanto às formas de detecções de quinas, Pedrini e Schwartz (PEDRINI e SCHWARTZ, 2007) comentam que os métodos existentes, em geral, são baseados na identificação de pontos de inflexão ou de alta curvatura ao longo dos contornos extraídos dos objetos (bordas).

Existem vários operadores já modelados para a detecção de quinas. Um dos primeiros foi o Moravec (1977) (MORAVEC, 1980). Pedrini e Schwartz (2007) resumem o modelo: a variação de intensidade para cada pixel da imagem é calculada como a soma dos quadrados das diferenças entre os pixels adjacentes, localizados em cada uma das quatro direções (horizontal, vertical, e duas diagonais) sobre uma vizinhança centrada no pixel. A variação mínima entre esses quatro valores é tomada como uma medida de interesse. Para avaliar a presença da quina, o seguinte teste é realizado: se essa medida for superior a um limiar, o pixel será considerado junção ou canto da imagem; caso contrário, o pixel é descartado.

Outros operadores são: Beudet (1978), Harris (1987) (HARRIS e STEPHENS, 1988), Noble (1988), Quddus e Fahmy (1999), Susan (SMITH e BRADY, 1995).

2.5 PADRÕES DO TIPO HAAR

Este recurso tem sido bastante explorado na detecção de objetos, com destaque para a identificação da face humana. A própria biblioteca de algoritmos OpenCV (BRADSKIE ADRIAN, 2008) utilizada durante o estudo, está capacitada a identificar faces nativamente.

Um processo de reconhecimento pode ser muito mais eficaz, se baseado na detecção de características que codificam algumas informações sobre a classe a ser detectada, ou seja, é necessário reconhecer uma assinatura para o objetivo. Este é o caso das *Haar like features*, que codificam a existência de um objeto pela orientação de contrastes entre as regiões da imagem. *Haar like features* são assim chamados porque são computados semelhante aos coeficientes da transformada *wavelet* (WIKIPEDIA...).

Haar Like Features consiste de vários classificadores simples (fases), que são aplicadas na sequência de uma região de interesse até que, em algum momento, o candidato é rejeitado ou todas as etapas são percorridas e o objeto é identificado. A aplicação destas características se dá a partir de um algoritmo de Boosting. *Boosting pode ser implementado facilmente a partir de um classificador razoável, transformando-o em um comitê de classificação com desempenho muito superior ao classificador original. Boosting funciona combinando sequencialmente classificadores simples, dando maior peso em cada passo as observações classificadas incorretamente no passo anterior* (RUBESAME DIAS, 2004). A Figura 2.7 representa este processo de percorrer vários classificadores até a identificação do objeto pesquisado.

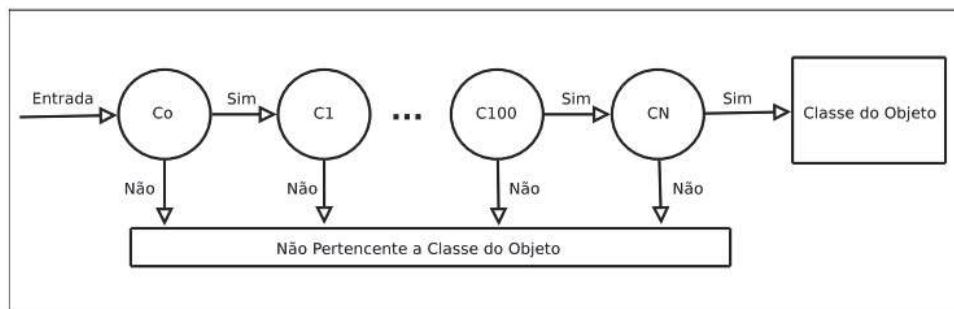


Figura 2.7: Estágios do Classificador

A ideia básica do algoritmo proposto por Viola e Jones (2004) é deslizar uma janela (W) na imagem para procurar os candidatos à face, o que se aplicaria exatamente da mesma forma para a procura de veículo. Para isso, os autores utilizam um conjunto de atributos *haar-like features* que pode ser visualizado na Figura 2.8. Esse conjunto de atributos codificam as diferenças médias de intensidade nas regiões retangulares vizinhas da imagem. É uma técnica de detecção através de padrões geométricos. Uma forma retangular simples *Haar-like feature* pode ser definida da diferença entre a diferença da soma dos valores dos pixels das áreas dentro do retângulo.

Para que tal cálculo seja rápido é necessário a aplicação da Integral da Imagem dada

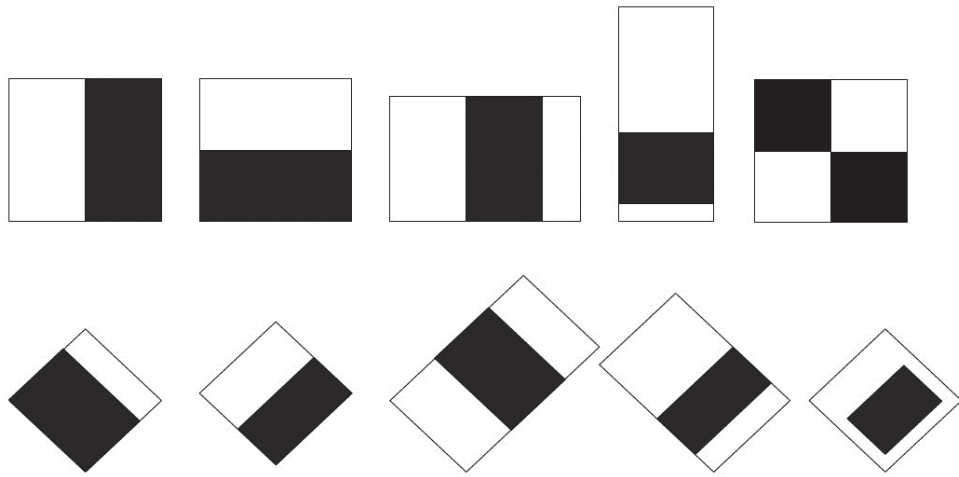


Figura 2.8: Tipos de Haar Like Features (FACE...,)

pela Equação:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x, y) \quad (2.8)$$

Onde $i(x, y)$ é o valor atual do pixel da imagem e $ii(x, y)$ é o valor integral da imagem em (x, y) .

A Figura 2.9 é a demonstração deste cálculo, contendo o somatório de qualquer região retangular da imagem original. Para Viola e Jones (2001), a vantagem na utilização desses atributos é que eles conseguem representar pequenas estruturas sem se ater ao detalhe, sendo menos sensível ao ruído e tolerando melhor pequenas variações na luminosidade. A soma dos pixels do retângulo D, pode ser computado pelos quatro vetores de referência. O valor da imagem integral (*integral image*) em P1 é a soma dos pixels do retângulo A. O valor da localização P2 é dado por A + B, a localização P3 por A + C, e a localização P4 é dada por A + B + C + D. A somatória dos pixels de D podem ser computados por P4 + P1 - (P2 + P3) (FACE...,)

A Figura 2.10 demonstra os componentes básicos do modelo. Primeiramente baseia-se no uso de características Haar básicas, avaliadas rapidamente por meio de uma nova forma de representação da imagem chamada Imagem Integral. Logo em seguida, é gerado um conjunto de características. Por meio da aplicação de um algoritmo de aprendizado *boosting*, neste trabalho o AdaBoost, filtra-se o conjunto no intuito de diminuir o número de características geradas. Por fim, ocorre a geração de uma árvore dos classificadores fracos *boosted*, para que gere inferências rápidas e robustas na construção de um classificador forte (VIOLAe JONES, 2001).

A implementação de *Haar Like Features* se desdobrou em uma série de algoritmos *Boosting*, sendo destacado para o presente trabalho o AdaBoost. O AdaBoost (do inglês, *Adaptive Boosting*) é um algoritmo genérico de aprendizagem de máquinas para construção de classificadores complexos a partir de classificadores simples. Eleito no

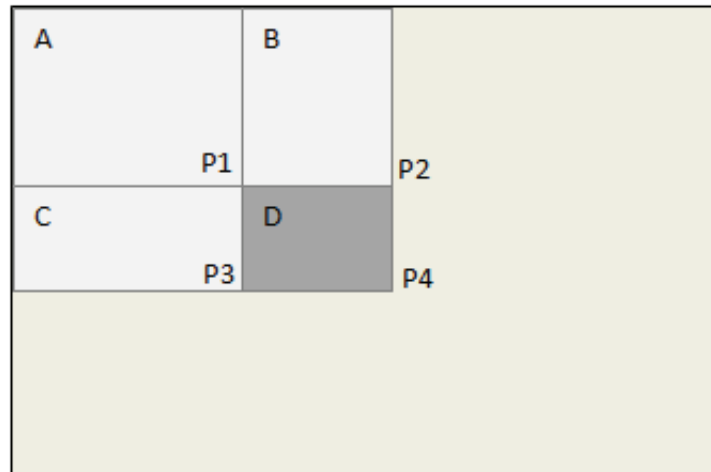


Figura 2.9: Calculo da Imagem Integral



Figura 2.10: Componentes Básicos do Modelo de Viola e Jones

IEEE *International Conference on Data Mining* (ICDM) (12/2006) como um dos 10 mais importantes algoritmos em mineração de dados (WU, 2008). A mais conhecida instância do AdaBoost, o “Robust real-time face detection” (VIOLA e JONES, 2004) é um dos mais conhecidos algoritmos para detecção de faces/objetos, sendo o algoritmo de detecção de faces do OpenCV.

O algoritmo genérico de *Haar like features* (FREUND, 1997) considera o seguinte: Boosting

- 1: D o conjunto de dados de treinamento originais
- 2: K o número de classificadores base
- 3: T os dados de teste
- 4: **for** $i = 1$ to k **do**
- 5: crie conjunto de treino D_i a partir de D
- 6: Construa um classificador h_i a partir de D
- 7: **end for**
- 8: **while** cada registro de teste $x \in T$ **do**
- 9: $C * (x) = \text{Voto}(h_1(x), h_2(x), \dots, h_k(x))$
- 10: **end while**

O detector de objeto do OpenCV foi proposto inicialmente por Paul Viola e

melhorado por Rainer Lienhart (LIENHART et al., 2003). Primeiro, um classificador é treinado com algumas centenas de pontos de amostragem de um objeto em particular (um veículo por exemplo), chamado de amostras positivas, e amostras negativas, que são imagens que não contenham o objeto procurado.

Depois de um classificador treinado, ele pode ser aplicado a uma região de interesse. Para procurar o objeto em toda a imagem, o classificador pode mover a janela de pesquisa através da imagem e avaliar, a cada local, a possível presença do objeto pesquisado. O classificador é projetado de modo que possa ser facilmente redimensionado, a fim de encontrar os objetos de interesse em tamanhos diferentes. Tal processo é chamado de digitalização e é mais eficiente do que redimensionar a imagem em si. Assim, para encontrar um objeto de tamanho desconhecido na imagem, o processo de digitalização deve ser feito várias vezes, em diferentes escalas.

Para Freund (1997), o AdaBoost considera um conjunto de N exemplos de treinamento $\{(x_i, y_i) | i = 1, 2, \dots, N\}$, onde $y_i \in \{+1, -1\}$ é o rótulo de classe associado com a instância $x_i \in \mathfrak{R}$. Dado um classificador aplicado ao conjunto de treinamento, é necessário estimar a sua taxa de erro na classificação realizada. A taxa de erro de uma hipótese (classificador fraco) h é dada por

$$\epsilon = Pr_{i \in \mathcal{D}}(h(x_i) \neq y_i).$$

A importância de um classificador fraco é dada pelo parâmetro

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right),$$

onde t é o índice de iteração. Portanto, quanto menor a taxa de erro, maior é a importância atribuída ao classificador em t . O parâmetro α_t será usado para atualizar os pesos de treinamento. Considerando $\omega_i^{(t)}$ o peso atribuído ao exemplo (x_i, y_i) na iteração t , a atualização dos pesos é dada por

$$\omega_i^{(t+1)} = \frac{\omega_i^{(t)}}{Z_t} x \begin{cases} \exp^{-\alpha_t} & \text{Se } h_t(x_i) = y_i \\ \exp^{\alpha_t} & \text{Se } h_t(x_i) \neq y_i \end{cases},$$

onde Z_t é o fator de normalização usado para assegurar que o somatório dos pesos seja 1. O resultado final do algoritmo se baseia na junção dos resultados obtidos pelos classificadores fracos, ponderada pelo peso calculado de cada um deles, ou seja,

$$H_f(x) = \text{sign} \left(\sum_t \alpha_t h_t(x) \right).$$

Desta forma o algoritmo do AdaBoost se constitui nos passos descritos no Algoritmo 2.5 (FREUND, 1997): [h!] AdaBoost

- 1: Inicializa o peso para todos os N exemplos
- 2: Seja t o número de iterações
- 3: **for** $i = 1$ to t **do**
- 4: Criar conjunto de treinamento D_i pela amostragem de D de acordo com o peso
- 5: Treinar um classificador base h_t sobre D_i
- 6: Aplicar h_t para todos os exemplos no conjunto de treinamento original D
- 7: Calcular a taxa de erro de h_t (ϵ)
- 8: **if** taxa de erro for maior que 50% **then**
- 9: Reinicializa os pesos para todos os N exemplos
- 10: Vai para o passo 4
- 11: **end if**
- 12: Calcular a importância do classificador (α)
- 13: Calcular o novo peso para cada exemplo (ω)
- 14: **end for**
- 15: Constrói o classificador final (forte) ponderando as importâncias (α) de cada classificador fraco

2.6 COMPONENTES CONEXOS

Pontos de uma imagem que apresentam propriedades similares podem ser agrupados para formar uma região. Este é o objetivo de um componente conexo, identificar um padrão de igualdade ou semelhança entre pixels ligados, e a partir disto tratar este conjunto de pixels como um conjunto único.

Este padrão de igualdade pode ser mapeado sobre um ou diferentes atributos. Os mais comuns são intensidade de cinza, cor e informações sobre textura.

Um componente conexo de uma imagem binária é o maior conjunto possível de pontos ligados (PEDRINI e SCHWARTZ, 2007). A adjacência entre os pixels depende do critério de vizinhança adotado. Um pixel p é conexo a um pixel p_2 , se existir um caminho de p a p_2 no grafo definido por A , sendo A a relação de adjacência. Tal relação de adjacência, nada mais é que o regramento da distância euclidiana entre dois pontos e o sentido (vertical, horizontal e diagonal) a ser considerado.

A extração de componentes conexos, pode considerar a utilização de diferentes algoritmos de busca em largura ou profundidade da teoria de grafos. Em (BRADSKI e ADRIAN, 2008) representa esta extração, assumindo a existência de um ponto semente p conhecido e pertencente ao componente conexo na imagem I . A extração pode ser realizada por meio de um processo iterativo, dado por:

$$X_k = (X_{k-1} \oplus B) \cup I$$

$$k = 1, 2, \dots$$

em que $X_0 = p$ e B é um elemento estruturante¹ adequado. Se $X_k = X_{k-1}$, então o resultado convergiu e o resultado é o conjunto de pixels conexos que formam o componente conexo.

A Figura 2.11 demonstra uma imagem em tons de cinza contendo duas regiões de forma oval em tons mais claros. Ao aplicar o detector de bordas foi possível identificar dois componentes conexos graças a regra de semelhança definida.

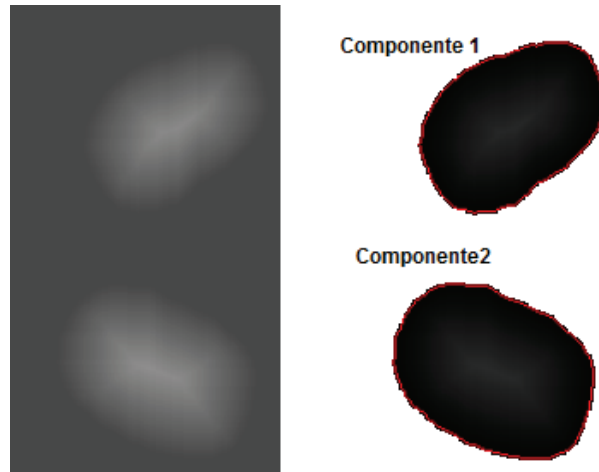


Figura 2.11: Identificação de Componente Conexos

2.7 OPERADORES MORFOLÓGICOS

O processamento de imagens, por vezes, requer atuadores que, além de avaliar e compreender a sua estrutura matricial (pixels), seja capaz de agir sobre ela. Os operadores morfológicos podem prover resultados como realce, filtragem e esqueletização². Para Pedrini e Schwartz (2007), a base da morfologia consiste em extrair de uma imagem desconhecida a sua geometria através da utilização da transformação de uma outra imagem completamente definida, ou seja, consiste em extrair as informações relativas a geometria e a topologia de um conjunto desconhecido (no caso uma imagem) pela transformação através de outro conjunto bem-definido, chamado elemento estruturante.

A morfologia matemática faz uso da teoria de conjuntos para representar a forma dos objetos em uma imagem. Originalmente foi desenvolvida para tratar imagens binárias. Os principais operadores morfológicos são a dilatação e a erosão:

1. A dilatação (\oplus) combina dois conjuntos usando a adição vetorial. Seu efeito é o

¹Elemento Estruturante: É definido por uma matriz que é caracterizado como conjuntos definidos e conhecidos (forma e tamanho), e que são comparados ao conjunto desconhecido da imagem. Alguns exemplos de elemento estruturante: cruz, linha horizontal, linha vertical, diamante, disco, quadrado.

²Consiste de reduzir a imagem a analisada em uma cadeia simples com largura de um pixel, mas preservando as características relevantes

alargamento das fronteiras gerando um objeto maior com buracos menores (nas regiões internas). A dilatação de um conjunto A pelo conjunto B é definida por:

$$A \oplus B = \{c | c = a + b, a \in A, b \in B\}$$

2. A Erosão (\ominus) representa o estreitamento de um conjunto pela subtração de outro. Seu efeito é a esqueletização da forma submetida ao processo de erosão. Ela é expressa como a interseção de A e B

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

A Figura 2.12 compara o efeito causado pelas operações de dilatação e erosão.

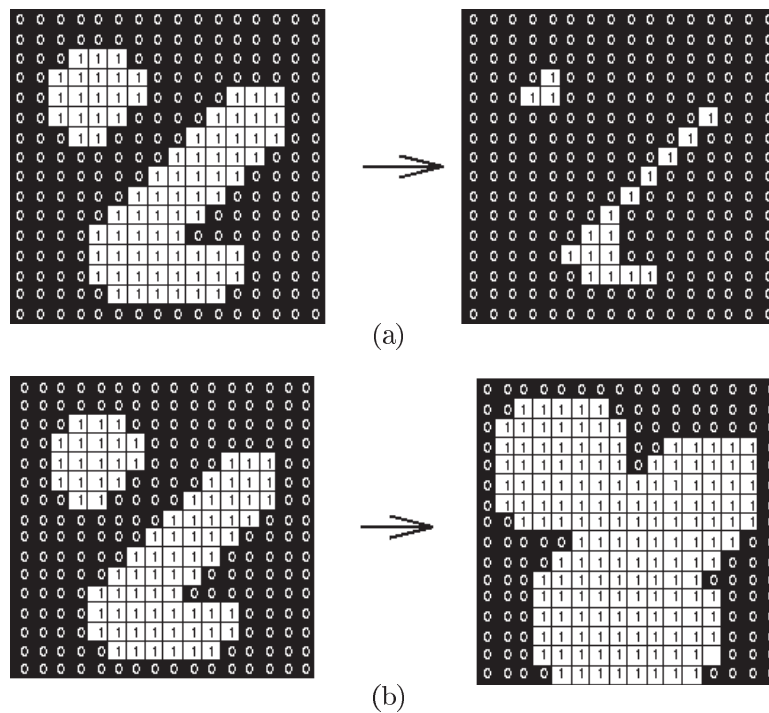


Figura 2.12: Operadores Morfológico: (a) Erosão. (b) Dilatação (MORPHOLOGY,)

Vale ressaltar que outras abordagens e representações poderão ser encontradas na bibliografia (BRADSKIE ADRIAN, 2008), e que aqui foram retratadas de forma simplista, considerando apenas operadores binários os mesmos que usados durante o desenvolvimento do projeto.

Pedrini cita outras duas operações morfológicas: a abertura e o fechamento. A Abertura é definida como a união de todas as transações de B que estão contidas na imagem A, ou seja:

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

Já o fechamento é definido como:

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

este, geometricamente se refere como a união de todas as transações de B que não estão contidas em A.

2.8 CALIBRAGEM DE CÂMERA

Cada imagem capturada pela câmera é uma representação bidimensional da cena tridimensional. Como ilustração, a Figura 2.13 demonstra a relação das coordenadas de mundo de uma estrada, comparada com as coordenadas de imagem desta mesma estrada. Existem diferentes modelos matemáticos para representar o mapeamento de cada coordenada tridimensional da imagem $\mathbf{x}_w = (x_w, y_w, z_w)^T$ em uma coordenada de imagem $\mathbf{u} = (u, v)^T$, sendo que uma das mais utilizadas é a projeção em perspectiva adotando um modelo de câmera *pinhole* (DAVIES, 2005).

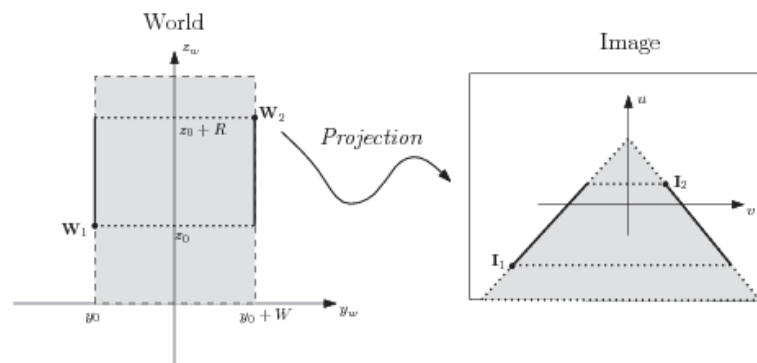


Figura 2.13: Esquerda: Estrada em coordenadas de Mundo. Direita: Estrada em coordenadas de imagem.

No modelo *pinhole*, há uma matriz de rotação \mathbf{R} e um vetor de translação \mathbf{x}_0 relacionados ao posicionamento da câmera com relação à cena:

$$\mathbf{x}_c = \mathbf{R}(\mathbf{x}_w - \mathbf{x}_0), \quad (2.9)$$

onde $\mathbf{x}_c = (x_c, y_c, z_c)$ é um vetor tridimensional com as coordenadas de cada ponto em coordenadas de câmera. As coordenadas de imagem \mathbf{u} são dadas pela projeção perspectiva:

$$u = \frac{f x_c}{z_c}, \quad v = \frac{f y_c}{z_c}, \quad (2.10)$$

onde f é a distância focal da câmera. Dessa forma, é possível mapear cada ponto \mathbf{x}_w no

espaço em uma coordenada de imagem \mathbf{u} , conhecendo os parâmetros \mathbf{R} , \mathbf{x}_0 e f .

Através de calibragem da câmera e tendo algumas informações sobre a cena, é possível criar uma relação mais clara e precisa entre o mundo real e a representação da imagem, podendo desta forma interpretar melhor as informações contidas e possibilitando a tomada de decisão por qualquer algoritmo.

3 REVISÃO BIBLIOGRÁFICA

A detecção de obstáculos em pista e em tempo real para Sun (SUN et al., 2006) é dividida em dois passos: o primeiro é o HG (Geração da Hipótese - *Hypothesis Generation*), onde são localizados candidatos a obstáculos em uma imagem. O segundo passo é o HV (*Hypothesis Verification*), onde testes são realizados sobre a região pré-selecionada pelo gerador de hipóteses para confirmar a presença de obstáculos na imagem.

3.1 GERAÇÃO DE HIPÓTESES

Sun (SUN et al., 2006) divide os métodos utilizados para a identificação de candidatos a obstáculos no passo HG como:

1. Base de Conhecimento;
2. *Stereo Based*, que não será abordado neste trabalho;
3. Fluxo Óptico.

3.1.1 Base de conhecimento

Ao se buscar candidatos a obstáculos, algumas premissas sobre a imagem e características de um veículo podem ser invocadas, tais como: simetria, cor, sombra e características geométricas.

3.1.1.1 *Simetria*

Uma informação importante é a simetria, pois como define Marola (MAROLA, 1989), a principal assinatura de um objeto feito pelo homem é a simetria. Sun (SUN et al., 2006), lembra que as imagens de veículos observadas da retaguarda ou frontalmente em geral são simétricas. Vários fatores influenciam a identificação da simetria, desde uma obstrução onde o objetivo não é visto na íntegra, até a questão da luminosidade e a intensidade.

3.1.1.2 *Cor*

O atributo cor também pode ser explorado em aplicações de visão computacional a veículos inteligentes. É possível observar nos artigos (JUNGe KELBER, 2005a, 2004b, 2004a, 2005b), a utilização da cor como forma de detectar as bordas da pista. Através da avaliação desta característica, pode-se trabalhar na identificação da estrada/pista e assim classificar como candidato a obstáculo tudo que estiver sobre a pista, mas que não se encaixou na classificação de pista. Além disso, a pintura de alguns veículos foge ao padrão de cores normalmente encontrado na natureza. Deve-se considerar que a cor

do objeto depende da iluminação, ou seja, a cor de um veículo pode ser percebida de forma diferente em diferentes momentos do dia e em diferentes condições climáticas, o que adiciona complexidade ao problema.

3.1.1.3 Bordas Horizontais e Verticais

As informações sobre as bordas da imagem são apontadas por Sun et al. (2006) como um dos atributos mais promissores na detecção de candidatos a obstáculos. Ao se visualizar um veículo tanto pela traseira como frontalmente, este veículo é composto por uma série de bordas extraídas de sua estrutura, como para-choques, para-brisa. Assim, a presença de uma borda no sentido vertical pode indicar a presença de um veículo, enquanto as bordas horizontais podem apoiar na delimitação do espaço deste veículo. Vários trabalhos tem se focado nesta linha (GOERICK et al., 1996; SUN et al., 2002).

A detecção destas linhas passa inicialmente pela detecção das bordas da imagem, uma vez identificadas as bordas, as linhas horizontais e verticais podem ser percebidas tanto pela avaliação de histogramas da imagem como pela transformada Hough¹ capaz de identificar linhas e curvas em uma imagem. A Figura 3.1 (TRANSFORM, 2009) demonstra a imagem original, a identificação das bordas e as linhas identificada pelo método Hough.

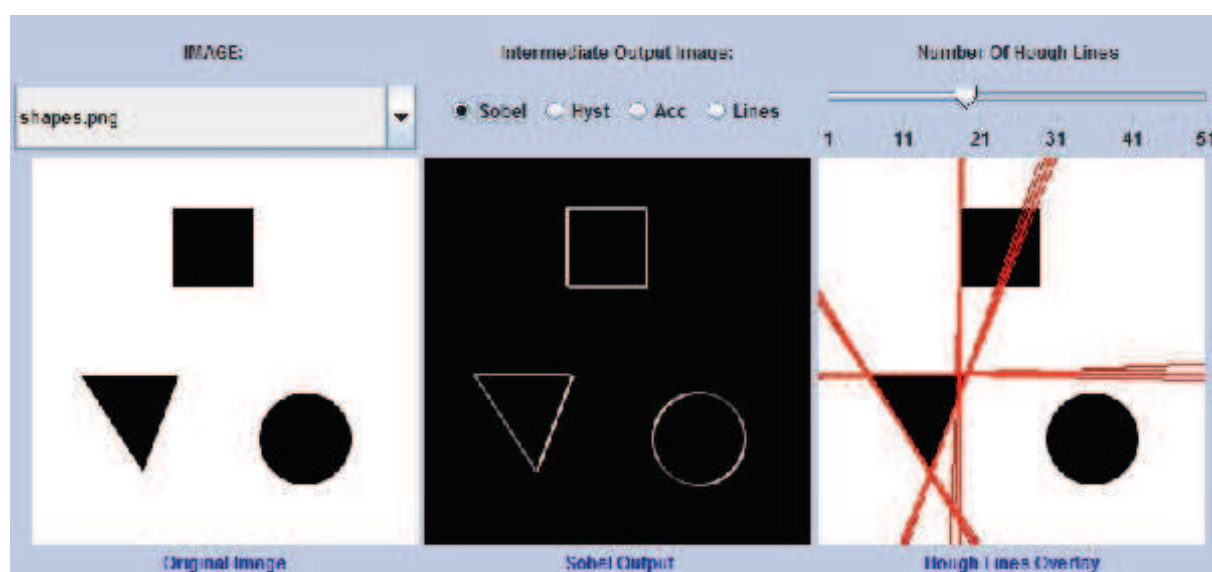


Figura 3.1: Transformada Hough evidenciando as retas de uma imagem

3.1.1.4 Quina

Este atributo está fundamentado na informação de que os veículos em geral, tem uma forma retangular, com quatro cantos. Bertozzi (BERTOZZI et al., 1997) propõem um

¹A transformada de Hough (HT) é um método para detecção de formas que são facilmente parametrizadas em especial linhas, círculos, elipses (NIXON e AGUADO, 2008) em imagens computacionais. Em geral, a transformada é aplicada após a imagem sofrer um pré-processamento, normalmente a detecção de bordas.

método considerando as informações de quina. No trabalho a identificação do veículo é dada pela identificação dos cantos dos vidros traseiros e dianteiros do veículo, além dos cantos formados pelos para-choques.

De fato, a identificação das quinas nada mais é do que a confirmação de cruzamento entre as linhas horizontais e verticais da imagem identificadas previamente.

Existem vários algoritmos utilizados para a detecção de quinas conforme destacados na sessão 2.4. Os trabalhos (MORAVEC, 1980), (HARRIS e STEPHENS, 1988), (SMITH e BRADY, 1995) detalham alguns destes algoritmos.

3.1.2 Fluxo Óptico

Fluxo Óptico pode ser definido como um campo de velocidade que descreve o movimento aparente dos padrões de intensidade em uma imagem. Dada uma sequência de imagens variando no tempo, é possível obter, para cada pixel, um vetor de velocidade. O fluxo óptico pode ser um importante sinalizador de obstáculos. É possível detectar que, de um quadro de vídeo para o seguinte, houve um movimento e que um possível objeto teve um fluxo óptico diferente do encontrado na maior parte da imagem. Porém Sun (SUN et al., 2006) levanta algumas preocupações, pois este método não permite identificar obstáculos estáticos, seu custo computacional é elevado, sua eficiência é maior na detecção de obstáculos distantes da câmera do que com objetos mais próximos. Soma-se a estes fatores a inevitável vibração que a câmera pode ter no veículo e que distorce o fluxo real. A avaliação do fluxo óptico também pode ser distorcida pela velocidade do veículo, pois em alta velocidade a distância entre os pixels de um quadro para outro pode ser tamanha que passa a prejudicar a identificação do fluxo. Os artigos (BARRON et al., 1994), (GIACHETTI et al., 1994) são exemplos de trabalhos nesta linha. A Figura 3.2 apresenta o fluxo óptico de um veículo em movimento nos moldes do que poderia ser realizado neste trabalho para a identificação de candidato a obstáculo.

3.2 VALIDAÇÃO DE HIPÓTESES

O resultado da Geração das Hipóteses é a geração de candidatos a obstáculo. Após a identificação destes candidatos é necessário refinar a avaliação e definir o que é obstáculo de fato. Ou seja, das hipóteses levantadas, é necessário testá-las e comprovar a sua validade. Sun (SUN et al., 2006), propõem que as abordagens para validação das hipóteses sejam divididas em duas categorias: Baseada em modelo (*Template-based*) e Baseado em aparência.

3.2.1 Validação Baseada em Modelo

Este método usa padrões predefinidos de veículos e faz uma correlação daquilo que foi classificado como candidato a obstáculo com este padrão. Estes padrões podem

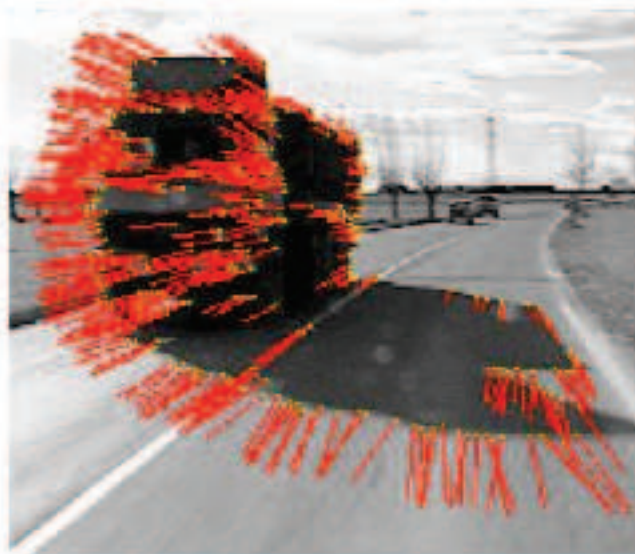


Figura 3.2: Fluxo Óptico da Imagem

ser mais genéricos ou mais detalhados conforme o modelo utilizado. Alguns trabalhos propõem que um veículo observado frontalmente ou pela traseira, tem a forma de um 'U' (uma borda horizontal, duas bordas verticais e ocorrendo o encontro entre as bordas verticais e a horizontal) (HANDMANN et al., 1998). Outro trabalho, propõe a identificação do emplacamento presente nos veículos ou a presença da janela traseira (KASPRZAK e NIEMANN, 1996).

Outros trabalhos nesta linha (ITO e YAMADA, 1995; GRAEFE e REGENSBURGER, 1996) propõem avaliações similares. E mesmo adotando um padrão de estrutura, este padrão pode ser distinto para objetos próximos e para objetos mais distantes da câmera, o que poderia ser contornado se uma câmera calibrada (JEONG e NEDEVSCHI, 2008) fosse utilizada. Além da calibragem, o uso de mais de um modelo pode ajudar para identificação de veículos: um modelo para identificação de obstáculos próximos à câmera e outro para obstáculos distantes da câmera. Soma-se a isto à validação de informações, como simetria, que pode ser caracterizada por um retângulo em volta do objeto, satisfazendo alguma condição de proporcionalidade.

Como visto anteriormente, diferentes técnicas são abordadas para a geração de hipóteses. Reforçando o argumento central desta divisão entre geração e validação da hipótese está o tempo de processamento e a divisão de tarefas que devem ser consecutivas, porém aplicadas sem desperdício de tempo e poder computacional.

De um modo geral, as hipóteses geradas são rotuladas e tem sua região na imagem delimitada e destaca daquilo que é considerado não hipótese e pode ser chamado de elementos de não interesse.

Em se tratando de identificação de obstáculos e sendo estes obstáculos delimitados a veículos, é criada a condição mínima para descartar toda e qualquer hipótese que não

tenha as dimensões de um veículo. A largura de um automóvel tem aproximadamente 1,8 metros. Segundo o Grupo Técnico de Pesos Dimensões e Combinações de Veículos do DENATRAN o limite para a largura de veículos é de 2,60 m (ABCRA...). Sendo assim, e considerando o objetivo do projeto em detectar obstáculos a partir de uma câmera embarcada em movimento e em tempo real, encontrar um meio de identificar a silhueta do veículo na visão frontal ou traseira, medir esta silhueta e comparar com os padrões estabelecidos, torna possível descartar toda e qualquer hipótese que não se encaixe neste padrão.

Uma proposição é destacada por (GAO et al., 2008), que identificam a posição das bordas da pista e do veículo, usando uma projeção inversa transformando as coordenadas da imagem em coordenadas de mundo a partir de uma câmera calibrada.

A Figura 3.3 extraída de (GAO et al., 2008), demonstra a delimitação da região da largura da pista e da largura do veículo estabelecendo visualmente uma distinção entre eles. É possível perceber na imagem a relação entre a largura da faixa da pista e a largura do veículo.

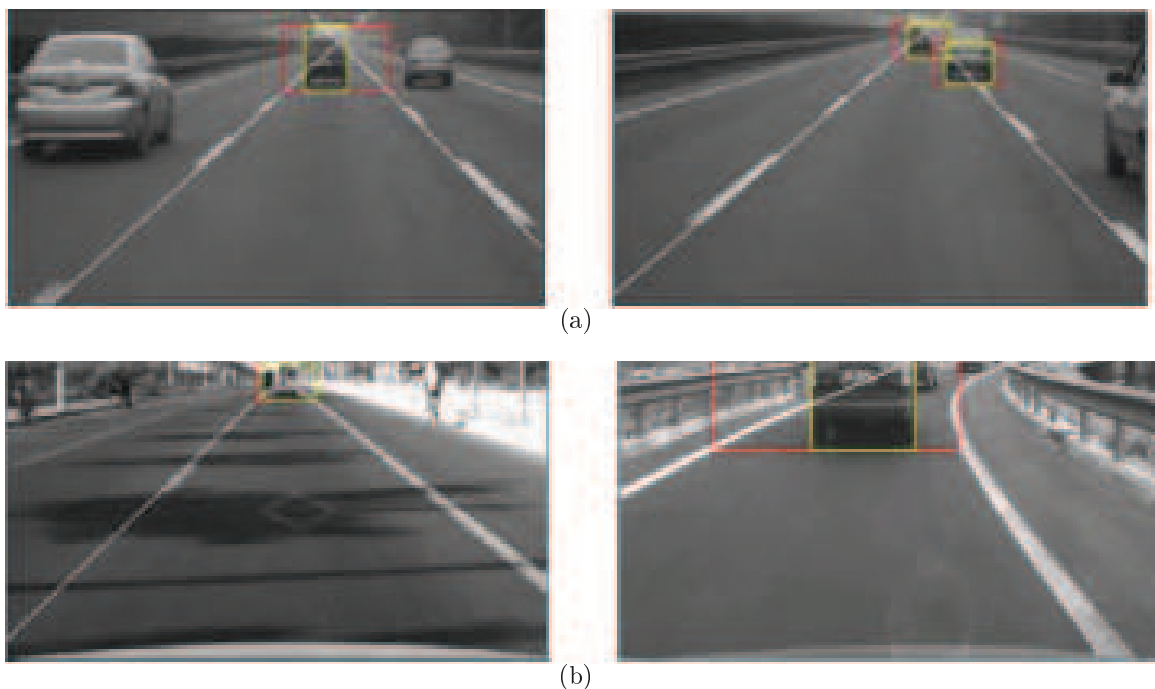


Figura 3.3: Proporção entre largura da pista e do veículo

O estudo realizado por (KOSTIA, 2009), apesar de não ter uma abordagem mais distante quanto ao objetivo, visto que considera uma câmera fixa também considera a largura do veículo como característica para qualificação das hipóteses. Também é considerada a calibragem da câmera, que por estar fixa e não em movimento, tem uma acuracia maior.

A Figura 3.4 demonstra a identificação dos faróis dos carros, e a marcação que

se estabelece para a distância entre eles. Este artigo também faz uso do realce e do isolamento dos *blobs* (componentes conexos) que também é tratado pelo trabalho desta dissertação.



Figura 3.4: Distância entre faróis e outros veículos (a) imagem destacando os faróis (b) Componentes conexos identificados

3.2.2 Validação Baseada em Aparência

Este método tem o desafio de classificar veículos e não veículos. Ele tenta aprender os aspectos que determinam a fronteira entre um e outro.

Deve-se considerar que este método pode pressupor de dois momentos. O primeiro é o aprendizado, onde o sistema deverá reconhecer veículos e não veículos através de treinamento e/ou por algum modelo probabilístico considerando um leque de características de cada uma das classes (SUN et al., 2006). O segundo momento é a classificação de veículos a partir do processamento do candidato a obstáculo em tempo real. As abordagens para a construção deste aprendizado podem ser via redes neurais (GOERICK et al., 1996), ou por modelagem de uma distribuição de probabilidades de redes baesianas (NOLL et al., 1995; SCHNEIDERMAN, 2000), além do uso de *Haar wavelet features* (SUN et al., 2002).

O método baseado em aparência pode ser dividido em dois grupos: métodos de extração de características e classificador propriamente dito (SUN et al., 2004). Em (MATTHEWS et al., 1995) o PCA (*Principal Component Analysis*) é utilizado para a extração de características, onde todos os candidatos a obstáculos são redimensionados para 20×20 , então esta imagem é dividida em 25 janelas de tamanho 4×4 . O PCA é aplicado em cada uma destas sub-janelas e o resultado do PCA local é encaminhado para a rede neural treinada para verificar a hipótese. Em (SUN et al., 2004) são citadas outras abordagens dentre as quais:

- Utilização do PCA combinado com o classificador de vizinho mais próximo (*nearest-neighbor classifier*);
- Uso de método LOC (*Local Orientation Code*) para a extração das informações de bordas da imagem, onde a informação do histograma do LOC dentro da área de interesse é encaminhada para a rede Neural fazer a classificação;
- O filtro Gabor fornece um mecanismo para obter a orientação e medida ajustada de bordas e linha da imagem. Veículos possuem bordas e linhas fortes em diferentes direções e escalas, sendo um classificador de veículo;

De um modo geral, os métodos baseados em aparência são mais exatos que métodos baseados em modelos (SUN et al., 2006). Por outro lado, o custo computacional é maior no método baseado na aparência que no baseado em modelos. Outra característica marcante é a assinatura de funções de inteligência artificial e pré-processamento para os métodos baseados em aparência, por sua vez, no método baseado em modelos cria regras executadas em tempo de execução.

4 METODOLOGIA PROPOSTA

A metodologia aqui proposta para a detecção de obstáculos em pistas, se baseia nas imagens obtidas de uma única câmera monocular fixa instalada em um veículo. Nessas imagens, serão aplicadas as técnicas descritas no capítulo de conceitos básicos, e mencionadas na revisão bibliográfica. Quanto ao tipo de obstáculo, a metodologia está restrita à identificação de outros veículos que se posicionem como obstáculo na pista considerada em relação ao veículo portador da câmera.

O funcionamento da metodologia proposta pode ser organizada em quatro etapas ou Módulos principais:

- Módulo 1: Delimitação da Região de Interesse (RDI): Restringe a aplicação do detector de obstáculos apenas para a região da pista considerando mão e contra-mão;
- Módulo 2: Identificação do Padrão de Pista: Mapeia e modela, dentro da região de interesse, os pixels em pista e não pista;
- Módulo 3: Geração de Hipóteses: Atua sobre os elementos considerados não pista, realiza filtros e gera componentes conexos como possíveis obstáculos;
- Módulo 4: Validação das Hipóteses: Analisa os possíveis obstáculos na busca de características de veículos, validando ou rejeitando o candidato a obstáculo.

Além desses módulos, considerou-se uma etapa de pré-processamento, necessária para o treinamento do classificador *haar like feature*. A utilização das tecnologias de *Haar Like Features* integra o algoritmo em dois momentos distintos: o primeiro é o treinamento que formará a árvore de decisão que avaliará a presença ou não de veículos na imagem. O segundo momento é a aplicação do algoritmo em tempo real. Em ambos os casos, a utilização desta tecnologia se faz de forma independente do resto do algoritmo.

O treinamento é algo anterior e independente da execução do algoritmo. São utilizadas ferramentas e recursos distribuídos juntamente com a biblioteca OpenCV para este treinamento (HAARTRAINING,). A implementação do OpenCV utiliza o AdaBoost como algoritmo.

O treinamento conta com um banco de imagens disponibilizada pelo MIT CBLD (*Center for Biological and Computational Learning*) (PEDESTRIAN...,) e imagens fornecidas por Fan Jiang (JIANG e LIN, 2006). Ao todo são 2268 imagens de automóveis: são fotos da frente e da traseira dos veículos que servirão como imagens positivas, além de 5000 imagens de arquivo pessoal utilizadas como imagens negativas.

A aplicação do AdaBoost se justifica para reforçar o classificador de obstáculos e para tentar antecipar a identificação de outros veículos, melhorando o algoritmo tanto em

acertividade quanto em eficiência, podendo informar ao motorista com maior antecedência da presença de obstáculos.

Outra etapa que merece ressaltar é a etapa de calibragem da câmera. Os parâmetros provenientes desta calibragem são importantes, tanto para a delimitação da região de interesse, quanto para a validação das hipóteses.

A Figura 4.1 é a reprodução do cenário proposto pelo algoritmo. A referência matemática para a conversão das coordenadas de imagem em coordenada de mundo são discutidas nas seções 2.8 e 4.2

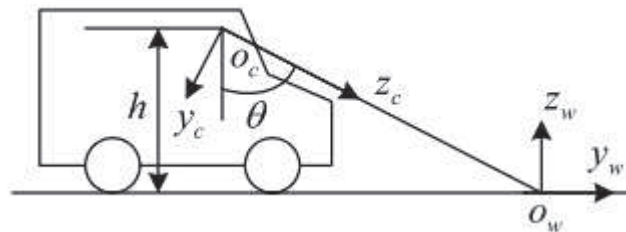


Figura 4.1: Calibragem da Câmera

Por último, tem-se uma interface com o usuário, pelo qual as informações obtidas da metodologia proposta se transmitam ao motorista através de sinais, sinalizando a possível presença de obstáculos na pista. Assim, tem-se uma ferramenta de apoio ao motorista que pode ser muito útil no aumento da segurança em viagens, diminuindo o número de acidentes.

A metodologia de detecção de obstáculos aqui proposta é o encadeamento de diversos processos que, em si, possuem suas sutilezas. A Figura 4.2 mostra o fluxo lógico do algoritmo que será tratado e detalhado nas próximas seções deste capítulo.

4.1 MÓDULO 1: DELIMITAÇÃO DA REGIÃO DE INTERESSE

O ponto de partida do algoritmo é a **captura da imagem**, que será realizada a partir de uma câmera monocular embarcada em um veículo.

A imagem capturada pela câmera normalmente contém diversas informações. Além da pista à frente do motorista, há também o firmamento (céu) e a região ao lado da pista, conforme demonstrado na Figura 4.3. Os candidatos a obstáculo que se deseja detectar são basicamente outros veículos trafegando na mesma rodovia, o que limita a região de busca por obstáculos. A limitação da região de interesse da imagem para seu posterior processamento apresenta as seguintes vantagens:

- O tempo de processamento deve ser menor em função da menor quantidade de informações a serem processadas;

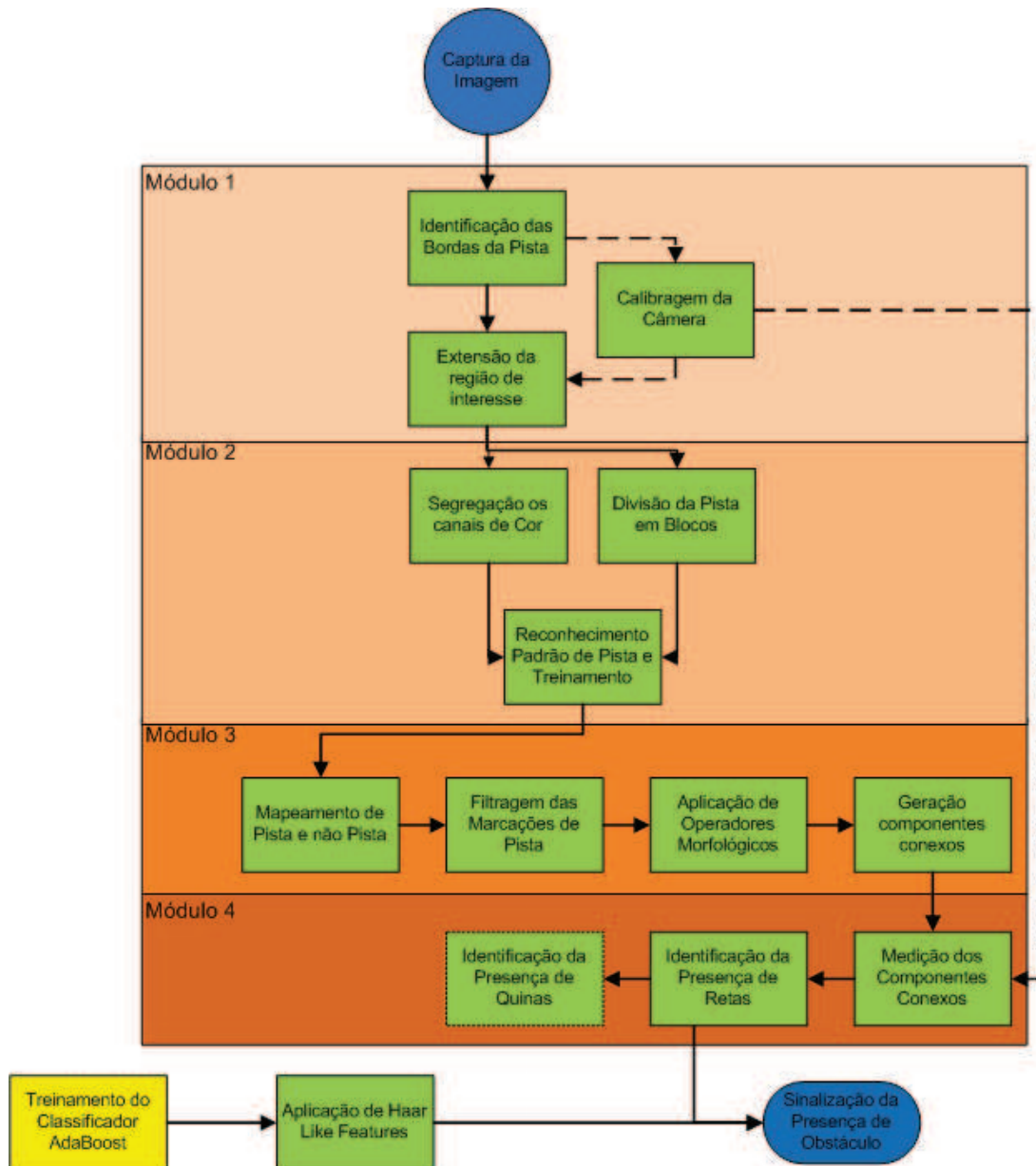


Figura 4.2: Fluxo Lógico de Desenvolvimento

- A diminuição da taxa de falsos positivos (ou seja, regiões da imagem que são falsamente detectadas como obstáculos), visto que apenas a região da pista de rolagem e sua vizinhança serão processadas;



Figura 4.3: Região de interesse da imagem

Os algoritmos descritos em (JUNGe KELBER, 2005a, 2004b, 2004a, 2005b), são empregado para **identificar as bordas da pista** de rolagem, onde se encontra o veículo com a câmera embarcada. Entretanto, a região de busca onde serão procurados os obstáculos deve incluir também uma vizinhança à esquerda, onde outros veículos possam estar trafegando ou parados. Embora existam diversos algoritmos para detecção das bordas da pista (SUN et al., 2004), a escolha pelos algoritmos propostos em (JUNGe KELBER, 2005a, 2004b, 2004a, 2005b) foi feita com base em sua baixa complexidade computacional, robustez com relação a variações de iluminação.

O modelo considera um sistema de coordenadas da imagem e um limiar (*threshold*) x_m que separa a pista em dois campos, um mais próximo e outro mais distante da câmera conforme mostra a Figura 4.4.

É uma combinação de uma função linear para o campo mais próximo e de uma função parabólica para o campo mais distante, contudo mantendo uma conexão contínua entre estes campos. A porção linear (mais próxima), fornece informações sólidas sobre a orientação local do veículo, enquanto que a porção parabólica é flexível para acompanhar eventuais curvas da estrada.

$$f^k(x) = \begin{cases} a^k + b^k(x - x_m), & x > x_m, \\ a^k + b^k(x - x_m) + c^k(x - x_m)^2, & x \leq x_m \end{cases} \quad (4.1)$$

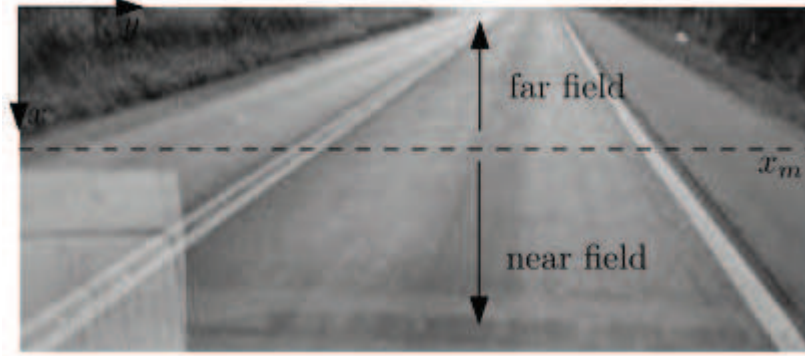


Figura 4.4: Divisão da pista em campos para detecção de bordas da pista

onde $k \in \{l, r\}$ delimita o limiar entre as pistas à direita (*right*) e à esquerda (*left*). Isso indica que cada faixa de fronteira é caracterizada por três coeficientes (a^k, b^k, c^k) em que b^k representa a orientação local da borda e c^k representa a curvatura presente no campo mais distante da câmera.

Uma vez que esses parâmetros são determinados por um determinado quadro, as bordas da imagem são consideradas em uma região de bordas de pista detectadas no quadro seguinte. Os parâmetros (a^k, b^k, c^k) passam a ser obtidos pela minimização do erro quadrático ponderado.

$$E^k = \sum_{i=1}^m M_{n_i}^{k} [y_{n_i}^k - f^k(x_{n_i}^k)]^2 + \sum_{j=1}^n M_{f_j}^{k} [y_{f_j}^k - f^k(x_{f_j}^k)]^2 \quad (4.2)$$

onde $(x_{n_i}^k, y_{n_i}^k)$, para $i = 1, \dots, m$, representa as coordenada do pixel não zero no limiar no campo mais próximo da imagem, e $M_{n_i}^k$ a respectiva magnitude para $k = \{l, r\}$. Da mesma forma $(x_{f_j}^k, y_{f_j}^k)$ e $M_{f_j}^k$ para $j = 1, \dots, n$, denota, respectivamente, estas características para o pixel de borda j no campo mais distante. É restrição do processo de identificação das bordas da pista a necessidade de marcações (sinalização) na estrada, uma vez que o algoritmo se baseia nesta informação para o processo de identificação das bordas da pista.

Este algoritmo necessita da especificação de alguns parâmetros para o posicionamento inicial do detector de bordas. Tais parâmetros variam em função da resolução e ângulo de visão da câmera e devem ser informados ao inicializar o detector. Conforme Jung e Kelber (2004b) a inicialização do detector deverá satisfazer às seguintes condições:

- o veículo deverá estar posicionado em um porção reta da pista;
- o veículo deverá estar alinhado com a pista;
- não deverão constar no primeiro quadro do detector outras estruturas lineares que

não as marcações das bordas da pista.

Para a **obtenção da região de interesse** (ou **ROI**, de *Region on Interest*), a pista de rolagem obtida pelos algoritmos de detecção de borda da pista, deve ser duplicada à esquerda, considerando a pista de contra-mão, que é uma **extensão da região de interesse** inicialmente delimitada pelas bordas da pista.

Devido à perspectiva da câmera, regiões mais próximas da câmera ocupam uma quantidade maior de pixels do que regiões mais afastadas. Para resolver este problema, em que precisa-se relacionar as distâncias em coordenadas de mundo em distâncias equivalentes em coordenadas de imagem. Para isto, usa-se uma câmera calibrada que permitirá estimar esta relação. O uso dos parâmetros da câmera calibrada, possibilita o mapeamento de coordenadas de mundo em coordenadas de imagem, conforme a Equação (2.10).

Assim, o processo de obtenção da ROI é feito linha a linha, da parte inferior da imagem até o ponto de fuga da pista (ponto que corresponde à linha do horizonte, detectado como o ponto de intersecção das bordas da pista em coordenadas de imagem). Primeiramente é medida a distância entre as bordas da pista que foram previamente detectadas. Esta informação é convertida em coordenadas de mundo e tem-se um valor em sistema métrico. Este valor, é utilizado para projetar a esquerda a mesma distância, convertido em coordenadas de imagem, ou seja, posição dos pixels.

Pretende-se utilizar o algoritmo descrito em (JUNG et al., 2009) para realizar uma **calibragem da câmera** embarcada de forma semi-automática, para obter os parâmetros \mathbf{R} , \mathbf{x}_0 e f . No caso particular da câmera embarcada em veículos, tem-se

$$\mathbf{x}_0 = (h, 0, 0)^T \text{ e } \mathbf{R} = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{bmatrix}, \quad (4.3)$$

significando que a câmera possui uma altura h com relação ao plano do pavimento, que com um ângulo de inclinação α com relação a esse plano, conforme ilustrado na Figura 4.5.

Juntando as Equações (2.9) e (2.10), tem-se:

$$u = \frac{f(x_w \cos \alpha - z_w \sin \alpha - h \cos \alpha)}{x_w \sin \alpha + z_w \cos \alpha - h \sin \alpha}, \quad (4.4)$$

$$v = \frac{f y_w}{x_w \sin \alpha + z_w \cos \alpha - h \sin \alpha}. \quad (4.5)$$

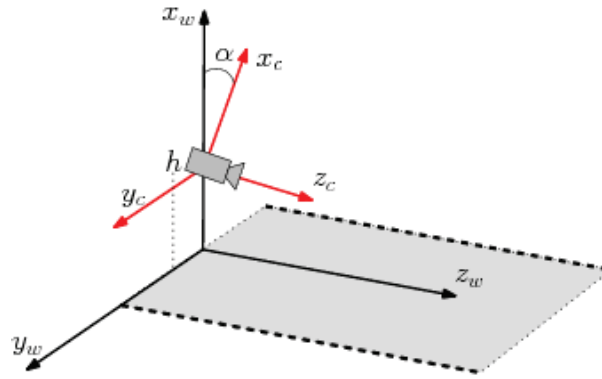


Figura 4.5: Relação entre o sistema de coordenadas 3D $(x_c, y_c, z_c)^T$ da câmera e o sistema de coordenadas 3D do mundo $(x_w, y_w, z_w)^T$.

Fazendo $x_w = 0$ nas Equações (4.4) e (4.5), e isolando y_w e z_w , obtém-se

$$y_w = -\frac{vh}{u \cos \alpha + f \sin \alpha}, \quad (4.6)$$

$$z_w = \frac{h(u \sin \alpha - f \cos \alpha)}{u \cos \alpha + f \sin \alpha}, \quad (4.7)$$

e então pode-se relacionar cada ponto (u, v) na imagem com suas coordenadas de mundo $(0, y_w, z_w)$.

O mapeamento inverso, de coordenadas de imagem para coordenadas de mundo, não é possível, já que há vários pontos no espaço tridimensional que podem ser mapeados para as mesmas coordenadas de imagem (DAVIES, 2005). Entretanto, se o objeto de interesse estiver em um plano, o mapeamento inverso torna-se possível. Em particular, assumindo que a parte frontal inferior de um obstáculo esteja aproximadamente no plano da estrada $x_w = 0$, pode-se usar as coordenadas de imagem desse candidato para descartar *blobs* (componentes conexos) que não são consistentes com o tamanho esperado de um obstáculo. Por exemplo, pode-se descartar um candidato a obstáculo muito grande em uma região distante da câmera, ou muito pequeno em uma região próxima à câmera, o que é utilizado no Módulo 4.

Dado um segmento de linha horizontal detectado em um *blob* (componente conexo) candidato à obstáculo, com extremidades $(u_0, v_0)^T$ e $(u_1, v_1)^T$, pretende-se aplicar o mapeamento dado pelas Equações (4.6) e (4.7) para achar os respectivos pontos em coordenadas de mundo. A distância entre esses pontos devem estar em uma faixa aceitável de valores (baseada na largura dos obstáculos que se espera em uma imagem) para ser validado como obstáculo. Os aspectos considerados no algoritmo levam em consideração o objetivo da proposta em detectar obstáculos, em especial detectar outros veículos.

A Figura 4.6 ilustra o funcionamento do Módulo 1. A Figura 4.6 (b) mostra, destacada em verde, o resultado da detecção das bordas da pista de mão, mostrada na

Figura 4.6 (a). A pista de mão é então duplicada para a obtenção da pista de contra-mão, completando assim a região de interesse. O resultado é destacado em vermelho na Figura 4.6 (c).

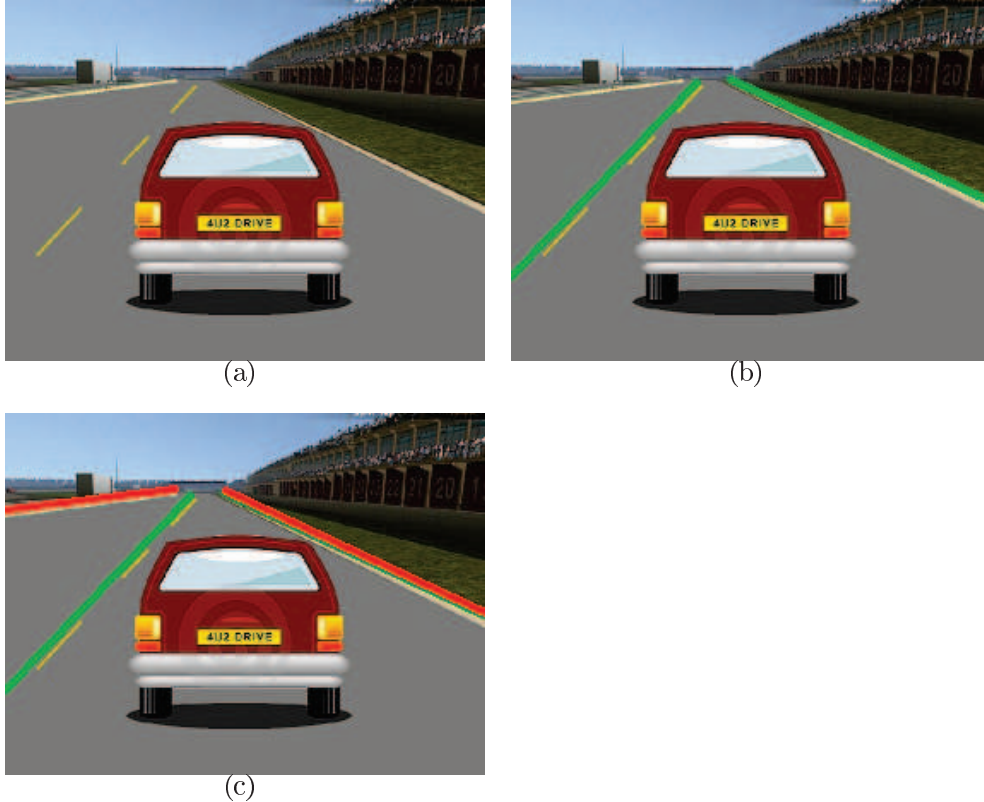


Figura 4.6: Etapas ROI: (a) Imagem original. (b) Etapa de identificação das Bordas da Pista (c) Delimitando a região de interesse

O resultado do Módulo 1 é um vetor. Este armazena a posição dos pixels a direita e a esquerda da região de interesse.

4.2 MÓDULO 2: IDENTIFICAÇÃO DO PADRÃO DE PISTA

Com a delimitação da região de interesse, ocorre a diminuição da área de atuação do algoritmo apenas para pista. Com uma área restrita de atuação, ganha-se em tempo de processamento e se torna mais simples modelar os obstáculos que são encontrados em uma pista. O que se inicia pela simples modelagem de pista e não pista, ou seja, a **identificação do padrão de pista**.

Para esta identificação do padrão de pista, o algoritmo tratará de distinguir pixels associados à pista dos demais pixels da imagem, que serão considerados candidatos iniciais a obstáculos. Inicialmente, pretende-se avaliar a informação de cor de cada pixel analisado, assumindo que os pixels relacionados com a pista são similares entre si, e suficientemente diferentes dos candidatos a obstáculos. Assumindo que os pixels do pavimento devem

ser aproximadamente homogêneos, uma distribuição gaussiana multivariada pode ser apropriada para representar este pavimento:

$$p(\mathbf{x}) = \frac{(2\pi)^{-N}}{\sqrt{|\mathbf{C}|}} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right\}, \quad (4.8)$$

onde \mathbf{x} é um vetor tridimensional contendo as cores RGB de um pixel, $\boldsymbol{\mu}$ é o vetor média da distribuição, e \mathbf{C} é a matriz de covariância. Dado um conjunto S de N pontos pertencentes ao pavimento, pode-se calcular sua média e matriz de covariância através de

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{\mathbf{x} \in S} \mathbf{x}, \quad (4.9)$$

$$\mathbf{C} = \frac{1}{N} \sum_{\mathbf{x} \in S} (\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T. \quad (4.10)$$

Dada uma amostra \mathbf{x} , pode-se avaliar sua coerência com uma distribuição Gaussiana n -dimensional através da distância de Mahalanobis, que retorna uma distância entre \mathbf{x} e o centro da distribuição $\boldsymbol{\mu}$ ponderada pela matriz de covariância \mathbf{C} . Tal distância é dada por

$$d(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1}(\mathbf{x} - \boldsymbol{\mu}), \quad (4.11)$$

e quanto maior o valor de $d(\mathbf{x})$, maior a chance da amostra \mathbf{x} não ser bem explicada pela distribuição (ou seja, maior a chance de \mathbf{x} não pertencer à distribuição do pavimento).

Embora a abordagem descrita acima seja simples, há algumas condições que complicam sua aplicação na prática. A primeira questão se refere ao conjunto de amostras usadas para calcular os parâmetros $\boldsymbol{\mu}$ e \mathbf{C} da distribuição do pavimento: se tal conjunto contiver amostras que não fazem parte do pavimento (*outliers*), as estimativas da média e matriz de covariância podem ser significativamente alteradas, e a identificação da pista ficará prejudicada. Possíveis *outliers* no contexto desses trabalhos são marcações de pista, sombra, ruído, outros objetos e até obstáculos.

Uma possível solução para esse problema é utilizar o resultado do algoritmo de detecção das bordas da pista, que localiza as marcações existentes nas rodovias, para eliminar tais marcações no cálculo dos parâmetros da distribuição. Tal solução deve funcionar quando as marcações da pista forem a única fonte de *outliers*, mas ainda deve apresentar problemas se houver outros fatores (sombras ou outros objetos).

Uma solução mais abrangente seria detectar que: o conjunto de treinamento usado para estimar os parâmetros da distribuição multivariada contém *outliers*, retirá-los e então estimar os parâmetros. Existem algumas abordagens para esse propósito (PENAE PRIETO, 2001), mas exigindo um custo computacional alto demais para implementação em tempo real.

Para distribuições normais univariadas, há soluções robustas e computacionalmente

eficientes. Por exemplo, a média μ da distribuição pode ser estimada robustamente através da mediana λ , ou seja,

$$\mu \approx \lambda = \text{mediana} \{x_i\}, \quad (4.12)$$

e o desvio padrão σ pode ser estimado através do MAD (*Median Absolute Deviation*) (HUBER, 1981):

$$\sigma \approx 1.4826 \text{ MAD}, \text{ onde } \text{MAD} = \text{mediana} |x_i - \lambda|. \quad (4.13)$$

Para a solução da estimação robusta dos parâmetros de distribuições univariadas, os canais de cor R, G e B serão tratados individualmente, gerando três distribuições Gaussianas univariadas. Em princípio, o uso individual dos canais de cor deve fornecer menos informação do que a análise conjunta dos mesmos (informação essa contida na matriz de covariância), e esse trabalho visa investigar as duas estratégias.

Um outro fator complicador na modelagem do pavimento, é o fato de que existe variação de intensidade dos pixels do asfalto em função da distância com a câmera, devido à perspectiva da câmera. Assim, a região de asfalto mais distante da câmera tem uma resposta diferente da região do pavimento mais próximo da câmera. Para lidar com esse segundo problema, é proposta uma divisão da Região de Interesse em blocos horizontais, assumindo que a distribuição do pavimento dentro de cada um dos blocos é uma gaussiana (possivelmente diferente de um bloco para outro).

Em síntese, a abordagem proposta para o reconhecimento do padrão de pista consiste das seguintes etapas:

1. Obtenção da Região de Interesse obtido no Módulo 1;
2. **Divisão da pista em blocos** horizontais: Uma mesma região da pista terá uma intensidade diferente condicionado a distância que esta região está da câmera, ou seja, a distância da câmera influencia na identificação de um padrão de câmera. Para alavancar a acurácia dos descritores estatísticos a pista foi dividida em blocos horizontais. Assim, dentro de cada bloco são definidos os descritores estatísticos, e a classificação do pixel em pista depende do bloco em que encontra. O efeito percebido é que quanto maior o número de blocos, menor é a variação na intensidade dos pixels dentro do bloco, como efeito colateral, quanto menor o bloco, menor o número de amostras e a presença de *outlier* ou de elementos como sombra, pinturas na pista, geram uma resposta errônea em todo o bloco. A quantidade de blocos deverá ser um parâmetro no sistema que deverá ser testado no decorrer do trabalho;
3. **Segregação de Cores**: O modelo considera a captura da imagem colorida, porém várias são as funções que necessitam a conversão da imagem para tons de cinza (*grayscale*) ou então processar a imagem em um ou mais canais de cores

específicas. Esta segregação é fundamental para a identificação do padrão de pista, isto proporcionará nos experimentos evidenciar a melhor resposta e maior eficiência em um canal de cor específico.

Além da identificação do padrão de pista, a segregação da imagem permite o processamento nas etapas de geração e validação das hipóteses tais como: identificação de retas, identificação de quinas e até mesmo geração de componentes conexos.

4. (Reconhecimento do padrão de pista: Para caracterizar a distribuição dos pixels associados à pista, pretende-se investigar os seguintes modelos estatísticos:
 - (a) Distribuição Gaussiana multivariada, onde cada entrada é o valor RGB do pixel. Para cada bloco definido no item 1 serão calculados o vetor média e a matriz de covariância, usando como amostras os pixels da ROI, eliminando aqueles que correspondem às marcações da pista.
 - (b) Distribuição Gaussiana univariada, onde cada entrada é um dos canais de cor (R,G ou B) do pixel. Para cada bloco definido no item 1 serão calculados a média e o desvio padrão de forma robusta (através da mediana e MAD) para cada um dos canais de cor. Essa abordagem não leva em consideração a correlação entre os canais de cor, mas espera-se que seja mais robusta na estimação dos descritores.
 - (c) Treinamento: Será realizado um treinamento com número parametrizável de quadros sobre o qual serão calculados os descritores estatísticos (média, mediana, MAD, etc). Este treinamento deverá ocorrer ao inicializar o algoritmo, antes de identificar obstáculos. Isto porque a cor e intensidade da pista varia de uma estrada para outra e varia também em função das condições do tempo e luminosidade.

Algumas condições geram a necessidade de re-treinamento tais como:

1. Mudanças na luminosidade: A fase de treinamento ocorre na inicialização do modelo proposto. Porém, pode ocorrer uma mudança nas condições de iluminação, por exemplo, a oclusão do sol em um túnel ou mesmo por uma nuvem. Tal ocorrência afetará a intensidade dos pixels da pista, o que poderá afetar a estimativa da média e covariância do modelo Gaussiano. Uma atualização do treinamento do modelo de pista pode auxiliar na solução desse problema;
2. Mudança do horário: e conseqüentemente da iluminação solar;
3. Mudanças de estrada: De uma estrada para outra, e muitas vezes na mesma estrada, a cor e a intensidade da pista podem mudar. Da mesma forma que mudanças na

luminosidade, este problema pode gerar falhas na detecção da pista e de candidatos a obstáculo, o que também pode ser atenuado com um re-treinamento do modelo;

O resultado do Módulo 2 é um conjunto de descritores estatísticos calculados para os diferentes canais de cor e tons de cinza e para os diferentes blocos da região de interesse. Estas informações são representativas do modelo de pista para o algoritmo de detecção de obstáculos.

4.3 MÓDULO 3: GERAÇÃO DE HIPÓTESES

A Geração de Hipóteses se inicia pela classificação dos pixels em pista e não pista. Cada pixel da ROI será comparado com o modelo da pista, e pixels que diferirem significativamente do modelo serão considerados candidatos a obstáculos. Tendo as distribuições (uni ou multivariadas) para cada bloco da imagem, obtidas usando um conjunto de quadros de treinamento, a distância de Mahalanobis (Equação (4.11)) será utilizada para classificar cada pixel de uma nova imagem de teste como pertencente à pista ou não.

O resultado desse teste será uma imagem binária apenas para identificação preliminar do que é pista e daquilo que não pertence à pista.

As marcações da pista normalmente têm uma intensidade elevada, serão consideradas como não pista e por consequência como candidatos a obstáculo, como ilustrado na Figura 4.7 (em vermelho estão destacados os pixels considerados não pista). São falsos positivos que, embora detectados na fase de geração das hipóteses, irão consumir recursos computacionais, prejudicando o desempenho do validador de hipóteses. A exclusão explícita da marcação (usando o algoritmo de detecção de bordas da pista) ou o uso de estimadores robustos para média e desvio padrão podem auxiliar na solução do problema.



Figura 4.7: Marcações da pista sendo classificadas como candidato a obstáculo

Esta etapa visa justamente filtrar estas marcações de pista, para que não gerem custo computacional nas etapas seguintes do algoritmo que busca por obstáculos. Por saber o posicionamento das marcações da pista, obtido na etapa de identificação das bordas da pista, é possível gerar o reconhecimento do padrão de marcação de pista.

Da mesma forma que o reconhecimento de padrão de pista, o reconhecimento do padrão de marcação de pista considera a intensidade como atributo e seu padrão será comparado apenas para pixels próximos as bordas da pista identificadas. Em se encaixando no padrão de marcação de pista, o pixel será descartado da imagem binária de pista e não pista. Assim, as etapas seguintes terão menor custo computacional, visto o menor volume de pixels a serem avaliados.

A Figura 4.8 ilustra a identificação de pista (em preto) e não pista (em branco). Na Figura a esquerda são percebidas as marcações da pista como não pista, já na Figura a direita as marcações são ignoradas. Isto significa que a sequência do algoritmo não analisará estas regiões da imagem.

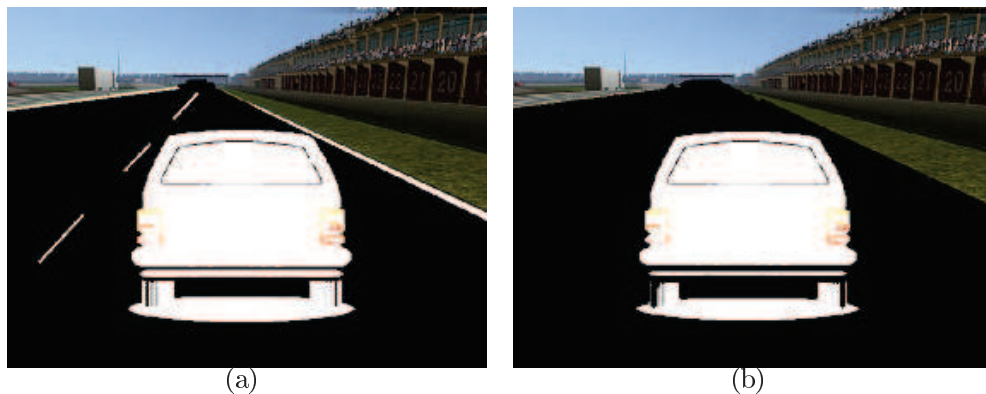


Figura 4.8: Etapas da identificação do padrão de pista: (a) Identificação inicial de pista e não pista (b) Filtro das marcações da pista

A imagem binária resultante das etapas anteriores contém as informações mapeadas de pista e não pista. Porém pela característica do classificador de pista de e não pista, alguns pixels podem estar próximos aos limiares de pista e não pista e não formarem uma estrutura unificada. Para buscar esta homogeneização dos pixels classificados como não pista, o algoritmo aplica operadores morfológico que, primeiramente causam um alargamento (dilatação) das áreas de não pista preenchendo eventuais 'buracos' na estrutura e unificando estruturas conexas muito próximas. Após a dilatação, o algoritmo aplicada a erosão na imagem, para que as bordas externas tomem as dimensões previamente identificadas. Tal sequência é uma operação de fechamento, onde o efeito final deverá ser o preenchimento de espaços entre as estruturas, mas o tamanho final não deverá ser afetado.

Uma vez identificados os padrões de pista e não pista, filtradas as marcações, e unificadas as estruturas próximas o resultado é uma imagem binária com pixels não

candidatos a obstáculos e pixels candidatos a obstáculos. Esta etapa visa somente o agrupamento dos pixels em componentes conexos, chamados de *Blobs*. Cada componente conexo gerado é um possível obstáculo, ou seja, esta a geração de *blobs* finaliza a etapa de geração de hipóteses.

4.4 MÓDULO 4: VALIDAÇÃO DAS HIPÓTESES

Uma vez identificados os componentes conexos, a quantidade destes representa um aumento no custo computacional. Porém, no conjunto de componentes conexos revelados, pode existir componentes, cujo tamanho equivalente em coordenadas de mundo, não tem possibilidade de representar um obstáculo, e que na verdade são elementos considerados não pista, mas que também não são obstáculos.

Esta etapa de medição dos componentes conexos, visa justamente avaliar quais destes componentes pode ser um obstáculo em potencial em função de seu tamanho. Para este fim, são utilizadas as informações da calibragem da câmera visto na sessão 4.2. Os componentes conexos são envoltos por um retângulo de onde são extraídas as coordenadas de imagem dos pontos inferiores direito e esquerdo do retângulo. Estas coordenadas de imagem são convertidas em coordenadas de mundo e pela distância entre os dois pontos em coordenadas de mundo é dado o tamanho, a largura do componente conexo. Só serão avaliados pelo restante do algoritmo os *blobs* com largura superior a 1,5 metros.

Dado um *blob* que representa um possível candidato a obstáculo, a etapa de verificação das hipóteses validará se o *blob* é efetivamente um obstáculo ou não. Conforme visto no capítulo anterior, há uma variedade de abordagens que podem ser empregadas na validação das hipóteses, com taxas de acerto e custos computacionais variados. Este trabalho focará na análise das linhas horizontais e quinas dentro de cada *blob*, além do uso de câmeras calibradas que permitem relacionar o tamanho de um *blob* em pixels (medida de imagem) com seu tamanho real em metros.

Em uma pista é bastante incomum a presença de linhas retas horizontais perpendiculares ao sentido da pista. A identificação destas retas são indícios fortes da presença de obstáculos e por isto se caracteriza como uma hipótese, um candidato a obstáculo. Ao se observar a Figura 4.9 (a) se entende que o veículo pode registrar não apenas uma, mas várias linhas, como para-choque, capô e teto.

A Figura 4.9 evidencia dois momentos do algoritmo, onde primeiramente é avaliado o tamanho do componente conexo, se este excede o valor de 1,5 metros é considerado na etapa seguinte. Na imagem a direita (b), o *blob* é submetido ao detector de retas e a presença destas reforça o candidato a obstáculo como sendo obstáculo.

Para detectar linhas aproximadamente horizontais, dentro dos *blobs* que representam candidatos a obstáculos, será utilizado a Transformada Hough (GONZALEZ e WOODS, 2006), que computa um mapa de votos para todas as possíveis linhas que passam em

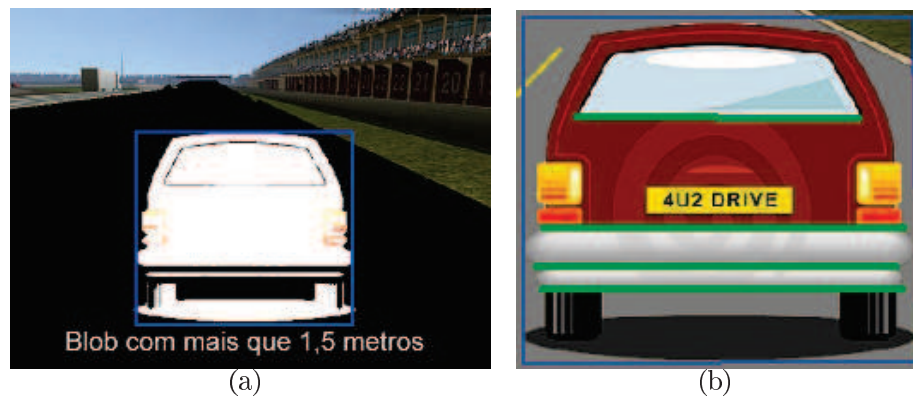


Figura 4.9: Avaliação do tamanho do componente conexo: (a) *Blob* selecionado (b) Identificação de retas

um determinado ponto. Para que não ocorra desperdício de poder computacional, a aplicação de Hough será restrita aos ângulos onde se espera encontrar a reta com um pequeno desvio para permitir flexibilizar a perspectiva da câmera. Sobre o ponto de vista de implementação, este processo demanda primeiramente a detecção de bordas da imagem, que é obtido pela aplicação do algoritmo de Canny. Sobre as bordas detectadas é aplicado o detector de retas de Hough. Tais passos são funções especificadas na biblioteca OpenCV.

Esta etapa tende a minimizar um problema comumente enfrentado que é a projeção de sombras sobre a pista de rolagem. A projeção de sombras sobre a pista altera a intensidade dos pixels, e podem ser identificados com candidatos a obstáculo, sendo assim um falso positivo. A Figura 4.10 evidencia em vermelho sombras sobre pista detectadas como não pista. A não detecção de retas na região indicada será o fator decisório para que tal candidato a obstáculo seja descartado.

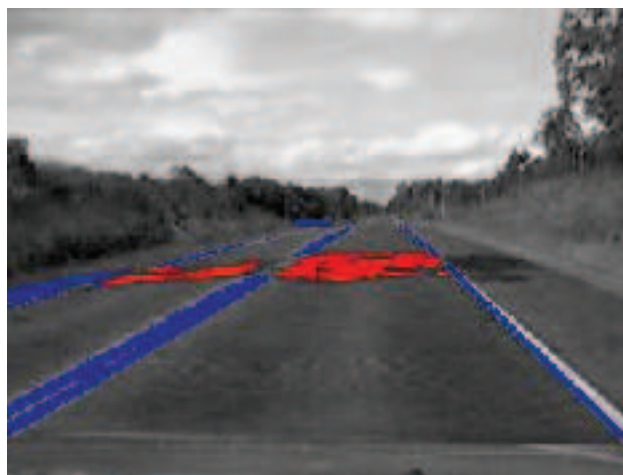


Figura 4.10: Projeção de sobras sobre a pista

Uma vez percebidas linhas aproximadamente horizontais, também é viável avaliar a

existência de quinas na imagem e que sejam encontradas nas extremidades das linhas horizontais. A Figura 4.11 ilustra a presença de quinas na extremidade das linhas horizontais e que reforçam a hipótese de um obstáculo do tipo veículo. A identificação de quinas será realizada utilizando o Operador de Harriss, disponível na biblioteca OpenCV.



Figura 4.11: Projeção de sombras sobre a pista

4.5 PERCORRENDO O ALGORITMO

As seções anteriores deste capítulo procuraram apresentar a estrutura geral do algoritmo desenvolvido. Sua organização modular, a função de cada um dos Módulos e os processos integrantes dos Módulos. Esta sessão percorre o algoritmo do início ao fim, com o objetivo de ilustrar o funcionamento geral do algoritmo.

O ponto de partida do algoritmo é um quadro de imagem, obtida diretamente a partir de uma câmera ou de um arquivo de vídeo previamente gravado. O primeiro quadro é utilizado para fazer a identificação inicial das bordas (marcações da pista). No quadro seguinte e tendo as marcações de pista localizadas, o processo de detecção de bordas da pista apenas faz o seguimento destas marcações, o que é menos custoso do que fazê-lo quadro a quadro localizar as marcações.

Uma vez identificadas as marcações, entra em funcionamento a calibragem da câmera. Este processo se baseia na informação de largura da pista e nas informações da marcação de pista para a calibragem semi-automática. Uma vez tendo os parâmetros da calibragem esta informação é armazenada e aplicada nos Módulos seguintes.

Com a câmera calibrada é possível estimar a pista a esquerda (a pista de contra-mão). O algoritmo armazena dois vetores, um contendo a informação da posição dos pixels mais a esquerda, e outro contendo a posição dos pixels mais a direita da região de interesse delimitada.

Paralelamente a este processo de estimativa da região de interesse, o algoritmo percorre 30 quadros no processo de reconhecimento do modelo de pista. Este reconhecimento considera a segregação dos canais de cores e também a divisão da pista em blocos horizontais. Ou seja, o algoritmo calcula os descritores estatísticos para cada canal de cor, dentro de cada um dos blocos e a armazena em variáveis de sistema para posterior

utilização. Este período, chamado de treinamento de padrão de pista, considera apenas os pixels delimitados pelas bordas da pista, e não considera toda a região de interesse.

A procura por candidato a obstáculos só inicia no quadro 31, no processo de classificação de pista e não pista, cada pixel é comparado com os dados do treinamento de padrão de pista dentro de cada bloco e considerando o canal de cor correspondente. Tudo aquilo que sai do padrão estabelecido e não é considerado marcação de pista, é armazenado em uma matriz que contém a posição de cada um desses pixels. Esta matriz representa os pixels candidatos a obstáculo e também é representada em uma imagem binária.

Antes de ser agrupados em *blobs* ou componentes conexos, esta imagem binária sofre a atuação de operadores morfológicos. Estes operadores visam aproximar e preencher os buracos entre pixels candidatos. Estes buracos podem existir por estarem em um limiar muito próximo ao padrão de pista, mas se este pixel é envolvido por uma série de candidatos, este processo tenta homogenizar a região da imagem.

Logo após a aplicação dos operadores morfológicos os componentes conexos são gerados e apesar de não compor necessariamente uma forma reta, estes componentes são envoltos por uma forma retangular. A Figura 4.12 demonstra em preto o componente conexo detectado, e em vermelho o retângulo que envolve o componente. Este retângulo delimita a fronteira do *blob*. Cada um destes retângulos será a partir deste ponto considerado candidato a obstáculo.



Figura 4.12: Limite do Componente Conexos

No Módulo de validação da hipótese, o primeiro processo é avaliar se o componente conexo tem tamanho suficiente, e até mesmo se ele é maior que um veículo. O algoritmo considera as posições em coordenadas de imagem do pixel inferior direito e inferior esquerdo do componente conexo, aplica as equações de conversão para coordenada de mundo e faz o seguinte teste: se o tamanho for menor que 1,5 metros ou maior que 3,5 metros então descarta o candidato a obstáculo, senão o candidato a obstáculo segue para a avaliação de presença de retas.

Ao aplicar a transformada Hough é possível identificar a presença de retas. Se não houver nenhuma reta, então o candidato é descartado. Na sequência é avaliada a presença de quinas. A ausência de quinas também é indicativo de não ser um obstáculo.

Se o candidato a obstáculo tiver um tamanho condizente, se for detectada a presença de retas e se contiver quinas, então o candidato trata-se de um obstáculo e será alertado ao usuário através de realce visual e da exibição da distância em metros do obstáculo.

Por fim, aplica-se o classificador em *Haar Like Features*. Vale ressaltar que esta etapa é executada em todos os quadros independente de no quadro terem sido descartados todos os candidatos a obstáculos. Se o obstáculo foi identificado pelas etapas anteriores, esta etapa fará o reforço desta classificação. Caso não tenha sido identificado nenhum obstáculo anteriormente, esta etapa ainda poderá fazê-lo.

A Figura 4.13 demonstra como a região de interesse se limita ao plano da pista, dependendo do ângulo e da posição do obstáculo ele poderá estar, pelo menos em parte, fora da região de interesse. A aplicação de *Haar Like Feature* é uma forma de ampliar a ação do detector de obstáculos para além da região de interesse, mitigando a limitação do modelo, em que partes do veículo fica fora da região de interesse.



Figura 4.13: Parte do veículo fora da região de interesse

5 RESULTADOS

Neste capítulo serão apresentados o relato dos resultados obtidos no âmbito experimental. Estes resultados são a comprovação da funcionalidade do algoritmo, além disso, merecem destaque pelo resultado propriamente dito, e por eventuais dificuldades que se apresentaram no curso da pesquisa.

O algoritmo foi implementado utilizando a Linguagem de Programação C/C++ tendo como biblioteca de apoio a Biblioteca OpenCV (BRADSKIE ADRIAN, 2008; NIXONe AGUADO, 2008; GRUPO... , 2009; OpenCV... , 2009; OPENCV... , 2009). Do OpenCV, foram utilizadas as ferramentas para treinamento e testes do classificador AdaBoost para o reconhecimento de automóveis. O algoritmo é capaz de processar as imagens obtidas diretamente da câmera, ou processar arquivos de vídeo quadro a quadro. Os experimentos aqui relatados foram realizados com arquivos de vídeo.

Os resultados serão apresentados na mesma sequência na apresentação da metodologia proposta no Capítulo 4.

5.1 MÓDULO 1: DELIMITAÇÃO DA REGIÃO DE INTERESSE

A identificação da pista de rolagem foi obtida a partir da implementação de algoritmos que consideram as marcações da estrada (as pinturas sobre o asfalto) descrita pelos artigos (JUNGe KELBER, 2005a, 2004b, 2004a, 2005b).

O processo de validação do detector de bordas foi realizado em 4 vídeos distintos. Ao todo são 5 minutos de filmagem com uma resolução de 320x240 pixels, com 25 quadros por segundo. As amostras consistem de cenas contendo: pistas em linhas reta, curvas, trocas de pistas, pistas simples e pistas de múltiplas faixas.

Como o algoritmo é baseado nas marcações da pista e por identificar a borda esquerda e direita da posição do veículo (pista de mão do veículo), as trocas de pista (necessárias em uma ultrapassagem) geram um período de transição em que o carro já esta na nova pista, mas ainda é considerada a marcação da pista anterior. Isto ocorre porque a troca de pista somente é reconhecida após alguns quadros na nova posição do veículo, que reconhece à marcação da borda esquerda na realidade a direita da câmera e compreende que houve uma troca de pista para a esquerda.

A Figura 5.1 mostra uma cena onde o carro provido de câmera está em uma estrada retilínea tendo um veículo em movimento na pista de contra-mão na direção contrária do sentido do carro. Em preto é evidenciada a detecção das bordas da pista, esta linhas em preto se sobrepõe as pinturas da estrada e demarcam a borda esquerda e direita da pista em que o carro provido da câmera esta posicionado.

Na totalidade dos experimentos, foi possível identificar as marcações da pista respeitando-se os pré-requisitos descritos por Jung e Kelber (2004b):



Figura 5.1: Identificação da bordas da Pista

- o veículo deverá estar posicionado em um porção reta da pista;
- o veículo deverá estar alinhado com a pista;
- não deverão constar no primeiro quadro do detector outras estruturas lineares que não as marcações das bordas da pista.

Para a efetiva obtenção da região de interesse é necessário a calibragem da câmera. Para obtenção desta calibragem são necessários especificar dois parâmetros: largura da pista e distância dos segmentos em metros (as marcações pontilhadas da pista). Tais parâmetros são especificados em arquivo de configuração com os nomes de `larg_pista` e `dist_seg` e receberam os seguintes valores: **larg_pista=3.2** e **dist_seg=5**.

Foram realizados experimentos rodando a calibragem em todos os quadros dos vídeos testados. Contudo não houve variação que justificasse este processo. De fato, percebeu-se que as variações dos resultados da calibragem não são grandes o suficiente para gerar impacto nos cálculos de conversão de coordenadas de imagem para coordenadas de mundo. Em função disto, não é necessário que tal procedimento seja rodado em todos os quadros, mas sim apenas uma vez, reduzindo o custo computacional do algoritmo. Como exemplo, a Figura 5.2, demonstra o comportamento típico do calibrador para 5 quadros distintos.

Somente identificar as marcações da pista, não é suficiente para delimitar a região de interesse do detector de obstáculos, pois além da pista onde o veículo esta posicionado (pista de mão) é necessário mapear a pista a esquerda do veículo (pista de contra-mão). Para isto são utilizadas as informações de calibragem da câmera que permitem identificar a distância entre a borda direita e esquerda da mão do veículo, e projetar a mesma distância para a pista da esquerda (pista de contra-mão), conforme descrito na Seção 4.1. O resultado é uma matriz que armazena os limites da esquerda e da direita da região de interesse do detector de obstáculos conforme ilustrado na Figura 5.3, onde a marcação em vermelho é o limite da região de interesse na pista a esquerda do veículo.

Pela Figura 5.3 é possível evidenciar uma das limitações do modelo. O veículo que se encontra na pista a esquerda esta acima da linha vermelha, ou seja, fora da região de

```

CAMERA :
  Altura: 1.12062
  Angulo Y em graus: -5.21383
  Deslocamento lateral da camera: -1.91062
  Dist. focal: 230.136
CAMERA :
  Altura: 1.1395
  Angulo Y em graus: -5.37379
  Deslocamento lateral da camera: -1.93869
  Dist. focal: 244.508
CAMERA :
  Altura: 1.15799
  Angulo Y em graus: -5.64552
  Deslocamento lateral da camera: -1.96655
  Dist. focal: 252.901
CAMERA :
  Altura: 1.16972
  Angulo Y em graus: -5.5731
  Deslocamento lateral da camera: -1.96655
  Dist. focal: 266.457
CAMERA :
  Altura: 1.15421
  Angulo Y em graus: -5.43325
  Deslocamento lateral da camera: -1.94783
  Dist. focal: 262.844

```

Figura 5.2: Resultado da Calibragem



Figura 5.3: Identificação da região de interesse

interesse. Por consequência, não seria tratado no algoritmo desenvolvido. Este problema é tratado na Seção 5.5.

5.2 MÓDULO 2: IDENTIFICAÇÃO DO PADRÃO DE PISTA

Foram explorados três descritores estatísticos para a modelagem de pista, obtidos a partir dos pixels posicionados entre as bordas direita e esquerda da pista de mão:

- Média e Desvio Padrão: teve um bom desempenho, porém a presença de marcações de pista ou qualquer outro elemento gerador de *outlier* distorce a média e o desvio padrão significativamente;
- Mediana e MAD: dentro os descritores estatísticos usados, foi o que obteve melhor desempenho, se diferenciando de média e desvio padrão por não ser penalizada pela presença de *outliers*;

- **Matriz de Covariância:** a intenção com o uso deste descritor foi de combinar os diferentes canais de cor e encontrar um modelo de pista mais robusto. Contudo, o trabalho com matrizes gerou grande sobrecarga computacional e, como consequência, o algoritmo se tornou extremamente lento. Tal lentidão chegou a ponto de inviabilizar a execução do detector de obstáculos em tempo real, custando 0,4 segundos por quadro. Em decorrência disso, não se aprofundou o estudo neste descritor.

Diante destas considerações foi escolhido os descritores Mediana e MAD para a modelagem da pista. Para este descritor, foi realizada uma bateria de testes visando identificar a quantidade n de MADs, denominada de relaxamento, capaz de equilibrar uma boa identificação de não pista com o menor número possível de falsos positivos. Dentro de cada bloco da imagem, cada pixel é comparado com a mediana. Se esta diferença for menor que n MADs, o pixel será classificado como pista. Caso contrário, se a diferença entre o valor do pixel a mediana for maior que n MADs, o pixel será classificado como não pista. Em outras palavras, o valor de n representa a tolerância e a flexibilidade do classificador. Um valor baixo de n tende a classificar menos pixels como pista tornando o classificador mais sensível e sujeito a falsos positivos. Um valor alto classificará mais pixels como pista, entretanto tornará o classificador menos sensível a detecção.

Para medir a eficiência do descritor foram escolhidos 3 quadros, manipulados para conter pista e não pista. Em outras palavras, as pistas nos quadros pintadas de modo a embutir pixels de não pista. Contados os pixels de pista e não pista real e comparado com o resultado obtido pelo classificador ao variar o número de MADs. Os testes realizados variaram o número de MADs entre 1 e 15 MADs. O melhor resultado foi obtido com o 10 MADs.

Uma vez se definindo o descritor estatístico por Mediana e MAD, foram realizados experimentos nos diferentes canais de cor. Os atributos utilizados para avaliar se um pixel pertence a pista foram fixados com o descritor estatístico Mediana e MAD, mas percebeu-se que estes atributos tiveram resposta diferentes, nos diferentes canais de cores (RGB) e *grayscale*.

O mesmo algoritmo, com os mesmos parâmetros e os mesmos vídeos (4 vídeos com resolução de 320x240) foram testados variando apenas o canal de cor. O melhor resultado foi obtido no canal de **cor azul**, com menor número de falsos positivos, sem contudo deixar de identificar os pixels de não pista. A Figura 5.4 demonstra a respostas nos diferentes canais RBG e *grayscale*. Sobre a pista se projeta a sombra de uma árvore, cuja detecção seria um falso positivo. Percebe-se a diferença da resposta de classificação de não pista nos diferentes canais, com vantagem para o azul.

Na Figura 5.5, em vermelho é destacado aquilo que não é pista segundo o classificador. No caso, o ônibus é percebido como não pista. Vale ressaltar que a parte

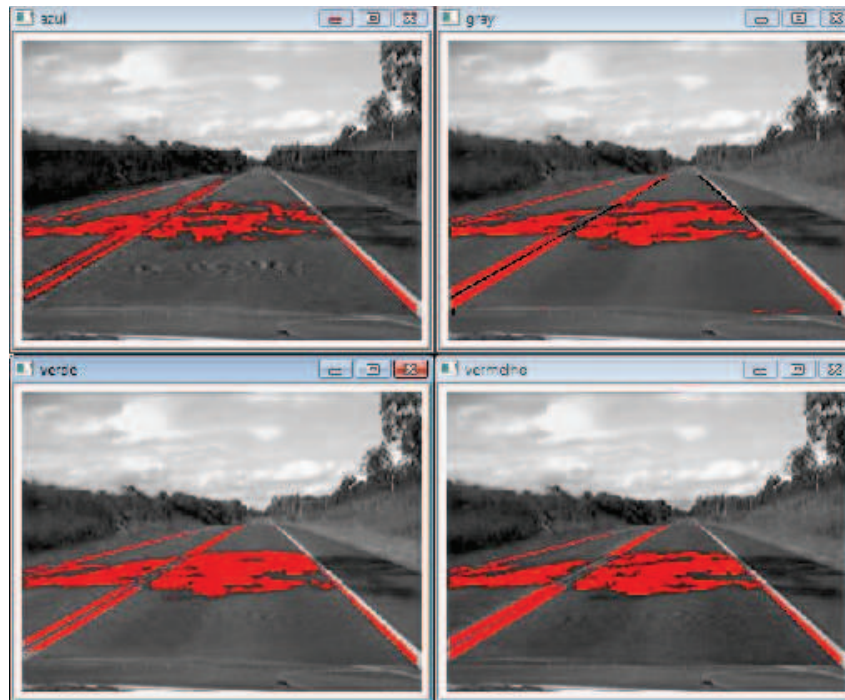


Figura 5.4: Identificação de não pista nos canais RGB e *Grayscale*

superior do ônibus não foi detectada por estar fora da região de interesse selecionada.



Figura 5.5: Identificação de Não Pista

Como é conhecida a região das marcações de pistas (as marcações são necessárias para a delimitação da área de interesse) o algoritmo tem a opção de ignorar estas informações para fins de comparação e para a não marcação como falsos positivos.

Foram realizados testes em dois vídeos, com resolução 320x240 pixels e 25 quadros por segundos e 10 quadros contendo somente pista. O objetivo do teste é identificar a quantidade de blocos capazes de equilibrar a necessidade de segmentação em função da variação da intensidade dos pixels pelo distanciamento da câmera, e que contivesse um número suficiente de amostras. Dos dois vídeos foram realizados treinamentos com 1, 3, 5 e 7 blocos e foram contados os pixels classificados como não pista. Tal resultado depende muito do relaxamento do classificador e da resolução da imagem. Uma resolução maior, necessita de mais blocos, e um relaxamento menor tende a necessitar de menos blocos em função da alta quantidade de amostras e da variação maior no valor dos pixels. Considerando o relaxamento de 10 MADs e os vídeos em questão, o melhor resultado foi obtido com 5 blocos horizontais conforme tabela 5.1.

Tabela 5.1: Ocorrência de falsos positivos (FP) em função do número de blocos (NB)

NB	FP (%)
1	13
3	9
5	5
7	8

A Figura 5.6 ilustra o modelo de cortes horizontais da pista. Na cor laranja são realçadas as fronteiras entre os blocos horizontais. Em preto são as marcações de pista detectadas e em vermelho é o limite esquerdo da região de interesse.

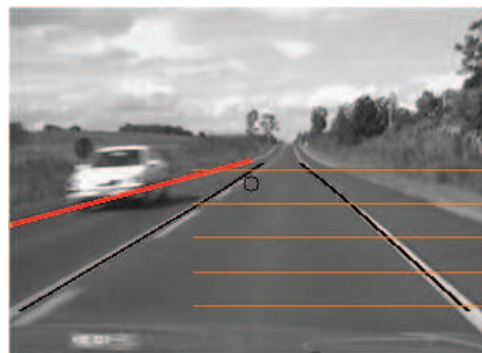


Figura 5.6: Divisão da pista em blocos horizontais

A utilização de NB blocos horizontais resulta em NB modelos (descritores estatísticos) de pistas independentes. Desta independência pode resultar que pixels vizinhos, mesmo que de mesma cor e intensidade, mas de blocos distintos, tenham classificações diferentes de pista e não pista. A Figura 5.7 evidencia uma linha divisória entre dois blocos, sendo que em um, os pixels são destacados em branco por serem considerados não pista, enquanto que no outro, os pixels são considerados pista. A imagem demonstra uma área de recapagem da pista que realmente difere do padrão

original, mas que teve uma diferente classificação nos blocos. A imagem da esquerda é a imagem original capturada convertida para tons de cinza. A imagem da direita é a imagem alterada realçando em branco os pixels considerados não pista.



Figura 5.7: Diferente classificação de pixel para blocos vizinhos

5.3 MÓDULO 3: GERAÇÃO DE HIPÓTESES

Vários são os fatores, além de obstáculos, que afetam a classificação de um pixel em pista ou não, dentre eles as marcações e pinturas na pista. Dentre eles, sombra, reparações no asfalto, entre outros fatores que não justificam a classificação do pixel como candidato a obstáculo.

Todo o pixel que não é considerado pista pelo padrão estabelecido é considerado como não pista e é posicionado em uma imagem binária de mesmas dimensões que a imagem original. A Figura 5.8, a imagem a esquerda é a imagem original capturada, já a imagem a direita demonstra a sombra de um veículo sendo identificada como não pista tendo estes pixels realçados em vermelho na Figura da direita.

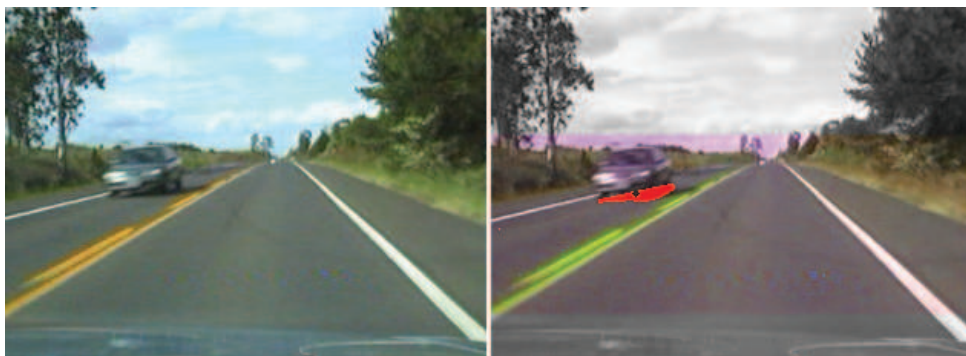


Figura 5.8: Classificação de pista e não pista

A Figura 5.9 destaca em vermelho o conjunto de pixels classificado como não pista. Trata-se de um falso positivo detectado pelo algoritmo.

Sobre a imagem binária de pixels considerados não pixels são aplicadas operações morfológicas de dilatação e erosão, que visam unificar pixels de não pista que estão



Figura 5.9: Classificação de pixels como não pista - falso positivo

muito próximos e que estruturalmente poderiam formar um mesmo bloco, mas foram considerados como pista (falso negativo). Uma vez unificados, o algoritmo gera os componentes conexos (*blobs*) destes conjuntos de pixels.

A Figura 5.10 mostra a delimitação do componente conexo através de um retângulo que envolve os pixels de não pista. Esta região contida pelo retângulo será processada nas etapas seguintes do algoritmo.



Figura 5.10: Delimitação do componente conexo

5.4 MÓDULO 4: VALIDAÇÃO DAS HIPÓTESES

Um componente conexo é delimitado por um retângulo, capaz de conter toda a extensão do componente conexo. O retângulo é medido em coordenadas de mundo e descartado caso não tenha tamanho superior ou igual a 1,5 metro de largura. Esta conversão das coordenadas de imagem em coordenadas de mundo, é possível pela utilização da calibragem da câmera, detalhados nas seções 2.8 e 4.2

Nos testes realizados, a quantidade de componentes conexos de pixels considerados como não pista, foi reduzida em 94%, pois nos vídeos utilizados haviam muitos *outliers*, que eram destacados isoladamente tendo dimensões muito distantes das dimensões de um

veículo que representaria um obstáculo.

Além disso, foram manipulados 20 quadros de um vídeo, inserindo uma forma retangular na cor vermelha com diferentes dimensões. O objetivo foi validar a capacidade do algoritmo em descartar estruturas pequenas e que não deveriam ser consideradas obstáculos. A Figura 5.11 é a representação de um dos quadros manipulados e que em função do tamanho da forma através da calibragem da câmera foi classificado como candidato a obstáculo ou descartado o componente conexo.



Figura 5.11: Teste de descarte de estruturas pequenas na imagem

Assim, as etapas seguintes do algoritmo são acionadas quando realmente há um componente conexo com dimensões representativas da estrutura de obstáculo avaliada pelo algoritmo, ou seja, um possível veículo.

Dentre os componentes conexos com dimensões compatíveis com o tamanho de um veículo podem haver diferentes estruturas que não são verdadeiramente de um obstáculo. Cita-se, por exemplo, a projeção de sombras sobre a pistas, restaurações na pista com cor diferente do asfalto original, pinturas sobre a pista, entre outros. Grande parte destes itens não tem uma formação retilínea na região marcada pelo componente conexo e podem ser descartadas.

A implementação consiste primeiramente, na identificação das bordas da imagem que é obtido pela aplicação de detector de bordas de Canny. A sequência é a aplicação do detector de linhas de Hough. Todas funções específicas da biblioteca OpenCV.

A Figura 5.12 (a) destaca o componente conexo, previamente identificado e delimitado. A Figura 5.12 (b) ilustra o resultado do detector de bordas de Canny, sendo que este detector é aplicado apenas a área correspondente ao componente conexo. A Figura 5.12 (c) mostra o resultado do detector de linhas de Hough, exibindo em verde a linha detectada.

Através desta etapa é possível descartar componentes conexos, onde não é encontrada a presença de retas. Com isto, as sombras que geram falso positivo no detector de obstáculo são descartados. A Figura 5.13 demonstra sequência do componente conexo

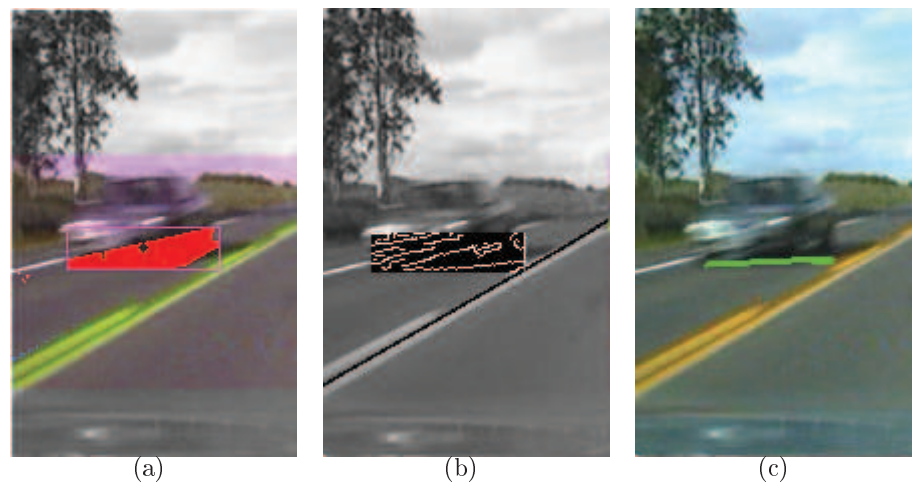


Figura 5.12: Detecção de Retas: (a) Componente Conexo (b) Detector de Bordas de Canny (c) Detector de linhas de Hough

identificado (a), a aplicação do detector de bordas de Canny (b) e por fim a aplicação do detector de retas de Hough (c), o qual não gera qualquer resultado.

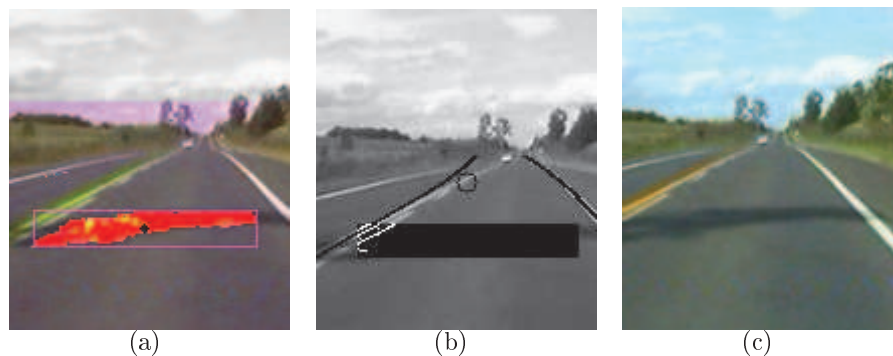


Figura 5.13: Descarte de sombra: (a) Componente Conexo (b) Detector de Bordas de Canny (c) Nenhuma reta detectada

Um elemento previsto na metodologia proposta e que não trouxe o resultado esperado é o detector de quinas. De fato, a aplicação do detector de quinas de Harris para as retas identificadas, sempre levaram a detecção de quinas. Por conseguinte, a detecção de quinas não refutou candidato algum apontado pela detecção de retas, sendo por isto considerado desnecessário e descartado do algoritmo.

5.5 APLICAÇÃO DE *HAAR LIKE FEATURES*

A utilização de *Haar Like Features* apresenta como desafio primeiro a necessidade de treinamento da árvore de decisão. Na biblioteca OpenCV são distribuídas funções específicas para este fim. Para o treinamento é necessário usar um banco de imagens

atendendo aos seguintes requisitos:

1. Imagens Positivas: devem conter tão somente o objetivo sobre o qual se deseja treinar o classificador.
2. Imagens Negativas: não devem conter o objeto a ser identificado;
3. Imagens para teste: imagens que não participarão do treinamento, mas que após a finalização destes serão utilizadas para avaliar a eficiência do treinamento.

No treinamento realizado haviam disponíveis 2.268 arquivos contendo fotos de veículos, sendo estas frontais e da traseira dos automóveis. A Figura 5.14 reproduz algumas das imagens utilizadas como amostras positivas nas sessões de treinamento. Com resolução de 64x64 pixels, as fotos estão armazenadas em tons de cinza. Para ampliar a quantidade de amostras positivas o OpenCV dispõe de mecanismos que duplicam as imagens, alterando a intensidade da Figura em função de uma terceira imagem, com isto foi possível utilizar 5.000 amostras positivas no treinamento. Além da imagem integral do veículo, foi criado um cenário que considerou apenas a metade inferior do veículo, seja na visão frontal ou na visão traseira.

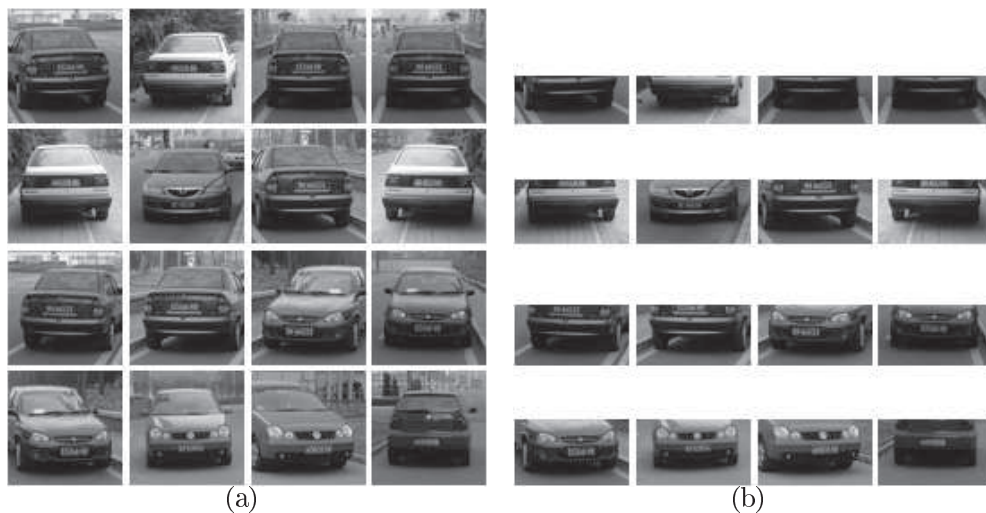


Figura 5.14: Amostras positivas para o treinamento: (a)Imagem inteira (b) Metade inferior da Imagem

Como amostras negativas foram utilizadas mais de 4.000 imagens de arquivo pessoal. Estas imagens foram convertidas para tons de cinza e tinham resolução de 400x600 pixels. De fato, a qualidade do banco de dados tanto de amostras positivas quanto negativas é fundamental para o resultado do treinamento.

As sessões de treinamento executadas consumiram cerca de 70 horas de processamento em um microcomputador de processador Dual Core Intel de 3MHz e

4 GB de memória, sistema operacional Windows XP. A aplicação de treinamento (disponibilizada pelo OpenCV) foi compilada com suporte a múltiplas *threads*.

Foram explorados os seguintes cenários:

- 2.000 Imagens Positivas contendo todo o veículo e 4.000 Imagens negativas;
- 5.000 imagens Positivas contendo todo o veículo e 4.000 Imagens negativas;
- 5.000 imagens Positivas da metade inferior do automóvel e 4.000 imagens negativas.

Uma vez treinado o classificador era testado contra um conjunto de imagens contendo o objeto a ser identificado, no caso um veículo. Para compor um conjunto significativo de amostras para teste, além de um conjunto de 40 imagens totalmente inéditas para o classificador, foram compostas imagens inserindo amostras positivas em amostras negativas e formando um conjunto misto, este também é um recurso disponibilizado pela biblioteca OpenCV. Ao gerar as imagens de teste o OpenCV posiciona a imagem positiva sobre a imagem negativa em uma posição aleatória, porém faz um registro desta posição. Assim ao executar o classificar sobre a imagem é comparada a distância entre posição do veículo inserida da imagem e a posição encontrada, evidenciando assim o nível de acerto na detecção e a quantidade de falsos positivos.

Através destes testes ficou comprovado que o melhor classificador foi o resultado do treinamento de 5000 imagens positivas e 4.000 imagens negativas, com grau de acerto de 58% porém com nenhuma incidência de falso positivo nos testes realizados. A Figura 5.15 é uma demonstração de identificação de um veículo. Tal identificação foi realizada em uma imagem 400x600 em tons de cinza e consumiu 120 ms no microcomputador descrito acima.



Figura 5.15: Teste com árvore de decisão treinada

Além da quantidade do banco de dados das imagens, uma série de parâmetros são necessários para o efetivo treinamento conforme segue:

Usage: c:\OpenCV2.0\bin\opencv_haartraining.exe

```

-data <dir_name>
-vec <vec_file_name>
-bg <background_file_name>
[-bg-vecfile]
[-npos <number_of_positive_samples = 2000>]
[-nneg <number_of_negative_samples = 2000>]
[-nstages <number_of_stages = 14>] % Número
[-nsplits <number_of_splits = 1>]
[-mem <memory_in_MB = 200>]
[-sym (default)] [-nonsym]
[-minhitrate <min_hit_rate = 0.995000>]
[-maxfalsealarm <max_false_alarm_rate = 0.500000>]
[-weighttrimming <weight_trimming = 0.950000>]
[-eqw]
[-mode <BASIC (default) | CORE | ALL>]
[-w <sample_width = 24>]
[-h <sample_height = 24>]
[-bt <DAB | RAB | LB | GAB (default)>]
[-err <misclass (default) | gini | entropy>]
[-maxtreesplits <max_number_of_splits_in_tree_cascade = 0>]
[-minpos <min_number_of_positive_samples_per_cluster = 500>]

```

Para o experimento de melhor resultado foi utilizado a seguinte linha de comando: *haartraining -data haarcascade -vec positives.vec -bg negatives.dat -nstages 20 -nsplits 2 -minhitrate 0.999 maxfalsealarm 0.5 -npos 5000 -nneg 4000 -w 32 -h 16 -mem 800 -mode ALL -nonsym*. Na imagem inteira do veículo utilizou-se parâmetros simétricos para *-w* e *-h*, no caso 24x24. No entanto para as imagens da metade inferior do veículo foram utilizados *-w 32* e *-h 16*.

Um fator que afetou bastante o resultado foi o parâmetro *nonsym*. Inicialmente entende-se que um veículo é sim um objeto simétrico, no entanto ele é simétrico horizontalmente, ou seja, direita e esquerda, mas não é simétrico em sua parte superior e parte inferior.

Na Figura 5.16 foi manipulado um arquivo contendo a imagem de uma pista com dimensões de 320x240 pixels. Nesta imagem foram inseridas duas imagens, uma da frente de um veículo e outra da traseira. Estas imagens foram posicionadas propositalmente fora do que seria considerada a região de interesse pelo algoritmo de detecção de obstáculo. Sobre esta imagem, o detector de veículos treinado identificou e destacou em azul os dois veículos inseridos, em 36 ms.

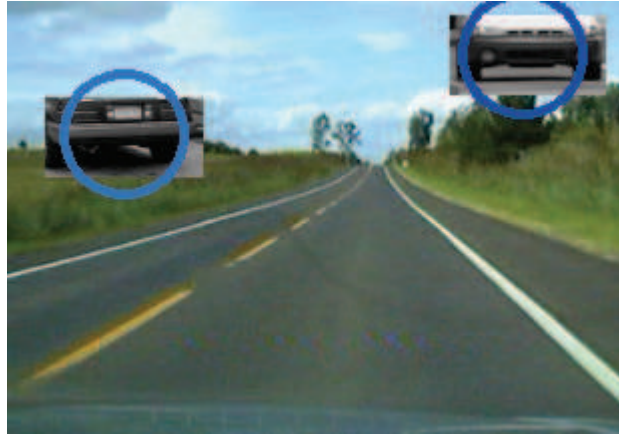


Figura 5.16: Teste em imagem manipulada

Na Figura 5.17 o detector de veículos identifica na imagem da esquerda a traseira de uma caminhonete e na imagem a direita, a frente de um caminhão. Estas identificações, destacadas em azul, são executadas sobre um arquivo de vídeo. Neste quadros de vídeo é possível perceber a presença de outros veículos que não são identificados pelo AdaBoost. Tais veículos não são visualizados integralmente o que aumenta o grau de dificuldade da detecção.

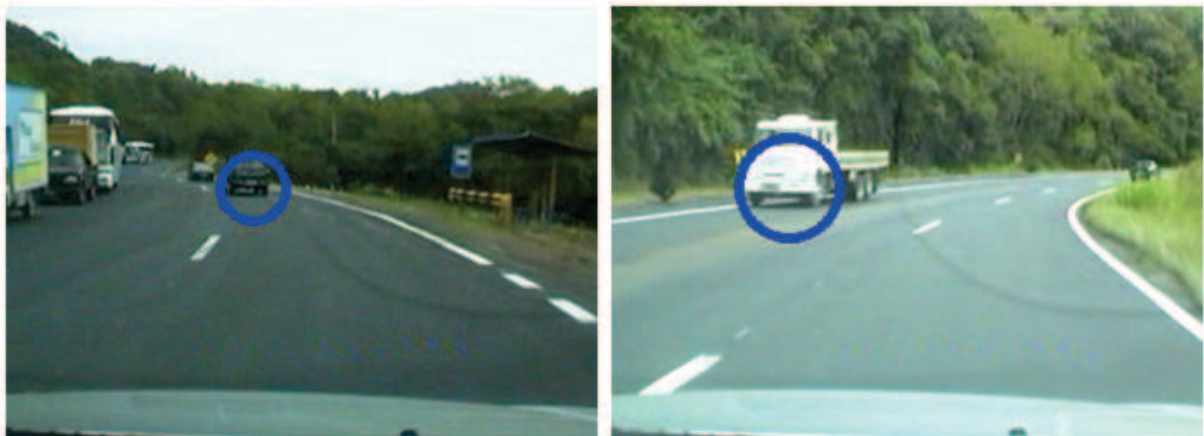


Figura 5.17: Detecções em Pista

Os classificadores foram testados sobre os vídeos já citados neste trabalho. Porém, para ter uma avaliação mais precisa e poder realizar uma comparação entre as árvores treinadas, foi utilizada uma ferramenta do OpenCV denominada `opencv_performance` com a seguinte notação em linha de comando:

```
Usage: opencv_performance
  -data <classifier_directory_name>
  -info <collection_file_name>
  [-maxSizeDiff <max_size_difference = 1.500000>]
```



```

[-maxPosDiff <max_position_difference = 0.300000>]
[-sf <scale_factor = 1.200000>]
[-ni]
[-nos <number_of_stages = -1>]
[-rs <roc_size = 40>]
[-w <sample_width = 24>]
[-h <sample_height = 24>]

```

Ao executar o `opencv_performance` em uma base 262 imagens é possível identificar o nível de acerto da árvore de decisão, seus acertos, suas não identificações e falsos positivos. Além disso o avaliador mede a variação de desempenho em função do número de estágios do classificador e quanto varia de um estágio para outro. O resultado do `opencv_performance` é reproduzido como:

```

+=====+=====+=====+=====+
|          File Name          | Hits |Missed| False|
+=====+=====+=====+=====+
+-----+-----+-----+-----+
|                                Total|   152|   48|   19|
+=====+=====+=====+=====+

```

Number of stages: 20

Number of weak classifiers: 262

Total time: 10.576000

20

152	19	0.760000	0.095000
152	19	0.760000	0.095000
124	1	0.620000	0.005000
96	0	0.480000	0.000000
78	0	0.390000	0.000000
58	0	0.290000	0.000000
40	0	0.200000	0.000000
29	0	0.145000	0.000000
17	0	0.085000	0.000000
10	0	0.050000	0.000000
7	0	0.035000	0.000000
4	0	0.020000	0.000000
3	0	0.015000	0.000000
2	0	0.010000	0.000000
1	0	0.005000	0.000000
1	0	0.005000	0.000000

Na Figura 5.18 é destacada uma cena onde um veículo no primeiro quadro não é identificado, mas alguns quadros depois, dentro da mesma cena, ele é reconhecido pelo classificador AdaBoost. Na imagem a esquerda o veículo não identificado é envolto de um retângulo de linhas pontilhadas em amarelo. Na imagem a direita, o mesmo veículo, mais próximo da câmera é identificado e destacado por um círculo azulado.

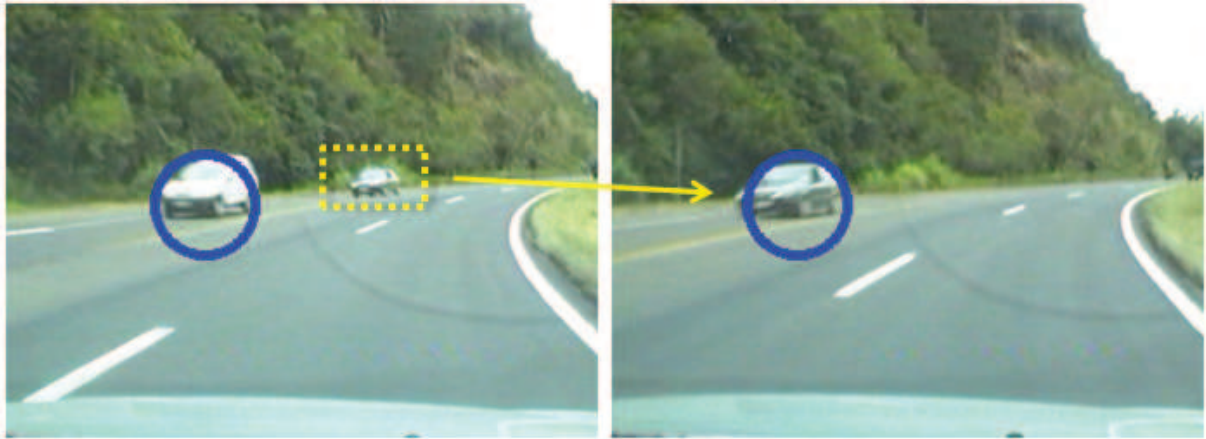


Figura 5.18: Sequencia de cenas

Não foi possível com as árvores treinadas identificar veículos parcialmente oclusos ou que não aparecessem por completo na imagem processada.

A abordagem do Validador de Candidatos (Módulos 1, 2, 3 e 4) é mais rígida e mais rápida, contudo tem as limitações impostas pela restrição da região de interesse, pela necessidade das marcações de pista e pela suscetibilidade na variação de luminosidade e variação de pista. A abordagem do Classificador do tipo Haar com o AdaBoost, é um pouco mais lenta, não faz um reconhecimento de veículos com o mesmo índice de acerto da abordagem do Validador de Candidatos, porém não se limita a região de interesse, não necessita de marcações de pista e não é influenciado pelas variações de intensidade da pista ou do veículo. Ao identificar os veículos esta identificação pode anteceder a identificação pelo Validador de Candidatos, o que representa comunicar antecipadamente ao motorista sobre um obstáculo. São abordagens distintas, mas complementares e que unificadas neste algoritmo. Combinar as informações obtidas via validação de candidatos a obstáculos, com o reconhecimento de veículos feita pelo AdaBoost, incrementando a qualidade dos resultados e da inteligência final do algoritmo.

6 ESTADO DA ARTE

A identificação de obstáculos em uma pista rodoviária é objeto de pesquisa de vários trabalhos acadêmicos, e objeto de desenvolvimento de produtos para uso comercial. A seguir são dados exemplos que contextualizam o presente trabalho.

A Volkswagen tem implementado em alguns de seu veículos, sensores que detectam a presença de obstáculos e inclusive comandam ações de frenagem no veículo (VM...). Tal recurso é chamado de *Automatic Distance Control* e esta presente numa versão do Passat desde 2008. Comercialmente é possível encontrar funções que detectam a saída do veículo da pista, funcionalidade esta, que está presente no mesmo Volkswagen Passat e é chamada de *Lane Hold Assist*.

O fabricante de automóveis Audi, dispõe de um recurso chamado *Audi Lane Assist*, no qual, se o veículo começa a desviar para uma das marcas da pista, o sistema detecta e alerta o motorista através de vibrações no volante, informando que o veículo está se movendo para fora da sua pista. Como no algoritmo proposto, o sistema da Audi (AUDI...), detecta marcações de pista através de uma câmera na parte frontal do veículo. O sistema torna-se operacional logo que a câmara detecta marcas rodoviárias em ambos os lados da pista de mão do veículo. Nenhum aviso é dado quando o condutor está indicando com um sinal de conversão (pisca-pisca). O sistema é projetado para uma condução em auto-estradas e estradas principais, e trabalha em velocidades acima 60 km/h.

A Audi também investiu na detecção de obstáculos através de um recurso chamado *Adaptive Cruise Control*. No modo de Cruzeiro o motorista define a velocidade máxima em que deve trafegar e sua função se restringe a dirigir o carro, sem ter de acelerar ou realizar trocas na marcha do veículo. No caso do *Adaptive Cruise Control*, o sistema mede a distância do veículo à frente por meio de um sensor de radar e controla a velocidade, garantindo que não exceda o valor definido. Além disso, o controle de cruzeiro adaptativo mantém automaticamente uma distância constante do veículo à frente. Uma restrição do sistema é que ele não reage em caso de objetos parados (estacionados). A Figura 6.1 é parte de um vídeo ilustrativo disponibilizado pela Audi que demonstra o funcionamento do controle de cruzeiro adaptativo (AUDI...), onde é possível identificar a presença de outro veículo na mesma pista.

Apesar de eficientes e já presentes em produtos comerciais, estes recursos são caros e por isto estão presentes apenas em carros com elevado valor de mercado, sendo inacessível para a absoluta maioria das pessoas. O método proposto neste trabalho teve como objetivo a simplicidade e completude, favorecendo o seu custo/benefício. De fato, o algoritmo proposto usa apenas um receptor simples (câmera monocular embarcada), e tem a vantagem de prover ao usuário o mesmo apoio dado pelas ferramentas acima citadas, simultaneamente. Isto é, o algoritmo é capaz de realizar, ao mesmo tempo,

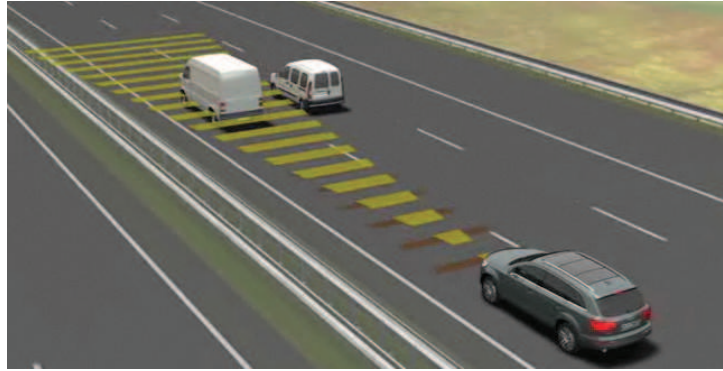


Figura 6.1: Controle de Cruzeiro Adaptativo da Audi

detecções similares ao *Automatic Distance Control* e *Lane Host Assist* da Volkswagen e *Audi Lane Assis* e *Adaptative Cruise Control* da Audi.

Em termos acadêmicos, vários trabalhos foram realizados com o mesmo enfoque. A seguir, serão identificadas as principais convergências e diferenciações destes estudos para com este trabalho.

A estrutura do algoritmo desenvolvido trata separadamente a identificação de candidatos a obstáculos (geração de hipóteses) de suas validações (validação de hipóteses). Tal divisão é baseada nos conceitos de Geração de Hipóteses e Validação de Hipóteses que são tratados por Sun et al. (2006) e também são utilizadas por Han e Han (2009) e por Tsai et al. (2007). O algoritmo desenvolvido utiliza uma série de técnicas que são distribuídas entre as funções de identificação e validação conforme o modelo descrito no Capítulo 4. Contudo, nos trabalhos pesquisados, é possível identificar que uma mesma abordagem é utilizada em um trabalho como Gerador de Hipóteses e em outro, pode ser usada na Validação das Hipóteses.

Exemplo disto, é a utilização de *Haar Like Features*, que em Han e Han (2009) é focada na Geração de Hipóteses. Isto se diferencia do algoritmo proposto neste trabalho e em Stanciulescu et al. (2009), que considera a utilização de *Haar Like Features* como um módulo para validar a hipóteses. Tal diferença de abordagem se reflete no treinamento e no número de falsos positivos que tende a ser muito maior no projeto desenvolvido por Han e Han (2009). No entanto, cabe ressaltar o diferente propósito da utilização de *Haar Like Features*, que é complementada pela determinação da simetria do componente candidato a obstáculo, ou seja, é de se esperar que mesmo com um número maior de falsos positivos, Han e Han (2009) eliminam estes, através da detecção da Simetria.

Posição	Mesma Pista	Pista de Contramão
Taxa de Detecção	91.2%	76.2%

Tabela 6.1: Resultados do trabalho de Han e Han (2009)

Comparar os resultados obtidos pelo algoritmo em relação a outros trabalhos, pode apresentar algumas oscilações, visto que, o conjunto de vídeos testados é diferente, e que, elementos como luminosidade, resolução da imagem e até mesmo velocidade podem interferir no resultado. Os seguintes resultados consideram os 4 vídeos utilizados durante o desenvolvimento do algoritmo e um universo de 35 carros a serem identificados. Na Tabela 6.1 são exibidos os resultados obtidos por Han e Han (2009). O algoritmo desenvolvido apresenta resultados iguais ou superiores aos apresentados por Han e Han (2009).

Já Stanciulescu et al. (2009) utilizou *Haar Like Features* focando na identificação específica do veículo/obstáculo e, por conseguinte, tem uma abordagem mais próxima da utilizada no algoritmo desenvolvido neste trabalho. Contudo, Stanciulescu et al., utiliza somente os recursos de *Haar Like Features*, enquanto o algoritmo desenvolvido neste trabalho utiliza calibragem de câmera, mapeamento de pista e não pista por atributo de cor, entre outros recursos já descritos no Capítulo 4. Com estes recursos adicionais o algoritmo consegue ser mais preciso, principalmente para situações onde o classificador *Haar Like Features* não tem um bom comportamento, como é o caso de veículos parcialmente oclusos ou modelos de automóveis que não se enquadrem na base de treinamento.

O recurso de calibragem da imagem está presente no trabalho de Seki A. Okutomi (2006), e da mesma forma que no algoritmo desenvolvido, procura converter as coordenadas de imagem para coordenadas de mundo, no entanto se utiliza de informações de recursos como Stereo. A utilização de stereo foi descartada por ser um recurso que encarece o valor final da solução, e iria na direção contrária ao objetivo deste trabalho explicitada no Capítulo 1.

No trabalho de Choi (2008) são abordados algoritmos de identificação de sombra e de fluxo óptico. Quanto ao fluxo óptico, este é considerado um recurso interessante, que apesar de ser revisado no Capítulo 2, não foi implementado neste trabalho. Contudo, por acreditar que através do Fluxo Óptico possam haver melhorias no conjunto final do algoritmo, é que ele é proposto como um trabalho futuro no Capítulo 7. Quanto a detecção de sombra de Choi (2008), foi implementado no algoritmo sua detecção através da identificação de não pista.

O trabalho de Huang (2010) visa a identificação e *tracking* (seguir) do veículo. Isto também é uma ideia interessante, contudo, no artigo é considerada a câmara estática (fixa em relação a rodovia), que permite a subtração do fundo. A subtração do fundo não se aplica com a câmera em movimento, uma vez que toda a imagem muda a cada quadro da imagem.

Como se pode ver, de um modo geral os trabalhos acabam por focar no uso de uma ou duas abordagens/ferramentas. Este trabalho se diferencia justamente por agregar uma série de técnicas e de abordagens: calibragem de câmera, classificador *Haar Like Features*, delimitação da região de interesse, identificação das marcações da pista, identificação

de pista e não pista por descritores estatísticos, utilização e medição dos componentes conexos. Tal composição procura somar os resultados das diferentes técnicas, suprimindo assim as deficiências individuais de cada uma delas. Como exemplo disso, no algoritmo desenvolvido, a utilização de *Haar Like Features* é capaz de identificar veículos em situações em que não existam as marcações de pista. Por outro lado, *Haar Like Features* não se mostra eficiente na presença de veículos parcialmente oclusos, mas pelo uso da calibragem da câmera e a identificação de componentes conexos esta situação é mitigada e o obstáculo identificado. Diante do estudo, podemos afirmar que o uso do algoritmo consegue integrar um maior número de técnicas que seriam vantajosas na identificação de obstáculos na pista, o que aumentaria significativamente a segurança dos motoristas, de seus ocupantes e de todos os que trafegam nas rodovias mantendo o baixo custo da solução.

7 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

7.1 CONSIDERAÇÕES FINAIS

Toda iniciativa envolvendo segurança no trânsito é um passo em direção a preservação da vida humana. Aumentar a segurança daqueles que trafegam nas estradas, diminuir o número de acidentes, ou mesmo minimizar a gravidade destes acidentes é justificativa suficientemente nobre para todo esforço de pesquisa e desenvolvimento de ferramentas para apoio ao motorista.

O objetivo principal deste trabalho foi desenvolver um algoritmo de visão computacional capaz de detectar obstáculos, em especial outros veículos. Tal detecção considerou a utilização de uma câmera monocular embarcada em um veículo em movimento e visa ser utilizada como recurso de apoio ao motorista. Este que é o ator principal do trânsito, sendo ao mesmo tempo a maior vítima e o maior causador de acidentes.

Tendo os aspectos de segurança como argumento e o apoio ao motorista como objetivo, o algoritmo foi implementado na linguagem de programação C/C++ fazendo uso da biblioteca gráfica OpenCV (Open Computer Vision Library) e procede a identificação de obstáculos em tempo real.

Ao implementar o algoritmo as informações de calibragem da câmera, apesar de não serem exatas, dão uma noção de grandeza e cria uma relação aceitável entre as coordenadas de imagem e coordenada de mundo. A utilização destas informações de calibragem, para estimar o tamanho em coordenadas de mundo de um candidato a obstáculo, foi muito bem sucedida e permitiu reduzir muito o número de falsos positivos.

A identificação da região de interesse teve seu sucesso baseado na identificação das bordas (marcações da pista). Uma vez identificadas as bordas (*lanes*) da pista de mão, estender a região de interesse para a pista de contra-mão é um passo simples. Como aspecto restritivo de ter a região de interesse limitada ao plano da pista, está o fato de que os veículos se projetam acima do plano da pista e podem ficar fora da região de interesse. Este problema é minimizado pela utilização do classificador AdaBoost.

Para o Mapeamento de Pista e não Pista, a utilização dos descritores estatísticos Mediana e MAD se mostrou mais eficiente que Média e Desvio Padrão, mesmo assim, é um elemento a melhorar no modelo. De fato, persistem uma série de falsos positivos, os quais necessitam ser tratados em outras etapas do algoritmo. Foi observado nos experimentos as diferentes respostas dos classificadores de pista e não pistas nos diferentes canais RGB e *Grayscale*, sendo o melhor canal de cor o canal Azul, sem contudo desqualificar ou inviabilizar os testes sobre os demais canais, visto se tratar de pequena vantagem. A

utilização da matriz de covariância seria uma abordagem que habilitaria a combinação dos diferentes canais, porém foi considerado uma abordagem inviável por consumir excessivo poder computacional e impossibilitar a execução do algoritmo em tempo real. Vale ressaltar que restrições de processamento computacional são circunstâncias temporais e que pela evolução natural da tecnologia de computadores logo será superada.

A identificação de retas utilizando a transformada *Hough* gerou um contribuição positiva para o resultado final do algoritmo. A presença da reta é forte indicador de obstáculo e sua ausência permite a eliminação de falsos positivos como a projeção de sombras de árvores sobre a pista.

Ao contrário da identificação de retas, a identificação de quinas não colaborou nem com o reforço positivo na confirmação de um obstáculo e nem para a eliminação de falsos positivos;

A treinamento da árvore de decisão foi algo bastante trabalhoso no desenvolvimento do algoritmo, e por vezes não produzia resultado algum. Porém os resultados finais obtidos foram animadores e a utilização de *Haar Like Features* agregou valor funcional ao modelo, resolvendo restrições impostas pela limitação da região de interesse. Além disso, este elemento se utiliza da abordagem baseada em modelo e difere das outras abordagens utilizadas no algoritmo sendo um elemento complementar.

A composição final do algoritmo permitiu a obtenção de bons resultados para o apoio ao motorista. Apresentou poucos falsos positivos, pequena incidência de não detecções, mas prevalecendo um bom desempenho na identificação de obstáculos. A diversidade das técnicas testadas e implementadas, sempre manteve o foco em algo que pudesse ser operado em tempo real, e tal premissa foi mantida e bem atendida.

7.2 TRABALHOS FUTUROS

Ao passo que este trabalho é finalizado, vale ressaltar que durante seu desenvolvimento foram percebidas uma série de oportunidades de melhorias, que deverão ser consideradas como próximos passos para a construção de um detector de obstáculo mais robusto.

Como trabalhos futuros, propõem-se intensificar esforços na utilização de *Haar Like Features*, para isto serão trabalhados seguintes pontos para treinamento do AdaBoost:

- Ampliar o banco de dados de imagens frontal e traseiras de veículos;
- Realizar treinamentos distintos com somente imagens frontais e outro com imagens traseiras. Uma vez treinado aplicar duas vezes o detector com as distintas árvores uma consecutiva a outra;
- Realizar treinamentos específicos como apenas a parte superior dos veículos para identificar janelas e para-brisas;

- Treinar apenas a metade inferior direita e inferior esquerda dos veículos, para identificar veículos oclusos parcialmente.

Um nova abordagem para modelar pista e não pista poderia utilizar as informações de textura (KALINKE et al., 1998) da pista em complemento ao descritor Mediana e MAD utilizado atualmente. A utilização da textura também pode ser explorada para a modelagem do veículo.

Até o momento optou-se por utilizar apenas uma câmera monocular, visto que outros sensores como radar, laser são sensores de maior monta financeiro. Recomenda-se a reavaliação da utilização destes sensores, bem como a adição de mais uma câmera em uma posição diferente do veículo para assim combinar duas visões e poder tirar conclusões mais precisas. Integrar o algoritmo com as informações de velocidade do carro (obtidas ou pelo velocímetro ou via GPS) permitirão estimar eventual tempo de colisão com o obstáculo detectado. Integrar com informações de mapas e GPS poderá posicionar o veículo e considerar informações como a presença de curvas e perímetro urbano.

A implementação de Fluxo Óptico é um recurso que pode ser explorado para identificar o sentido dos obstáculos e eventuais rotas de colisão. Pois considerando uma sequência de imagens variando no tempo, é possível obter, um vetor de velocidade para cada pixel. Ao passo que uma determinada região apresentar uma região de pixels cujo vetor velocidade seja destoante do conjunto geral da imagem, pode ser um sinalizador de obstáculos.

Uma série de avaliações realizadas durante o trabalho considerou os canais RGB e tons de cinza, uma possível abordagem em trabalhos é a avaliou o espaço cromático HSV.

A realização deste trabalho, vinculada a um termo de inter cooperação entre PUC-RS e UNISINOS, e patrocinado pela FINEP, tem boas perspectivas de continuação. Disto, possivelmente resultarão desenvolvimento de um protótipo, bem como artigos para publicação.

BIBLIOGRAFIA

ABCR - Associação Brasileira de Concessionárias de Rodovias. Disponível em:
<http://www.abcr.org.br/publi/seminario_pdf/0206.pdf>.

AUDI Glossary. Disponível em:
<<http://www.audi.com/com/brand/en/tools/advice/glossary.browser.html>>.

BARRON, J. L. et al. Performance of optical flow techniques. *International Journal of Computer Vision*, v. 12, n. 1, p. 43–77, 1994. Disponível em:
<<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.5313>>.

BERTOZZI, M.; BROGGI, A.; CASTELLUCCIO, S. A real-time oriented system for vehicle detection. *J. Syst. Archit.*, Elsevier North-Holland, Inc., New York, NY, USA, v. 43, n. 1-5, p. 317–325, 1997. ISSN 1383-7621.

BONES, C. C.; BRUNO, O. M. Análise do desempenho da transformada de hough paralela em arquiteturas de memória compartilhada. *INFOCOMP Journal of Computer Science*, v. 6, n. 3, p. 75–82, 2007.

BRADSKI, G.; ADRIAN, K. *Learning OpenCV - Computer Vision with OpenCV Library*. : O´REILLY, 2008. 555 p.

CANNY, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 8, n. 6, p. 679–698, November 1986. ISSN 0162-8828. Disponível em:
<<http://portal.acm.org/citation.cfm?id=11275>>.

CHOI, J. *Realtime On-Road Vehicle Detection with Optical Flows and Haar-like feature detector*. 2008.

DAER. *Estatística de Acidentes Daer*. 2009. Disponível em:
<http://www.daer.rs.gov.br/geral_1991_2003.pdf>. Acesso em: 20/03/2009.

DAVIES, E. *Machine Vision: Theory, Algorithms, Practicalities, Third Edition*. : Elsevier, 2005.

FACE Detection Research Methodology - Chapter 3.

http://ycode.org/index.php?option=com_content&view=article&id=9:face-detection-research-methodology
Disponível em: <http://ycode.org/index.php?option=com_contentview=articleid=9:face-detection-research-methodology-chapter-3catid=3:javaItemid=14>.

FREUND, Y. A decision theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, no. 55., 1997.

GAO, D. et al. Automatic detection technique of preceding lane and vehicle. *Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on*, v. 1, p. 408–412, 2008.

GIACHETTI, A.; CAMPANI, M.; TORRE, V. The use of optical flow for the autonomous navigation. In: *ECCV '94: Proceedings of the third European conference on Computer vision (vol. 1)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1994. p. 146–151. ISBN 3-540-57956-7.

GOERICK, C.; NOLL, D.; WERNER, M. Artificial neural networks in real-time car detection and tracking applications. *Pattern Recogn. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 17, n. 4, p. 335–343, 1996. ISSN 0167-8655.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

GRAEFE, V.; REGENSBURGER, U. Ua novel approach for the detection of vehicle on freeways by real-time vision. *Intelligent Vehicles*, 1996.

GRUPO OpenCV Yahoo. 2009. Disponível em:
<<http://tech.groups.yahoo.com/group/OpenCV/>>. Acesso em: 20/03/2009.

HAARTRAINING. *Tutorial: OpenCV haartraining (Rapid Object Detection With A Cascade of Boosted Classifiers Based on Haar-like Features) - Naotoshi Seo*. <http://note.sonots.com/SciSoftware/haartraining.html>. Disponível em:
<<http://note.sonots.com/SciSoftware/haartraining.html>>.

HAN, S.; HAN, Y. Vehicle detection method using haar-like feature on real time system. *World Academy of Science, Engineering and Technology*, p. 5, 2009.

HANDMANN, U. et al. *An Image Processing System for Driver Assistance*. 1998.

HARRIS, C.; STEPHENS, M. A combined corner and edge detection. In: *Proceedings of The Fourth Alvey Vision Conference*. 1988. p. 147–151. Disponível em:
<http://www.csse.uwa.edu.au/pk/research/matlabfns/Spatial/Docs/Harris/A_Combined_Corner_a>.

HUANG, L. Real-time multi-vehicle detection and sub-feature based tracking for traffic surveillance systems. *Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference on*, IEEE Press, Piscataway, NJ, USA, p. 324 – 328, 2010. ISSN 1948-3414.

- HUBER, P. J. *Robust Statistics*. New York: John Wiley and Sons, 1981.
- ITO, K. T.; YAMADA, K. N. Understanding driving situations using a network model. *Intelligent Vehicles*, 1995.
- JEONG, P.; NEDEVSCHI, S. Obstacle detection based on the hybrid road plane under the weak calibration conditions. In: . 2008. p. 446–451. ISSN 1931-0587.
- JIANG, F.; LIN, X. A learning based approach for vehicle detection. In: . Hong Kong: , 2006. p. 1 – 4. ISBN 1-4244-0548-3.
- JONES, W. D. Building safer cars. *IEEE Spectr.*, IEEE Press, Piscataway, NJ, USA, v. 39, n. 1, p. 82–85, 2002. ISSN 0018-9235.
- JUNG, C. R.; KELBER, C. R. A lane departure warning system based on a linear-parabolic lane model. In: *IEEE Intelligent Vehicles Symposium*. Parma, Italy: , 2004.
- JUNG, C. R.; KELBER, C. R. A robust linear-parabolic model for lane following. *SiBGRAPI*, 2004.
- JUNG, C. R.; KELBER, C. R. An improved linear-parabolic model for lane following and curve detection. *SIBGRAPI*, 2005.
- JUNG, C. R.; KELBER, C. R. A lane departure warning system using lateral offset with uncalibrated camera. In: *IEEE Conference on Intelligent Transportation Systems*. 2005.
- JUNG, C. R.; SALDANHA, J. S.; JR., L. G. d. S. Vehicle detection and recognition in greyscale imagery. In: *Proceedings of the XI Symposium on Virtual and Augmented Reality*. 2009. p. 29–36.
- KALINKE, T.; TZOMAKAS, C.; SEELEN, W. V. A texture-based object detection and an adaptive model-based classification. In: *in Procs. IEEE Intelligent Vehicles Symposium*. 1998. p. 341–346.
- KASPRZAK, L.; NIEMANN, H. *Applying a Dynamic Recognition Scheme for Vehicle Recognition in Many Object Traffic Scenes*. : IAPR Workshop on Machine Vision Application, 1996.
- KOSTIA, R. Video-Based traffic monitoring at day and night. In: . St. Louis, MO, USA: , 2009.

LIENHART, R.; KURANOV, E.; PISAREVSKY, V. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. In: *In DAGM 25th Pattern Recognition Symposium*. 2003. p. 297–304.

MAROLA, G. Using symmetry for detecting and locating objects in a picture. *Comput. Vision Graph. Image Process.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 46, n. 2, p. 179–195, 1989. ISSN 0734-189X.

MATTHEWS, N. et al. Vehicle detection and recognition in greyscale imagery. In: *2nd Int. Workshop on Intelligent Autonomous Vehicles*. 1995. p. 1–6. Disponível em: <<http://eprints.ecs.soton.ac.uk/366/>>.

MORAVEC, H. P. *Obstacle avoidance and navigation in the real world by a seeing robot rover*. Tese (Doutorado) — Stanford University, Stanford, CA, USA, 1980.

MORPHOLOGY. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>. Disponível em: <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/morops.htm>>.

NIXON, M.; AGUADO, A. *Feature Extraction & Image Processing*. : Elsevier, 2008. 399 p.

NOLL, D.; WERNER, M.; SEELEN, W. von. Real-time vehicle tracking and classification. In: *IEEE Intelligent Vehicles Symposium (IV 1995)*. 1995. p. 101–106.

OPENCV - Image Processing and Computer Vision Reference Manual. 2009. Disponível em: <http://www.tecgraf.puc-rio.br/~malf/opencv/ref/opencvref_cv.htm>. Acesso em: 20/03/2009.

OpenCV Wiki. 2009. Disponível em: <<http://opencv.willowgarage.com/wiki/Welcome>>. Acesso em: 20/03/2009.

PEDESTRIAN Data. <http://cbcl.mit.edu/software-datasets/CarData.html>. Disponível em: <<http://cbcl.mit.edu/software-datasets/CarData.html>>.

PEDRINI, H.; SCHWARTZ, W. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. : Thomson Learning, 2007. 528 pp. p. ISBN 978-85-221-0595-3.

PENA, D.; PRIETO, F. Multivariate outlier detection and robust covariance matrix estimation. *Technometrics*, v. 43, p. 286–310, 2001.

RENAEST. *Registro Nacional de Acidentes e Estatísticas de Trânsito*. 2009. Disponível em: <<http://www2.cidades.gov.br/renaest/detalheNoticia.do?noticia.codigo=115>>. Acesso em: 20/03/2009.

- RUBESAM, A.; DIAS, R. *Estimação não paramétrica aplicada a problemas de classificação via bagging e boosting*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, SP, 2004. Instituto de Matematica, Estatística e Computação Científica.
- SCHNEIDERMAN, H. A statistical approach to 3d object detection applied to faces and cars. In: . 2000. p. 0–6.
- SEARA, D. M. *Algoritmos para detecção de bordas*. Florianópolis, 1998. Disponível em: <<http://www.inf.ufsc.br/visao/1998/seara/index.html#3>>.
- SEKI A. OKUTOMI, M. Robust obstacle detection in general road environment based on road extraction and pose estimation. *‘Intelligent Vehicles Symposium, 2006 IEEE*, p. 437 – 444, 09 2006.
- SIMULADOR de Canny. 2010. Disponível em: <<http://matlabserver.cs.rug.nl>>. Acesso em: 23/02/2010.
- SMITH, S. M.; BRADY, J. M. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, v. 23, p. 45–78, 1995.
- STANCIULESCU, B.; BREHERET, A.; MOUTARDE, F. Introducing new adaboost features for real-time vehicle detection. *CoRR*, abs/0910.1293, 2009.
- SUN, Z.; BEBIS, G.; MILLER, R. Quantized wavelet features and support vector machines for on-road vehicle detection. In: *Seventh International Conference on Control, Automation, Robotics and Vision*. 2002. p. 1641–1646.
- SUN, Z.; BEBIS, G.; MILLER, R. On-road vehicle detection using optical sensors: A review. In: *In IEEE International Conference on Intelligent Transportation Systems*. 2004. p. 585–590.
- SUN, Z.; BEBIS, G.; MILLER, R. On-road vehicle detection: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 28, n. 5, p. 694–711, 2006. Disponível em: <<http://dx.doi.org/10.1109/TPAMI.2006.104>>.
- SUN, Z. et al. A real-time precrash vehicle detection system. In: *IEEE International Workshop on Application of Computer Vision*. 2002. p. 171–176.
- TRANSFORM, H. *Computer Vision Website Header*. 2009. Disponível em: <http://users.ecs.soton.ac.uk/msn/book/new_demo/hough/>. Acesso em: 20/03/2009.

TSAI, L.; HSIEH, J.; FAN, K. Vehicle detection using normalized color and edge map. v. 16, n. 3, p. 850–864, March 2007.

VALE, G. M.; GALVANIN, E. A.; POZ, A. P. D. O detector de canny-edp: Uma combinação entre as teorias de canny e de difusão anisotrópica não linear. *Revista Brasileira de Cartografia*, n. 56, p. 156 – 168, 2004.

VALE, G. M.; POZ, A. P. D. Processo de detecção de bordas de canny. *Boletim de Ciências Geodésicas*, v. 8, n. 2, p. 67 – 78, 2002.

VIOLA, P.; JONES, M. J. Robust real-time face detection. *Int. J. Comput. Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 57, n. 2, p. 137–154, 2004. ISSN 0920-5691.

VIOLA, P. A.; JONES, M. J. Rapid object detection using a boosted cascade of simple features. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Kauai, HI: IEEE Computer Society, 2001. p. 511–518. ISBN 0-7695-1272-0.

VM Passat CC. Disponível em:

<<http://sempreon.blogspot.com/2008/04/vw-passat-cc.html>>.

WIKIPEDIA Wavelet. Disponível em: <<http://en.wikipedia.org/wiki/Wavelet>>.

WU, V. K. X. Top 10 algorithms in data mining. knowledge and information systems. *EuroCOLT 95: Proceedings of the Second European Conference on Computational Learning Theory*, London, UK, p. 1–37, 2008.