

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

THIAGO WASZAK ALVES

**SISTEMA DE DETECÇÃO EM TEMPO
REAL DE FAIXAS DE SINALIZAÇÃO DE
TRÂNSITO PARA VEÍCULOS
INTELIGENTES UTILIZANDO
PROCESSAMENTO DE IMAGEM**

Porto Alegre
2017

THIAGO WASZAK ALVES

**SISTEMA DE DETECÇÃO EM TEMPO
REAL DE FAIXAS DE SINALIZAÇÃO DE
TRÂNSITO PARA VEÍCULOS
INTELIGENTES UTILIZANDO
PROCESSAMENTO DE IMAGEM**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Engenharia da Computação

ORIENTADOR: Prof. Dr. Altamiro A. Susin

Porto Alegre
2017

THIAGO WASZAK ALVES

**SISTEMA DE DETECÇÃO EM TEMPO
REAL DE FAIXAS DE SINALIZAÇÃO DE
TRÂNSITO PARA VEÍCULOS
INTELIGENTES UTILIZANDO
PROCESSAMENTO DE IMAGEM**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Altamiro A. Susin, UFRGS
Doutor pelo Instituto Nacional Politécnico de Grenoble – Grenoble, França

Banca Examinadora:

Prof. Dr. Tiago Roberto Balen, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Cláudio Rosito Jung, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Prof. Dr. Ivan Müller, UFRGS
Doutor pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

Coordenador do PPGEE: _____
Prof. Dr. Valner João Brusamarello

Porto Alegre, Março de 2017.

DEDICATÓRIA

Dedico este trabalho a minha mãe, em especial pela dedicação e apoio em todos os momentos difíceis.

AGRADECIMENTOS

Agradeço, primeiramente, a minha mãe Zorane pelo suporte durante essa caminhada e aos demais familiares pelo apoio.

Aos professores, pesquisadores, doutorandos, mestrandos e graduandos do LaPSI pelas contribuições e debates nos diversos trabalhos que desenvolvemos juntos e aos colegas e professores do PPGEE pelos conhecimentos e experiências e todos que de alguma maneira foram importantes para a realização desse trabalho.

Ao meu orientador e amigo Prof. Dr. Altamiro Amadeu Susin, pela acolhida, confiança, orientação e incentivo dados durante esses dois anos de trabalho, dos quais este trabalho não teria sido possível.

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização do trabalho de pesquisa na área de Processamento de Sinais. À CAPES pela provisão de bolsa de mestrado, permitindo total dedicação ao trabalho.

*“Eu sou atormentado por uma vontade insaciável de tudo
aquilo que é remoto, amo navegar em mares proibidos.”
(Moby Dick, Herman Melville)*

RESUMO

A mobilidade é uma marca da nossa civilização. Tanto o transporte de carga quanto o de passageiros compartilham de uma enorme infra-estrutura de conexões operados com o apoio de um sofisticado sistema logístico. Simbiose otimizada de módulos mecânicos e elétricos, os veículos evoluem continuamente com a integração de avanços tecnológicos e são projetados para oferecer o melhor em conforto, segurança, velocidade e economia.

As regulamentações organizam o fluxo de transporte rodoviário e as suas interações, estipulando regras a fim de evitar conflitos. Mas a atividade de condução pode tornar-se estressante em diferentes condições, deixando os condutores humanos propensos a erros de julgamento e criando condições de acidente. Os esforços para reduzir acidentes de trânsito variam desde campanhas de re-educação até novas tecnologias. Esses tópicos têm atraído cada vez mais a atenção de pesquisadores e indústrias para Sistemas de Transporte Inteligentes baseados em imagens que visam a prevenção de acidentes e o auxílio ao seu motorista na interpretação das formas de sinalização urbana.

Este trabalho apresenta um estudo sobre técnicas de detecção em tempo real de faixas de sinalização de trânsito em ambientes urbanos e intermunicipais, com objetivo de realçar as faixas de sinalização da pista para o condutor do veículo ou veículo autônomo, proporcionando um controle maior da área de tráfego destinada ao veículo e prover alertas de possíveis situações de risco.

A principal contribuição deste trabalho é otimizar a forma como as técnicas de processamento de imagem são utilizadas para realizar a extração das faixas de sinalização, com o objetivo de reduzir o custo computacional do sistema. Para realizar essa otimização foram definidas pequenas áreas de busca de tamanho fixo e posicionamento dinâmico. Essas áreas de busca vão isolar as regiões da imagem onde as faixas de sinalização estão contidas, reduzindo em até 75% a área total onde são aplicadas as técnicas utilizadas na extração de faixas.

Os resultados experimentais mostraram que o algoritmo é robusto em diversas variações de iluminação ambiente, sombras e pavimentos com cores diferentes tanto em ambientes urbanos quanto em rodovias e autoestradas. Os resultados mostram uma taxa de detecção correta média de 98,1%, com tempo médio de operação de 13,3 ms.

Palavras-chave: Detecção de faixas de sinalização, Transformada de Hough, Veículos Inteligentes, Filtro de Kalman, RANSAC.

ABSTRACT

Mobility is an imprint of our civilization. Both freight and passenger transport share a huge infrastructure of connecting links operated with the support of a sophisticated logistic system. As an optimized symbiosis of mechanical and electrical modules, vehicles are evolving continuously with the integration of technological advances and are engineered to offer the best in comfort, safety, speed and economy.

Regulations organize the flow of road transportation machines and help on their interactions, stipulating rules to avoid conflicts. But driving can become stressing on different conditions, leaving human drivers prone to misjudgments and creating accident conditions. Efforts to reduce traffic accidents that may cause injuries and even deaths range from re-education campaigns to new technologies. These topics have increasingly attracted the attention of researchers and industries to Image-based Intelligent Transportation Systems that aim to prevent accidents and help your driver in the interpretation of urban signage forms.

This work presents a study on real-time detection techniques of traffic signaling signs in urban and intermunicipal environments, aiming at the signaling lanes of the lane for the driver of the vehicle or autonomous vehicle, providing a greater control of the area of traffic destined to the vehicle and to provide alerts of possible risk situations.

The main contribution of this work is to optimize how the image processing techniques are used to perform the lanes extraction, in order to reduce the computational cost of the system. To achieve this optimization, small search areas of fixed size and dynamic positioning were defined. These search areas will isolate the regions of the image where the signaling lanes are contained, reducing up to 75% the total area where the techniques used in the extraction of lanes are applied.

The experimental results showed that the algorithm is robust in several variations of ambient light, shadows and pavements with different colors, in both urban environments and on highways and motorways. The results show an average detection rate of 98.1%, with average operating time of 13.3 ms.

Keywords: Lane Detection, Hough Transform, Intelligent Vehicles, Kalman Filter, RANSAC.

LISTA DE ILUSTRAÇÕES

Figura 1:	Ilustração do sistema de permanência na pista da Audi (AUDI, 2016).	28
Figura 2:	Ilustração do assistente de detecção de faixas de sinalização da Volkswagen (CARICOS, 2012).	29
Figura 3:	Ilustração de uma rua em uma área urbana de Pasadena (CA), que mostra alguns dos desafios que devem ser superados pelos sistema de detecção de faixa de sinalização (ALY, 2008).	30
Figura 4:	Ilustração da visão do sistema Autopilot da Tesla (JEN, 2015).	34
Figura 5:	Ilustração da visão do sistema autônomo da Google (WOOLLASTON, 2013).	35
Figura 6:	Ilustração da funcionalidades prevista na plataforma para veículos inteligentes lançados pela Intel (INTEL, 2016).	36
Figura 7:	Ilustração do conceito de pixel e da convenção do sistema de coordenadas para representação de imagens digitais (PEDRINI; SCHWARTZ, 2007).	41
Figura 8:	Ilustração do procedimento de conversão dos canais RGB de uma imagem colorida para uma imagem em escala de cinza. Adaptado de (DILMEN, 2012).	42
Figura 9:	Ilustração da posição da câmera na parte externa do veículo (WOOLLASTON, 2013).	43
Figura 10:	Ilustração da posição da câmera na parte interna do veículo (INAVI, 2015).	43
Figura 11:	Ilustração do processo de aquisição de imagens digitais (THOME, 2004).	43
Figura 12:	Ilustração da primeira região de interesse (ROI-1) que isola a região da imagem onde a pista está contida utilizada no banco de dados da Caltech. Adaptado de (ALY, 2008).	45
Figura 13:	Ilustração da primeira região de interesse (ROI-1) que isola a região da imagem onde a pista está contida utilizada no banco de dados da UFRGS. Adaptado de (PAULA, 2015).	45
Figura 14:	Ilustração do funcionamento de um filtro de mediana em uma imagem ruidosa. Adaptado de (WHATWHENHOW, 2016).	46
Figura 15:	(1) A imagem da pista na escala de cinza e filtrada por um filtro de mediana e (b) A imagem de saída da transformação de perspectiva (<i>Bird's Eye</i>).	47
Figura 16:	Ilustração dos planos da imagem que cortam o mesmo feixe de raios (PRINCE, 2011).	48

Figura 17:	Ilustração da transformação de um quadrilátero de um sistema de referência em um quadrilátero correspondente em outro sistema de referência. Adaptado de (CORRMAP, 2013).	48
Figura 18:	As figuras da coluna da esquerda (A) ilustração os 4 primeiros pontos geométricos que formam uma área trapezoidal da superfície da estrada (PLANO 1). As figuras da coluna da direita (B) ilustração os 4 últimos pontos geométricos que formam uma área retangular (PLANO 2), este pontos são utilizados para formar a matriz de transformação homográfica.	50
Figura 19:	Ilustração da área central isolada da imagem da pista (ROI-2), onde as faixas de sinalização estão contidas (retângulo esverdeado), e da divisão realizada nessa área em três regiões menores: Região (a), Região (b) e Região (c).	51
Figura 20:	Ilustração dos diferentes métodos de limiarização de imagens.	52
Figura 21:	Ilustração do procedimento de detecção de linhas da transformada de Hough.	54
Figura 22:	Ilustração do procedimento de estimação RANSAC de uma reta.	57
Figura 23:	Ilustração do funcionamento geral do algoritmo MSLC.	58
Figura 24:	Ilustração do efeito do filtro de Kalman em dois pontos que definem os segmentos de linha mais próximos do veículo de ambas faixas detectadas. As linhas verde (<i>line 1</i>) e roxa (<i>line 2</i>) corrigidas pelo filtro de Kalman. As linhas azul (<i>line 3</i>) e vermelha (<i>line 4</i>) são os pontos sem correção.	62
Figura 25:	Ilustração das áreas de buscas utilizadas pelo sistema para extrair os segmentos de linha que descrevem as faixas de sinalização.	63
Figura 26:	Ilustração do processo de extração dos segmentos de linha utilizando as áreas de busca obtidas através das previsões feitas pelo filtro de Kalman.	63
Figura 27:	Ilustração do processo de extração dos segmentos de linha utilizando as área de busca obtidas pelo MEL.	64
Figura 28:	Ilustração das regiões de busca fixa 3 e 4. A área de busca 3 está delimitada pelos retângulos na cor vermelha e área de busca 4 pelos retângulos na cor verde.	65
Figura 29:	Ilustração do procedimento de preenchimento de lacunas na detecção quando as linhas na pista estão na forma tracejada.	66
Figura 30:	Ilustração do procedimento de detecção de linhas falso-positivas devido a interferência de veículos e ruído na pista.	67
Figura 31:	Ilustração do procedimento de correção do comprimento das linhas detectas.	67
Figura 32:	Ilustração das doze etapas do sistema de detecção de faixas de sinalização desenvolvido neste trabalho.	68
Figura 33:	Ilustração do sistema implementado, suas componentes, ordenamento e interações.	72
Figura 34:	Ilustração das etapas de pré-processamento executados pelo sistema.	73
Figura 35:	Ilustração do procedimento de aquisição da imagem da pista utilizando as câmeras do veículo autônomo Alice.	74

Figura 36:	Ilustração do procedimento separação da imagem de interesse 1 (ROI-1). A imagem (1) representa a imagem da pista obtida pelas câmeras, a imagem (2) delimita a região de interesse 1 e a imagem (3) representa a região de interesse extraída.	74
Figura 37:	Ilustração das etapas de conversão da escala de cores e da aplicação do filtro de mediana.	75
Figura 38:	Ilustração das etapas de transformação de perspectiva e extração da segunda região de interesse.	75
Figura 39:	Ilustração da primeira possibilidade de sequência de execução do sistema.	77
Figura 40:	Ilustração dos resultados obtido através da execução do algoritmo de MSLC sobre as linhas encontradas pela transformada de Hough.	77
Figura 41:	Ilustração da segunda possibilidade de sequência de execução do sistema.	78
Figura 42:	Ilustração da terceira possibilidade de sequência de execução do sistema.	79
Figura 43:	Ilustração da quarta possibilidade de sequência de execução do sistema.	80
Figura 44:	Ilustração da quinta possibilidade de sequência de execução do sistema.	81
Figura 45:	Ilustração das etapas de Interpretação executadas pelo sistema.	82
Figura 46:	Ilustração da correção das lacunas entre os segmentos de linha, devido as faixas de sinalização tracejadas.	82
Figura 47:	Ilustração da saída do sistema de detecção de faixas de sinalização desenvolvido nesse trabalho.	83
Figura 48:	Ilustração dos limites usados para definir a região da pista em que as faixas de sinalização e borda da pista detectados são consideradas corretas.	87
Figura 49:	Ilustração dos resultados obtidos no cenário Cordova 1 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).	91
Figura 50:	Ilustração dos resultados obtidos no cenário Cordova 2 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).	92
Figura 51:	Ilustração dos resultados obtidos no cenário Washington 1 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).	93
Figura 52:	Ilustração dos resultados obtidos no cenário Washington 2 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).	94
Figura 53:	Ilustração dos resultados obtidos no cenário clip_i5s_0085 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).	95
Figura 54:	Ilustração dos resultados obtidos no cenário clip_i5s_0094 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).	96
Figura 55:	Ilustração dos resultados obtidos no cenário clip_i5s_0096 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).	97
Figura 56:	Ilustração dos resultados obtidos no cenário clip_i5s_0104 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).	98
Figura 57:	Ilustração de algumas detecções falso-positivas, obtidas em ambos os bancos de dados utilizados para validação neste trabalho.	99

LISTA DE TABELAS

Tabela 1:	REVISÃO BIBLIOGRÁFICA	39
Tabela 2:	BANCO DE DADOS - CALTECH	85
Tabela 3:	BANCO DE DADOS - UFRGS	86
Tabela 4:	RESULTADOS - LAPSI	88
Tabela 5:	RESULTADOS - CALTECH	88
Tabela 6:	COMPARAÇÃO - CALTECH Vs. LAPSI	89
Tabela 7:	RESULTADOS - UFRGS	89

LISTA DE ABREVIATURAS

2D	Bidimensional
3D	Tridimensional
ABS	<i>Anti-lock Breaking System</i>
ADAS	<i>Advanced Driver Assistance System</i>
A/D	Analógico/Digital
BIRD	<i>Boundary-based IPM for Road Detection</i>
CALTECH	California Institute of Technology
CCD	<i>Charge-Coupled device</i>
CMOS	<i>Complementary Metal Oxide Semiconductor</i>
CS	Conjunto Consenso
CAM	Conjunto de Amostras Mínimas
DARPA	Defense Advanced Research Projects Agency
DSP	<i>Digital Signal Processor</i>
FP	Falso Positivo
FPGA	<i>Field Programmable Gate Array</i>
GB	Giga Byte
GPS	<i>Global Positioning System</i>
IDE	<i>Integrated Development Environment</i>
IPM	<i>Inverse Perspective Mapping</i>
LAPSI	Laboratório de Processamento de Sinais e Imagens (UFRGS)
LDR	<i>Lane Detection and Reconstruction</i>
LIDAR	<i>Light Detection And Ranging</i>
MEL	Método de Extensão de Linha
MRU	Movimento Retilíneo Uniforme
MRUV	Movimento Retilíneo Uniformemente Variado
MSLC	Método de Seleção de Linha Candidata

OMS	Organização Mundial de Saúde
ONU	Organização das Nações Unidas
OPenCV	<i>Open Source Computer Vision</i>
RANSAC	<i>Random Sample Consensus</i>
RGB	<i>Red, Green and Blue</i>
ROI-1	Região de Interesse 1
ROI-2	Região de Interesse 2
LDW	<i>Lane Departure Warning</i>
LKS	<i>Lane Keeping System</i>
UFRGS	Universidade Federal do Rio Grande do Sul

LISTA DE SÍMBOLOS

α	Fator de escala da matriz Gaussiana
ρ	Eixo vertical do Espaço de Parâmetros da transformada de Hough
θ	Eixo horizontal do Espaço de Parâmetros da transformada de Hough
w_k	Ruído de processo
e_k	Ruído de medição
e_k	Fator de Amortecimento dos elementos da matriz Gaussiana
ε	Limiar da região <i>inliers</i>
\mathbf{p}_i	Vetor de pontos do sistema de coordenadas mundos 3D
\mathbf{p}'_i	Vetor de ponto do sistema de coordenadas mundos 2D
\mathbf{R}_k	Matriz de covariância do ruído medido
\mathbf{Q}_k	Matriz de covariância do ruído do processo
\mathbf{I}	Matriz Identidade
y	Distância da equação do MRUV
\dot{y}	Velocidade da equação do MRUV
a	Aceleração da equação do MRUV
\mathbf{H}	Matriz Homográfica
\mathbf{H}_k	Matriz de relação entre estado previsto e valor medido
$\mathbf{X}_{k k-1}$	Matriz de Estado futuro do filtro de Kalman
$\mathbf{X}_{k k}$	Matriz de Estado atual do filtro de Kalman
$\mathbf{X}_{k-1 k-1}$	Matriz de Estado passado do filtro de Kalman
$\mathbf{F}_{k k-1}$	Matriz de representação da relação entre o estado anterior, medido e o estado previsto do filtro de Kalman
$\mathbf{P}_{k k-1}$	Matriz de representação covariância do erro de estimativa a priori
$\mathbf{P}_{k k}$	Matriz de representação covariância do erro de estimativa a posterior
\mathbf{K}_k	Matriz de ganho de Kalman
\mathbf{y}_k	Vetor de entrada do filtro de Kalman

\mathbf{G}_i Matriz Gaussiana
 $T(x, y)$ Valor de limiar
 C Valor de correção do limiar

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Motivação	30
1.2	Objetivo	31
1.3	Organização do documento	32
2	MATERIAIS E MÉTODOS	33
2.1	Histórico	33
2.2	Revisão Bibliográfica	36
2.3	Estado da Arte	38
3	FUNDAMENTAÇÃO TEÓRICA	41
3.1	Representação Digital da Imagem	41
3.2	Aquisição das Imagens	43
3.3	Região de Interesse (ROI)	44
3.4	Filtro de Mediana	46
3.5	Transformação de Perspectiva	47
3.6	Filtro de Limiar Adaptativo	51
3.7	Transformada de Hough	53
3.8	Método de Seleção de Linha Candidata (MSLC)	55
3.9	Filtro de Kalman	58
3.10	Áreas de Busca	62
3.11	Correção de Erros	65
3.12	Metodologia	67
4	IMPLEMENTAÇÃO	71
4.1	Sistema	71
4.1.1	Inicialização	72
4.1.2	Pré-Processamento	73
4.1.3	Extração de características	76
4.1.4	Interpretação	81
5	EXPERIMENTOS E RESULTADOS	85
6	CONCLUSÕES	101
	REFERÊNCIAS	103

1 INTRODUÇÃO

O desenvolvimento tecnológico trouxe, indubitavelmente, melhores condições de vida para a humanidade. Motor das revoluções industriais, a tecnologia permitiu a substituição da força humana e animal pela máquina, aumentou a produtividade dos campos, melhorou a saúde com a higiene e os medicamentos, etc. No século passado, a eletricidade e a eletrônica foram responsáveis pela onda das comunicações, seguida, na segunda metade do século, pelo surgimento da sociedade da informação com a evolução do mundo digital.

A flexibilidade dos sistemas digitais permitiu a simbiose das atividades de instrumentação, controle e processamento de informações com praticamente todas as áreas do conhecimento ligadas à tecnologia: da medicina à agricultura, da geração de energia à produção industrial, do comércio às finanças, das artes às interações entre pessoas e grupos entre outras... Evidentemente que as funções trazidas pelo universo digital trouxeram comodidades e facilidades para as pessoas mas cobram um preço exigindo adaptação e treinamento e, muitas vezes, um esforço para o qual o ser humano não foi preparado ao longo da evolução, gerando estresse e até doenças. A administração do uso dos bens tecnológicos é, assim, um tema altamente relevante pelo impacto sobre a sociedade quanto a própria sustentabilidade do processo devido ao impacto no meio ambiente.

Este trabalho estuda um dos setores do mundo moderno: o do transporte e mais especificamente o meio de transporte individual. Não resta dúvida que o deslocamento de automóveis é um fenômeno marcante em todos os centros urbanos. Os automóveis são realmente máquinas espetaculares sob os mais diversos aspectos de conforto, eficiência, design, comodidade, segurança, etc. Entretanto, os motoristas imersos por longos períodos a um trânsito muitas vezes caótico, sofrem agressões físicas e psicológica que podem tirar-lhes a necessária atenção ou até desestabilizá-los, o que pode provocar manobras inadequadas com risco de acidentes, até fatais. Pode-se dizer que grande parte das mortes involuntárias (isto é, excluindo-se doenças, guerras, assassinatos e catástrofes) são devidas ao trânsito e, nesse, a maioria é causada por falha humana.

O automóvel é, atualmente, um sistema eletro-mecânico com dezenas ou centenas de sensores, atuadores e microprocessadores. Neste trabalho estudamos e desenvolvemos uma função para ajudar ao condutor no contexto do que é conhecido na literatura “Sistema Avançado de Auxílio ao Condutor (do inglês, *Advanced Driver Assistance System* (ADAS)). O foco do trabalho é a detecção em tempo real da sinalização horizontal para auxiliar o condutor a manter-se na faixa de rolamento. Aplicações destinadas a automóveis, que ajudam os motoristas a evitar colisões e melhorar a segurança na estrada vem sendo cada vez mais pleiteadas pela sociedade e por agências governamentais em resposta ao crescente número de acidentes no trânsito (STASZEWSKI; ESTL, 2013). Apesar da utilização da visão computacional para aplicações automobilísticas já ser estudada há anos, apenas recentemente ela vem sendo posta em prática e adentrando no mercado de

automóveis. Exemplo destes avanços já podem ser vistos no carro autônomo da Google. Desde o seu lançamento piloto no ano de 2010, o mesmo percorreu aproximadamente 225.000 km causando apenas um acidente (THEGUARDIAN, 2016).

Ao longo das duas últimas décadas os fabricantes de automóveis vêm investindo na pesquisa e desenvolvimento de novas tecnologias, que utilizam processamento de imagem para auxiliar o motorista em diversas tarefas, como estacionar o veículo de forma automática, monitorar a proximidade de outros veículos durante o deslocamento, evitar colisões e maximizar a segurança dos veículos na estrada. Muitas dessas pesquisas concentraram-se na extração das informações sobre o ambiente que cerca o veículo utilizando apenas câmeras de vídeo, como por exemplo: o Controle Ativo de Cruzeiro, Frenagem Automática de Emergência, Proteção Pré-Colisão e o Sistema de Permanência na Pista (KOWALENKO, 2016). O sistema *lane-assist da Audi* (2016) e da Volkswagen (2014), ilustrados nas Figuras 1 e 2, é uma ferramenta de monitoração em tempo real que detecta as faixas de sinalização da pista e gera alertas ao motorista quando o veículo começa a se mover para fora da região delimitada pelas faixas de sinalização e no caso do sistema da Ford (2016) corrige automaticamente a trajetória do veículo na pista.

Essa detecção pode ser realizada através de uma variedade de sensores, porém o método de aquisição de dados da pista mais comum é através de uma câmera instalada no painel frontal do veículo, na altura do retrovisor, de forma a garantir uma vista da pista robusta.

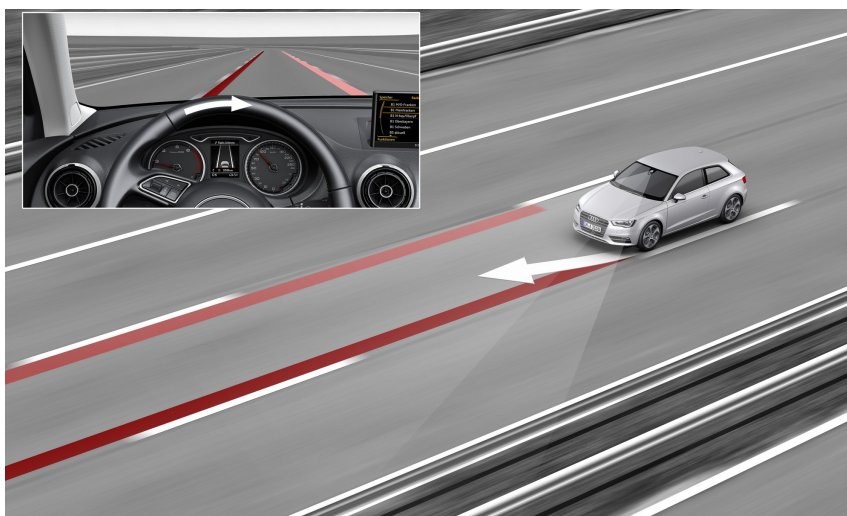


Figura 1: Ilustração do sistema de permanência na pista da Audi (AUDI, 2016).

Estes sistemas são projetados para minimizar algumas das principais causas de acidentes causados por erro humano: imprudência, distrações e sonolência. Eles podem ser classificados em dois tipos (HOWARD, 2013; VOUSDEN, 2014):

O Sistema LDW (*Lane Departure Warning*) que geram alertas ao condutor quando o veículo está deixando a região delimitada pelas faixas de sinalização. Esses alertas podem ser do tipo: sonoros, visuais ou vibração (VOUSDEN, 2014).

O Sistema LKS (*Lane Keeping System*) que além de gerar alertas ao condutor como no sistema LDW, podem tomar medidas automáticas para garantir a permanência do veículo na pista. A Mercedes utiliza esse tipo de controle automático, que é realizado através de travagem das rodas do carro, que criam uma força de tração que impele o veículo para dentro da região limitada pelas faixas de sinalização (VOUSDEN, 2014). Outro método é



Figura 2: Ilustração do assistente de detecção de faixas de sinalização da Volkswagen (CARICOS, 2012).

o utilizados pela Audi e a Volkswagens que conduz o veículo de volta para a pista através de sinais de controle aplicados sobre a direção hidráulica do veículo. Em ambos os casos, o motorista pode facilmente superar as intenções do carro girando o volante (VOUSDEN, 2014).

A Volkswagen (2016), Audi (2016) e a BMW (2016) são algumas das fabricantes que atualmente oferecem sistemas do tipo LDW de monitoramento do veículo sobre a pista. A Mercedes (2016) e Ford (2016) são alguns dos fabricantes que oferecem os sistema do tipo LKS.

O grande desafio de desenvolver sistemas desse tipo, assim como boa parte de todos os sistemas que processam imagens em tempo real, é o seu custo computacional. Muitas das técnicas de processamento de imagens necessária para realizar determinadas tarefas essenciais para detecção das faixas são bastante lentas e requerem processadores com uma alta capacidade de processamento, como processadores de DSP e FPGA. O emprego de processadores de alto desempenho tem um grande impacto no preço de qualquer sistema, muitas vezes resultando no descontinuação da pesquisa. Nessa ótica, o desenvolvimento de um sistema otimizado que seja capaz de realizar as tarefas de processamento de imagem com um baixo custo computacional pode ser fundamental para o barateamento desse sistema.

Outra grande dificuldade é a detecção das faixas de sinalização em ambientes urbanos. A maioria dos trabalhos sobre detecção de faixas de sinalização disponíveis na literatura aborda o tráfego em rodoviais, autoestradas e vias de alto fluxo de veículos, pois a detecção nesse tipo de pista é mais fácil, uma vez que elas estão sujeitas a uma menor incidência de oclusões e deterioração da sinalização em comparação a ambientes urbanos. A Figura 3 ilustra alguns dos problemas encontrados em ambiente urbano e que devem ser superados, dentre eles podemos enfatizar: pavimentos de cor irregular, muitas fontes de sombras, borda da pista (cordão da calçada) de cor irregular e regiões de forte reflexão de luz solar devido a cor do pavimento.

Nesse trabalho, foram investigadas e desenvolvidas técnicas para a otimização da extração das linhas que descrevem as faixas de sinalização na pista, para aplicações em tempo real com um foco especial em ambientes urbanos. A pesquisa realizada visou o desenvolvimento de estratégias para superar o problema do alto custo computacional devido as técnicas de processamento de imagens utilizadas para extração das faixas de sinalização de trânsito.



Figura 3: Ilustração de uma rua em uma área urbana de Pasadena (CA), que mostra alguns dos desafios que devem ser superados pelos sistema de detecção de faixa de sinalização (ALY, 2008).

1.1 Motivação

Na última década, o aumento percentual do número de veículos foi onze vezes maior que o da população. De 2001 a 2012, a frota brasileira passou de 24 milhões para 50 milhões de veículos (BARROS, 2015). Como consequência, a quantidade de acidentes e mortes no trânsito brasileiro aumentou em quantidades alarmantes. A OMS (Organização Mundial da Saúde) considerou as mortes em acidentes de trânsito como um problema mundial de saúde pública, alcançando a marca de nona maior causa de morte no mundo, totalizando 1,3 milhões de mortes por ano, o que motivou a ONU (Organização das Nações Unidas) lançar a década de ação pelo trânsito seguro¹ (OMS, 2011; WORLD HEALTH ORGANIZATION, 2015). Com o objetivo de desacelerar o aumento de mortes no trânsito, o governo vem incentivando a indústria de automóveis a desenvolverem sistemas de prevenção de acidentes. Pesquisas apontam que 90% dos acidentes nas estradas brasileiras acontecem devido à falha humana (ONSV, 2015). A combinação de imprudência dos motoristas, péssimas condições de vias, perda de controle do veículo, somados, a situações meteorológicas adversas são alguns dos principais fatores que contribuem para essa aumento (PINHONI, 2011).

Em consequência ao aumento das estatísticas de acidentes de trânsitos, muitas fabricantes de automoveis vem desenvolvendo novos sistemas de segurança para seus veículos (CALMON, 2015) com objetivo de garantir maior segurança em seus automoveis e como forma a conquistar diversificação no mercado automobilístico.

¹A Década de Ação pela Segurança no Trânsito 2011-2020 é uma campanha lançada em 11/05/2011 pela ONU na qual governos de todo o mundo se comprometem a tomar novas medidas para prevenir os acidentes no trânsito, que matam cerca de 1,3 milhão de pessoas por ano (ONSV, 2015).

As tecnologias de segurança convencionais se dividem em dois grupos: passiva e ativa. Os itens de segurança passiva, como cinto de segurança, *airbag* ou freios ABS, agem após a colisão e minimizam os danos ao motorista e passageiros. Já as tecnologias de segurança ativa evitam acidentes e diminuem as chances de eles acontecerem. Sistemas de monitoramento da concentração do motorista ao volante, controlador de velocidade e detector de distância dos veículos em todas as direções são alguns exemplos (ISTOÉDINHEIRO, 2013). Muitos desses sistemas utilizam-se de sensores de laser e sensores de radar para realizar a detecção. Embora esses tipos de sensores sejam mais preciso, eles são extremamente caros, custando em alguns casos até 10 vezes a mais do que o preço do veículo², além de gerar e sofrer interferência com sistemas semelhantes.

Os sistemas que utilizam câmeras de vídeos para extrair as informações do ambiente são uma alternativa robusta e de baixo custo, uma vez que sensores passivos não produzem ou sofrem interferência, porém, assim como na maioria dos sistemas de processamento de imagens, o custo computacional para realizar a detecção de faixas é bastante alto. Fator esse que contribui com a baixa procura por esse sistema, devido ao alto custo de sua aplicação. Um sistema com um baixo requisito computacional poderia tornar esse tipo de sistema mais viável a indústria e torná-la mais acessível a população de baixa renda e conseqüentemente sua popularização, dessa forma, proporcionando uma baixa significativa de acidentes de trânsito.

1.2 Objetivo

O objetivo deste trabalho é desenvolver um sistema de detecção das faixas de sinalização em tempo real capaz de operar em áreas urbanas e intermunicipais, utilizando técnicas de processamento de imagem tais como o filtro de limiar adaptativo, transformada de Hough e um Método de Seleção de Linhas (MSLC) para realizar a extração dos segmentos de linha que descrevem as faixas de sinalização na pista de uma forma otimizada.

A principal contribuição deste trabalho é a otimização realizada para minimizar as regiões onde o procedimento de extração das linhas da imagem é aplicado, ou seja, utilizar as previsões feitas por um filtro de Kalman e pelo Método de Extensão de Linha (MEL) para isolar regiões da imagem onde as faixas de sinalização estão contidas, com o objetivo de reduzir o custo computacional das técnicas de processamento de imagem utilizadas para extrair as faixas de sinalização. Essa redução do custo computacional é essencial para utilização de hardwares de baixa performance e reduzir custos.

Além do objetivo principal, o sistema implementado neste trabalho buscou atender alguns requisitos elementares para sistemas de detecção de faixas de sinalização robustos, esses requisitos são:

- A qualidade da detecção de pista não deve ser afetada por sombras, decorrentes de árvores, edifícios, etc.
- Dever ser capaz de operar em ambientes com diferentes níveis de luminosidade recorrentes de condições atmosféricas diversas e devido a fontes de iluminação artificiais.

²O sensor LIDAR utilizado pelo veículo autônomo da Google para mapear o ambiente tridimensionalmente, chegam a custar \$80.000 (MOTAVALLI, 2015)

- Deve lidar com as estradas retas e curvas ao invés de assumir que as estradas são sempre retas.
- Deve identificar individualmente as faixas de sinalização para detectar os dois lados demarcatórios da pista para eliminar erros provenientes de ruídos nas imagens.
- Deve demandar um custo computacional baixo, para que o sistema seja capaz trabalhar em tempo real.

1.3 Organização do documento

O capítulo dois apresenta um breve histórico do sistema de detecção de faixas de sinalização desenvolvidos ao longo da última década pelas principais montadoras de automóveis. Em seguida, a revisão bibliográfica é apresentada, tentando contemplar as principais técnicas de detecção disponíveis na literatura. Por último, o estado da arte, onde são evidenciados os avanços mais recentes em sistemas de detecção de faixas de sinalização é apresentado.

No capítulo três a fundamentação teórica é apresentada, detalhando as técnicas de processamento de imagem, os métodos e os algoritmos utilizados neste trabalho, assim como a forma de interação entre eles. A estrutura de funcionamento do sistema implementado, suas componentes e suas sequências de execução são descritas no capítulo quatro. No capítulo cinco os resultados dos experimentos são apresentados e avaliados. Por fim, no capítulo seis são apresentadas as conclusões do trabalho desenvolvido.

2 MATERIAIS E MÉTODOS

Neste capítulo serão apresentados: um breve histórico dos sistemas de detecção das faixas sinalização, a revisão bibliográfica e o estado da arte.

2.1 Histórico

Embora, o primeiro sistema comercial de detecção de faixas de sinalização tenha sido lançado pela Mitsubishi Motors em 1992, sua popularização entre as montadoras ocorreu somente depois de 2003, quando alguns modelos de veículos com assistente de monitoração do veículo sobre a pista ou ADAS (*Advanced driver Assistance Systems*) foram lançados em modelos de luxo no Japão pela Honda (2007) e Nissan (2006). Estes sistemas alertavam o motorista se o veículo estava saindo fora da sua faixa de tráfego, e no caso do sistema da Honda, exercia um controle automático sobre o veículo para mantê-lo dentro da área destinada ao veículo.

Não demorou muito tempo para as outras grandes montadoras entrarem na briga e comecem a desenvolver seus próprios sistemas para adentrarem na era dos veículos inteligentes. O estopim da produção em larga escala de veículos inteligentes com tecnologias de assistência ao motorista, como sistema de piloto automático (veículo autônomo), foi o desafio lançado pela DARPA (*Defense Advanced Research Projects Agency*) em 2004, criado com o objetivo de impulsionar a pesquisa de tecnologias de veículos autônomos. Os veículos precisavam completar um trajeto de 242 km de forma totalmente autônoma. Na primeira edição do desafio nenhum dos concorrentes foi capaz de completar o desafio. No ano seguinte, cinco equipes foram capazes de concluir o trajeto. A equipe da universidade de Stanford, chegou em primeiro lugar e foi premiada com 2 milhões de dólares (BIBULI; CACCIA; LAPIERRE, 2007).

A Lexus (2006) lançou o seu sistema *multi-mode Lane Keeping Assist* no seu veículo LS460 em 2006, que utilizava câmeras e processadores de reconhecimento de padrões mais sofisticados. Este sistema era capaz de emitir um alerta audiovisual e controlar a direção do veículo para mantê-lo na pista.

Na Europa, o primeiro ADAS foi lançado em 2000, desenvolvido pela United States Iteris para os caminhões comerciais da Mercedes. O mesmo sistema foi adotado pela montadora de caminhões norte-americana Freightliner Trucks em 2002. Em ambos os sistemas, o motorista é alertado da saída não intencional da pista por um sinal de alerta sonoro (TJ, 2008). A Citroën foi a primeira montadora a oferecer sistemas de monitoração do veículo sobre a pista, para veículos de passeio na Europa em seus modelos C4, C5 e C6, lançados em 2005. Este sistema utilizava sensores de infravermelho para localizar as faixas de sinalização sobre a superfície da estrada, e através de um mecanismo de vibração instalado no assento, alertava o motorista se o veículo saísse da região de tráfego

destinada a ele (KOMPETENZZENTRUM, 2013).

Em 2007, a Audi e a BMW lançaram os seus sistemas *Lane Assist* (2007) e *Lane Departure Warning* (MASSY, 2007). Ambos os sistemas operavam utilizando uma câmera instalada na parte frontal do veículo e emitiam alertas sonoros no painel de instrumentação ou na forma de vibrações no volante, quando o veículo estava deixando a zona de tráfego limitada pelas faixas de sinalização presentes na pista (MASSY, 2007).

A General Motors (2016) entrou na briga ao lançar o seu sistema *Lane Departure Warning* no Cadillac STS, que utilizava câmeras de alta resolução para detectar de forma muito mais precisa as faixas de sinalização. O sistema da General Motors Produzia alertas sonoros e luzes de alerta no painel. No mesmo ano a Fiat lançou o seu *Lane Keep Assist*. Este sistema combinavam uma câmera que detectava as faixas de sinalização com um sistema de direção assistida da TRW (também conhecido como *Haptic Lane Feedback system*). Neste sistema, quando uma saída não intencional da região delimitada pelas faixas de sinalização for detectada, o sistema aplica um torque suave na direção automaticamente que ajuda a orientar o motorista de volta para o centro da pista (TRW, 2016; SYSTEM, 2016).

Atualmente o sistema de detecção faixas de sinalização se tornou uma das componentes principais do sistema de controle de veículo autônomo. Diversas empresas no vale do Silício vêm trabalhar em carros autônomos, suas abordagens adotadas variam muito, e algumas divergem no foco de aplicação. A Tesla, adotou um dos caminhos mais divergentes ao começar a vender carros com funções básicas de autodireção, que em sua essência não deve ser considerado um veículo autônomo como o da Google (MIGLIACCI, 2016). O Model S da Tesla Motors, combina as informações obtidas através de GPS, sensores de ultrassom e de uma câmera para detectar o caminho que o veículo deve seguir. Esse sistema ainda conta com um banco de dados compartilhado entre todos veículos Model S, o qual é utilizado para que o veículo aprenda quais decisões tomar em situações atípicas baseada no aprendizado obtido pelos outros veículos (JEN, 2015). A Figura 4 ilustra o sistema *Autopilot* da Tesla.

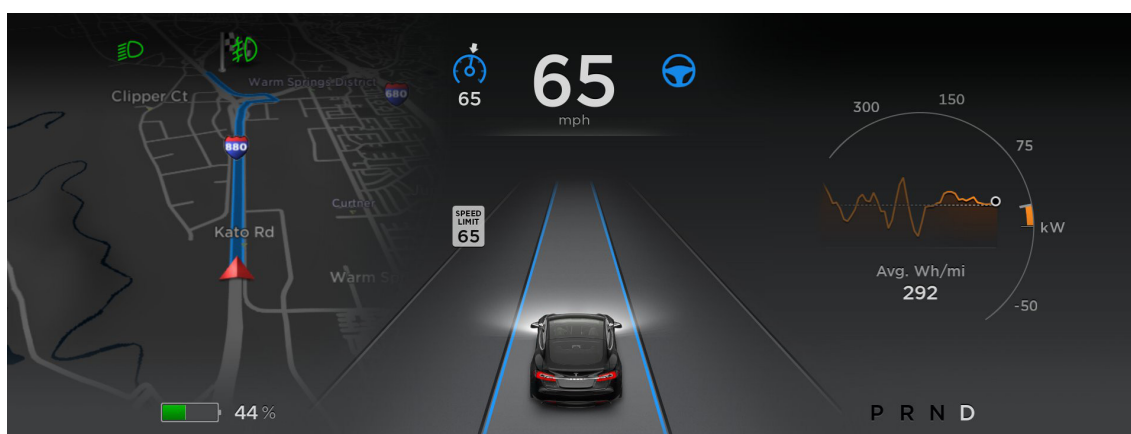


Figura 4: Ilustração da visão do sistema Autopilot da Tesla (JEN, 2015).

Os esforços iniciais do Google, em 2010, tinham por foco carros que se guiam de forma autônoma, mas sempre contando com uma pessoa por trás do volante para assumir o controle ao primeiro sinal de problemas. O sistema desenvolvido conta com quatro radares e uma câmera para realizar a detecção da região de tráfego. Essa fusão de sensores é fundamental para obter uma detecção dinâmica e precisa dos limites da pista, dos outros

veículos e da sinalização de trânsito que o veículo deve obedecer (GUIZZO, 2011). A Figura 5 ilustra a visão do sistema autônomo da Google.

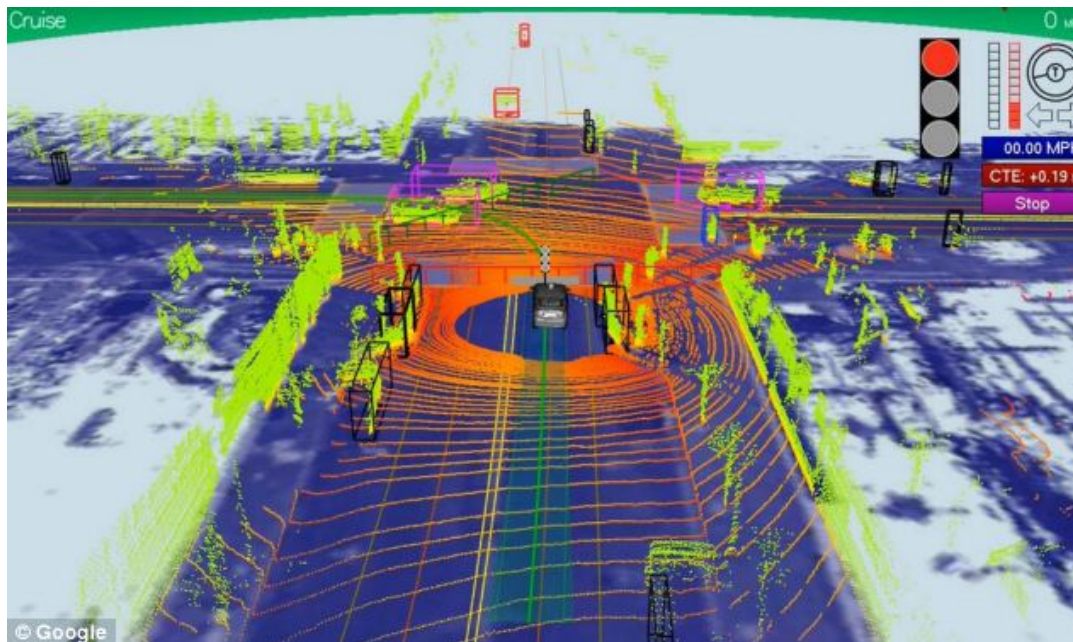


Figura 5: Ilustração da visão do sistema autônomo da Google (WOOLLASTON, 2013).

Ao contrário da Tesla, o Google optou por continuar em modo experimental, como consequência de um experimento realizado pela empresa em 2013, onde permitiu que alguns de seus funcionários utilizassem carros autoguiados em seus percursos diários de casa para o trabalho. Foram utilizadas câmeras remotas instaladas nos carros para monitorar o comportamento dos funcionários. O experimento mostrou que talvez não fosse possível para um motorista humano retomar a direção instantaneamente a tempo de evitar algum acidente. Assim os engenheiros do Google tomaram outro caminho, eliminando o motorista humano do circuito completamente, direcionando o foco principal do sistema para a segurança (MIGLIACCI, 2016).

Nessa ótica, fica evidente que a quantidade de tecnologia necessárias para construir um veículo inteligente ou totalmente sem motorista é surpreendente. Os veículos necessitam de diversos tipos de sensores de mapeamento tridimensional, sensores ópticos, sonares, radares; *Hubs* para centralizar milhões de pontos de dados; Microprocessadores *in-car* e conectividade de alta velocidade para capturar, processar e transportar dados.

Com isso em mente, a Intel lançou a iniciativa *The Internet of Things is on the Road to Autonomous Driving* com objetivo de introduzir plataforma de desenvolvimento para próxima geração de veículos inteligentes até 2021. Depois de ter colaborado com Ford, Hyundai e Kia Motors, a Intel está na vanguarda do desenvolvimento de ecossistemas voltados para automóveis. A plataforma da Intel conta com computação central no veículo, controle dinâmico dos movimentos do veículo, um *cluster* de sensores, análise de dados através do aprendizado de máquina, somado ao acesso a informações na nuvem, como mapas e fluxo de tráfego para criar novas funções e uma rede 5G para lidar com a troca de dados entre o carro e a nuvem (SULEMAN, 2016). A Figura 6 ilustra de forma geral as funcionalidades prevista na plataforma lançada pela Intel.



Figura 6: Ilustração da funcionalidades prevista na plataforma para veículos inteligentes lançados pela Intel (INTEL, 2016).

2.2 Revisão Bibliográfica

Pesquisa sobre veículos inteligentes visando melhorar a segurança do tráfego remontam à década de 1990 (TSUGAWA, 2005). Desde então, ADASs (*Advanced Driver Assistance Systems*) têm ganhado cada vez mais relevância como uma das componentes fundamentais dessa linha de pesquisa. Os sistemas do tipo LDR (*Lane Detection and Reconstruction*) extraí as faixas demarcatórias da pista a partir de imagens e estima na região de segurança adequada, ao mesmo tempo que apresentam um alto desempenho em tempo real e uma taxa de detecção satisfatória. Para tanto, diversos algoritmos e técnicas de processamento de imagens são utilizados no processo de extração das linhas que compõem a faixa de segurança.

Basicamente, existem dois tipos de abordagens utilizadas na detecção de faixas de sinalização. A primeira e mais comum utiliza técnicas de extração de características da pista e a partir delas determinam diretamente a posição das linhas e curvas que descrevem a pista. Os autores Wadhwa (2015), Saudi (2008) e Aung (2014) utilizam a transfor-

mada de Hough para extrair retas em imagens limiarizadas que descrevem as faixas de sinalização a frente do veículos resultantes de uma etapa de pré-processamento utilizado para destacar os contornos da pista. Já Song (2014) propõem um método que utiliza as linhas obtidas através da transformada de Hough para detectar a curvatura da pista. A transformada de Hough é dos métodos de detecção de linha mais utilizados em sistemas de detecção de faixas de sinalização. Embora, essa transformada seja extremamente robusta, é um processo que requer um alto nível de processamento, aumentando o custo das componentes do *hardware* do sistema no qual é utilizada.

Por isso diversos autores exploram alternativas à transformada de Hough, como a apresentada por Borkar (2010), que aplica uma correlação cruzada utilizando templates (forma padrão de faixa de sinalização) de faixas de sinalização pré-definidos, para encontrar as faixas de sinalização na pista. Embora, esse método apresente uma taxa de detecção baixa, devido a dificuldade de definir templates que se encaixam a todas as variações de faixas, outra alternativa é apresentada em Zhang (2013), que aplica uma mudança de perspectiva na imagem binarizada e utiliza o algoritmo RANSAC (*RANdom SAmple Consensus*)¹, para estimar a posição de faixas de sinalização da pista e seguida de uma técnica de *Probability Hypothesis Density* para extração das faixas de sinalização.

Por sua vez, uma alternativa a transformada de Hough mais eficiente foi proposta por Kim (2008), que propõe um algoritmo de detecção de pista capaz de lidar com ambientes e situações complexas, tais como faixas de sinalização desbotadas e curvaturas da pista. Nesse sistema são utilizadas redes neurais artificiais sobre as imagens para a detecção das faixas de sinalização. Os pixels das faixas de sinalização encontrados são agrupados usando splines cúbicas. As hipóteses são geradas a partir de conjunto aleatório de segmentos de linha. O algoritmo RANSAC é utilizado para validar as hipóteses encontradas, seguido de um filtro de partículas para realizar o rastreamento. A utilização de redes neurais oferece uma forma de classificação e identificação das faixas de sinalização extremamente eficiente, porém é limitada pela quantidade de variações de cenários utilizados para realizar o treinamento da rede.

A segunda abordagem utiliza modelos predefinidos da pista. Os autores Lopez (2005) e Guo (2005) utilizam métodos de detecção baseados em modelos da pista para detectar faixas de sinalização. Em ambos os métodos, é definido um modelo da pista, onde a posição de cada ponto de controle desse modelo é predefinida no sistema de coordenadas cartesianas que englobam a região da pista. Esses pontos de controle se ajustam no sistema de coordenadas de acordo com o movimento do veículo. Para estimar os parâmetros do modelo da pista é utilizado um algoritmo de RANSAC.

Outra metodologia relevante é aplicada por Xu (2009), Wang (2014) e Deng (2013), que utilizam um modelo B-Spline da pista que descreve o efeito de perspectiva de linhas paralelas construído a partir de faixas ou demarcações. Esse modelo é capaz de descrever uma ampla gama de estruturas de pistas onde outros modelos como em linha reta e modelos parabólicos não conseguem ao mesmo tempo que é robusto contra sombras e ruídos.

O método apresentado por Zhou (2010) predefine um conjunto de modelos de curvatura da estrada e seleciona o modelo que mais se aproxima das características extraídas da estrada. Embora, possa superar a maioria dos problemas encontrados na detecção de faixas, como imprecisões na detecção das faixas, causados por sombra ou objetos diversos, é extremamente difícil de combinar esses modelos. Sistemas que utilizam técnica de extração

¹RANSAC é um robusto estimador de parâmetros de modelos matemáticos, proposto por (FISCHLER; BOLLES, 1981).

de faixas de sinalização em imagens na escala de cinza são utilizadas por Huang (2015), Paetzold (2000) e Kluge (1995) para a extração das faixas de sinalização. No entanto, não apresentam um bom desempenho em diferentes condições de luz ou oclusões da pista.

O filtro de Bayes é utilizado por Coulombe (2002), Takahashi (1996) e Deusch (2012), para estimar as faixas de sinalização. Este método assume que a pista tem largura constante em uma superfície plana e apresentou bons resultados em cenários não muito complexos. Porém, não obtendo bons resultados em cenários com muitas curvas e variações de relevo. Ainda podemos citar o trabalho de Heij (2006), onde é feito um fracionamento vertical da imagem e cada fração é analisada em busca das linhas que descrevem as faixas de sinalização da pista sem realizar nenhuma transformação de perspectiva para isso.

Um sistema que utiliza um modelo de borda linear-parabólico é proposto por Jung (2004), combinando a robustez do modelo linear com a flexibilidade de modelo parabólico, capaz de detectar curvas com bastante precisão ao mesmo tempo que delimita regiões de busca das faixas de sinalização utilizando como referência as posições das linhas encontradas nas imagens da pista anteriores. O sistema apresentou bons resultados de detecção na presença de ruído, sombras e diferentes condições de iluminação.

Uma solução um pouco mais aprimorada é apresentada por Paula (2013), que além de realizar a detecção em tempo real o sistema consegue classificar os diferentes tipos de faixas de sinalização. O método proposto utiliza um modelo estatístico simples para representar píxeis relacionados à pista e extrair as faixas. Um conjunto de características é calculado com base nas faixas detectadas ao decorrer do deslocamento do veículo sobre a pista, e um classificador em cascata é usado para reconhecer até cinco tipos de faixas de sinalização.

2.3 Estado da Arte

Em trabalhos mais recentes podemos citar o trabalho de Huang (2015), que apresenta um método de detecção de pista com extensão de base que procura blocos de contorno a partir de uma posição inicial pré-determinada, de baixo para cima e ao longo da pista, utilizando imagens na escala de cinza. Também é utilizado um método de predição, utilizando informações da posição do bloco atual, para estipular uma região de busca do próximo bloco de contorno, dessa forma haverá uma redução da área da imagem onde será realizada a busca pelo próximo bloco.

Sistemas de detecção de faixas de sinalização que utilizam somente informações de contornos da imagem da pista, perdem muita informação devido a diversas condições ambientais em que a pista está sujeita. Uma alternativa mais eficiente é apresentada por Gu (2015), que utiliza a transformada de Hough sobre os contornos extraídos da imagem da pista, para extrair a região da imagem onde provavelmente as faixas vão estar contidas. Essa região é definida no das linhas encontradas pela transformada. Em seguida, os autores propõem uma operação que realça as faixas de sinalização chamada *extremal-region enhancement*, que consiste na aplicação de um *kernel* composto por quatro operadores Sobel, com diferentes orientações e informações de largura das faixas de sinalização. Essa operação é aplicada sobre a região definida no das linhas encontradas pela transformada de Hough e realça a intensidade das linhas brancas, tornando a detecção das faixas mais robusta às sombras e variações de iluminação. Um rastreamento baseado em probabilidade é realizado sobre as linhas encontradas para decidir se a linha detectada é provavelmente uma faixa de sinalização. Esse rastreamento baseado em probabilidade utiliza as informações de largura de faixa, inclinação e número de píxeis contidos na faixa que foram

detectados nas imagens da pista anteriores para determinar a probabilidade da linha encontrada ser uma faixa de sinalização. Embora, esse método tenha se mostrado bastante eficiente na detecção de faixas de sinalização, ele não detecta a curvatura da pista e não demonstra ser um método robusto para ambientes urbanos.

Um método eficiente de detecção baseado em imagem espaço temporais (*spatiotemporal*) é proposto por Jung (2016), onde o sistema proposto cria uma imagem com uma perspectiva do tipo *Bird's Eye*, obtido através de um acumulo de um conjunto de trechos da região da pista, amostrados de uma localização constante de uma sequência de imagens da pista. As informações obtidas a partir dessas imagens são utilizadas para fazer uma estimação dos pontos que descrevem as faixas de sinalização. Esses pontos são obtidos utilizando a transformada de Hough em cada segmento da imagem que compõem espaço temporal. São aplicados ainda métodos de correção da posição dos pontos obtidos na análise atual, utilizando as informações dos pontos obtidos nas análises anteriores. Os pontos obtidos na análise atual são utilizados para alimentar um modelo de curva cúbica para descrever as linhas na pista.

O mapeamento inverso de perspectiva ou IPM (*Inverse Perspective Mapping*) é uma das técnicas mais úteis na detecção e análise da estrutura da pista. Uma variação do método do mapeamento inverso de perspectiva é apresentada por Zhenqiang (2016), que propôs um sistema de detecção das bordas da pista utilizando uma transformação geométrica baseada na informação de contornos da imagem, sem precisar fazer o procedimento de calibração da câmera, eliminando alguns problemas do método em relação a parâmetros fixos. Esse método é chamado de BIRD (*Boundary-based IPM for Road Detection*). Embora, muito robusto ao lidar com cenários muito ruidosos, devido a sombras e variação de luminosidade, este método não detecta com precisão a curvatura da pista e a detecção sofre muita interferência de veículos, que trafegam sobre as faixas de sinalização e bordas da pista.

Tabela 1: REVISÃO BIBLIOGRÁFICA

Nº	Autor	Título	Pré-processamento	Deteção	Rastreamento	Comentários
1	Alexandru Gurghian, Tejaswi Koduri, Smita V. Bailur, Kyle J. Carey, Vidya N. Murali (2016)	DeepLanes: End-To-End Lane Position Estimation using Deep Neural Networks	-	Redes Neurais Artificiais	-	A posição das linhas neste trabalho é estimada diretamente usando uma rede neural que utiliza imagens da vista lateral do veículo e não necessita de pré-processamento ou pós-processamento.
2	Yadi Li, Liguang Chen, Haibo Huang, Xiangpeng Li, Wenkui Xu, Liang Zheng, Jiaqi Huang (2016)	Nighttime lane markings recognition based on Canny detection and Hough transform	Filtro de suavização e borda	Transformada de Hough	-	Neste trabalho os autores apresentam um sistema robusto, destinado para cenários noturnos, porém necessita em média de 165.45 ms para processar uma imagem da pista.
3	Hao Yu, Yule Yuan, Yueting Guo, Yong Zhao (2016)	Vision based Lane Marking Detection and Moving Vehicle Detection	Filtro de Sobel, de Shen Jun e de limiar adaptativo	Transformada de Hough	-	Nesse trabalho é proposto um algoritmo que utiliza filtro de Sobel e Shen Jun de forma limitada para detectar as faixas de sinalização.
4	Toan Minh Hoang, Hyung G.Hong, Husan Vokhidov, Kang R. Park (2016)	Road Lane Detection by Discriminating Dashed and Solid Road Lanes Using a Visible Light Camera Sensor	-	Detector de Segmento de Linha (LSD)	-	Nesse trabalho foi proposto um algoritmo de detecção das faixas de sinalização capaz de distinguir e a linha detectada é do tipo tracejada ou contínua. Embora, o algoritmo não necessita de etapas de pré-processamento para extrair as linhas na imagem da pista ele apresenta algumas limitações em cenários com muitas sombras e oclusões.

5	Rodrigo Berriel, Edilson de Aguiar, Vanderlei V. S. Filho, Thiago O. Santos (2015)	A Particle Filter-based Lane Marker Tracking Approach using a Cubic Spline Model	Filtro Gaussiano, Filtro de limiar adaptativo, Transformação Inversa de Perspectiva (IPM)	Filtro de Partículas	Filtro de Partículas	Neste trabalho, os autores combinam transformação de perspectiva, filtro de partículas e um modelo de spline cúbica para estimar a posição as faixas de sinalização na imagem da pista.
6	Youngwan Lee, Hakil Kim (2016)	Real-time lane detection and departure warning system on embedded platform	Filtro convolutivo, Filtro de limiar, Transformação Inversa de Perspectiva (IPM)	RANSAC	Filtro de Kalman	Através da combinação de transformação de perspectiva, RANSAC e filtro de Kalman, o algoritmo proposto apresentou bons resultados na detecção de faixas de sinalização mesmo em ambientes urbanos complexos
7	Seongrae Kim, Junhee Lee, Youngmin Kim (2016)	Speed-Adaptive Ratio-Based Lane Detection Algorithm for Self-Driving Vehicles	Filtro de Sobel e detector de borda Canny	Transformada de Hough	-	Os autores propuseram um método para estimar a posição do veículo nas imagens futuras para diminuir o erro na detecção das faixas de sinalização.
8	Zuo-quan LI, Hui-min MA, Zheng-yu LI (2016)	Road Lane Detection With Gabor filters	Filtro de Gabor e filtro de limiar	Transformada de Hough	-	O sistema apresentado é bastante robusto em diferentes condições de iluminação. Ele detecta o ponto de fulga da imagem e a partir dele, restringe áreas de busca das faixas de sinalização. As faixas são extraídas utilizando um filtro de Gabor para detectar bordas em múltiplos ângulos e a transformada de Hough.
9	Seokha Hwang, Youngjoo Lee (2016)	FPGA-based Real-time Lane Detection for Advanced Driver Assistance Systems	Filtro de limiar, Gaussiano, Sobel e detector de borda de Hysteresi	Transformada de Hough	-	Nesse trabalho é proposto uma arquitetura de FPGA para um sistema de detecção de faixas de sinalização em tempo real para otimizar o custo computacional do sistema. Os autores conseguiram aumentar a taxa de processamento em 41%.
10	Tin T. Duong, Cuong C. Pham, Tai H. Tran, Tien P. Nguyen, Jae W. Jeon (2016)	Near Real-Time Ego-Lane Detection in Highway and Urban Streets	Transformação Inversa de Perspectiva (IPM), detector de borda, filtro de limiar e morfológico	Transformada de Hough	-	Os autores propuseram uma sistema de detecção de faixas de sinalização quase em tempo real destinado para rodovias e áreas urbanas. O sistema proposto nesse trabalho utiliza janelas deslizantes, que percorrem toda a imagem analisando as bordas nelas contidas, para isolar as áreas onde as faixas estão contidas.
11	Rui Fan, Victor Prokhorov, Naim Dahnoun (2016)	Faster-Than-Real-Time Linear Lane Detection Implementation Using SoC DSP TMS320C6678	Filtro Gaussiano e filtro de Sobel	Transformada de Hough	-	Nesse trabalho foi desenvolvido um sistema de detecção de faixas de sinalização utilizando o Open Multi-Processing (OpenMP) para utilizar todos os cores do SOC do chip de Processamento Digital de Sinais TMS320C6678. Os resultados demonstram um aumento de 76,7% na distribuição adequada de carga utilizada. Os resultados experimentais mostraram atingiu um alto desempenho de 81 FPS em imagens de resolução 1242x375.

A Tabela 1 contempla uma revisão de alguns dos últimos trabalhos sobre detecção de faixas de sinalização disponíveis na literatura, e enfatiza que a transformada de Hough é o método de extração de linhas mais utilizado atualmente em sistema de detecção de faixas de sinalização, razão essa pela qual ela foi utilizado neste trabalho. Também mostra que poucos trabalhos recentes utilizam técnicas de correção e rastreamento como o filtro de Kalman e mudanças de perspectiva (IPM) da vista da pista, o que motivou a utilização destas técnicas neste trabalho, e principalmente, destaca que poucos trabalhos propõem estratégias para a redução do custo computacional associado as técnicas de processamento de imagem utilizadas.

3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão revisados alguns conceitos fundamentais para o entendimento do sistema desenvolvido nesse trabalho. Inicialmente será dada uma visão global dos fundamentos de técnicas de processamento de imagem voltadas ao tratamento, transformação de imagens, técnicas de identificação de linhas e rastreamento de pontos que são utilizadas pelo sistema. Por último, será apresentado o funcionamento do sistema implementado neste trabalho.

3.1 Representação Digital da Imagem

Uma imagem pode ser definida como uma função bidimensional de intensidade luminosa $f(x,y)$, onde x e y são as coordenadas espaciais de linha e coluna, e o valor de f representa a informação de cor ou escala de cinza da imagem (monocromática). Dessa forma uma imagem digital pode ser considerada uma matriz cujos índices das linhas e colunas identificam um ponto dentro da imagem e o correspondente valor do elemento. O termo imagem digital refere-se a uma imagem que pode ser discretizada quanto as suas coordenadas espaciais e quanto a intensidade de seu brilho. Os elementos desta matriz digital são chamados de elementos da figura ou de píxeis (*picture elements*) (GONZALEZ; WOODS, 2006). A Figura 7 ilustra o conceito de pixel e mostra a convenção espacial apresentada por Pedrini (2007) e utilizada neste trabalho, onde a origem da imagem está localizada no canto superior esquerdo da mesma.

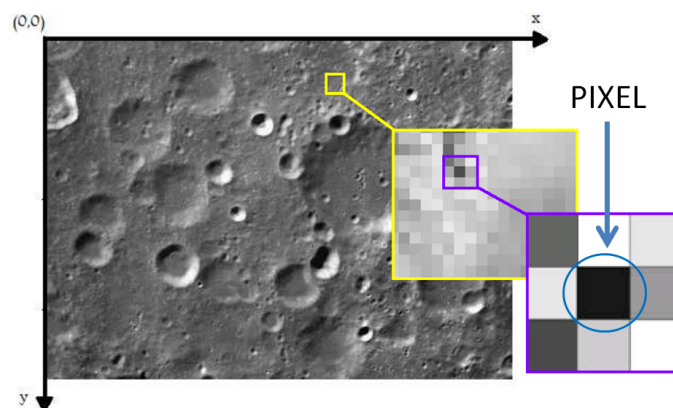


Figura 7: Ilustração do conceito de pixel e da convenção do sistema de coordenadas para representação de imagens digitais (PEDRINI; SCHWARTZ, 2007).

Como os computadores não manipulam dados analógicos, é necessário converter uma imagem contínua em sua forma digital, ou seja, a função $f(x,y)$ precisa ser convertida para

uma descrição discreta, assim a imagem digital é obtida por um processo de digitalização, envolvendo dois passos, a amostragem e a quantização.

A amostragem consiste em discretizar o domínio de definição da imagem nas direções dos eixos y e x da imagem, gerando uma matriz de $M \times N$ amostras, onde M representa o eixo vertical e N o eixo horizontal da imagem ilustrada na Figura 7.

A quantização incide em escolher o número inteiro L de níveis de cinza (no caso de imagens monocromáticas) permitidos para cada ponto da imagem. A intensidade de iluminação do pixel pode variar entre os valores 0 a L , e, uma convenção comum, é atribuir o valor zero a cor mais escura da escala, onde zero representa a máxima ausência de iluminação e 255 o valor máximo de brilho (PEDRINI; SCHWARTZ, 2007).

Segundo Pedrini (2007), uma imagem colorida é uma imagem multibanda ou multiespectral, onde a cor em cada ponto (x,y) é definida por meio de três grandezas: luminância, matiz e saturação. A luminância está associada com o brilho da luz, o matiz com o comprimento de onda dominante e a saturação com o grau de pureza da matiz. De acordo com Castleman (1996), para algumas aplicações pode ser mais conveniente tratar a imagem colorida como uma função bidimensional que possui duas coordenadas espaciais e uma de cor onde a informação de cor normalmente é fornecida a partir de três componentes com significado que varia de acordo com o modelo utilizado para representação de cores. A maioria das cores visíveis ao olho humano pode ser representada por uma combinação de bandas das cores primárias, vermelha (R , *Red*), verde (G , *Green*) e azul (B , *Blue*). Uma representação comum para imagem colorida utiliza três bandas R , G e B com profundidade de 1 byte por pixel, para cada banda, ou seja, profundidade de 24 bits por pixel”.

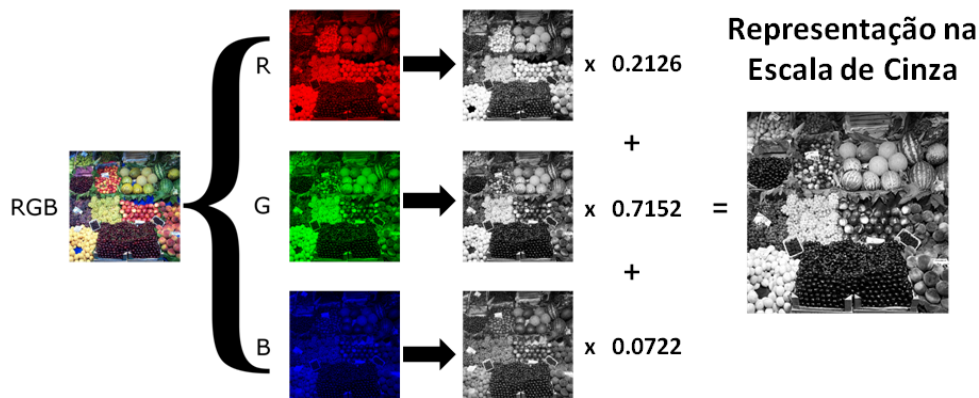


Figura 8: Ilustração do procedimento de conversão dos canais RGB de uma imagem colorida para uma imagem em escala de cinza. Adaptado de (DILMEN, 2012).

Em algumas aplicações a informação de cor pode não ser necessária para realizar determinadas tarefas, como extrair informações de contorno e estruturas na imagem. Neste caso, utilizamos a imagem na escala de cinza, que é uma representação de uma imagem colorida em uma única componente, regida por uma função bidimensional que possui duas coordenadas espaciais e uma de intensidade. Essa escala usualmente é definida com valores de 0 até 255, e representa 255 tons de cinza.

Neste trabalho também serão utilizadas imagens limiarizadas ou imagens binárias que são imagens onde os píxeis assumem os valores 0 (preto) ou 1 (branco), assim necessitam de apenas 1 bit por pixel para serem representados. As imagens binárias ou binarizadas são normalmente resultado de um processo chamado "segmentação" e são utilizadas para

destacar os objetos de interesse. A conversão para escala de cinza de uma imagem no formato RGB pode ser obtida através de várias equações, a Equação (1) descreve uma delas, descrita por Hoffmann (2001).

$$y = 0.2126R + 0.7152G + 0.0722B \quad (1)$$

Onde R é o componente vermelho da imagem, G é a componente verde, B é a componente azul e Y é a intensidade na escala de cinza. Todos os recursos são normalizados na faixa de 0 a 255. A Figura 8 ilustra o procedimento de transformação de uma imagem colorida para a escala de cinza.

3.2 Aquisição das Imagens

A etapa de aquisição, envolve a captura da imagem por meio de um dispositivo ou sensor, como por exemplo, uma câmera de vídeo, e a conversão em uma representação adequada para o processamento digital subsequente. Estes dispositivos devem ser escolhidos levando em consideração o tipo de sensor, resolução e o número de níveis de cinza ou cores da imagem digitalizada, taxa de amostragem e codificação. A posição desse dispositivo também deve ser escolhida com cuidado, quando se quer obter uma vista vantajosa para a detecção de sinalização horizontal utilizando uma câmera de vídeo. As posições mais proveitosas são na parte superior-frontal do veículo e na parte frontal-interna do veículo, perto da posição do retrovisor do veículo, como ilustrado nas Figuras 9 e 10.

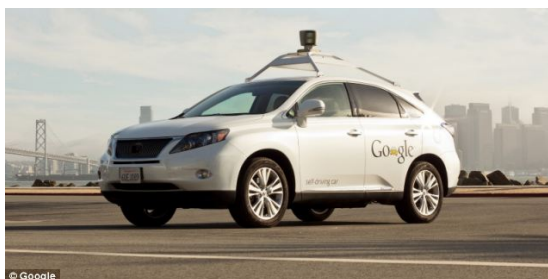


Figura 9: Ilustração da posição da câmera na parte externa do veículo (WOOLLASTON, 2013).

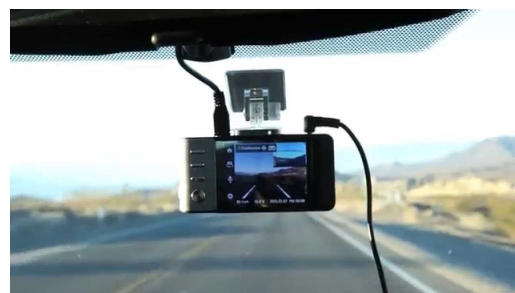


Figura 10: Ilustração da posição da câmera na parte interna do veículo (INAVI, 2015).

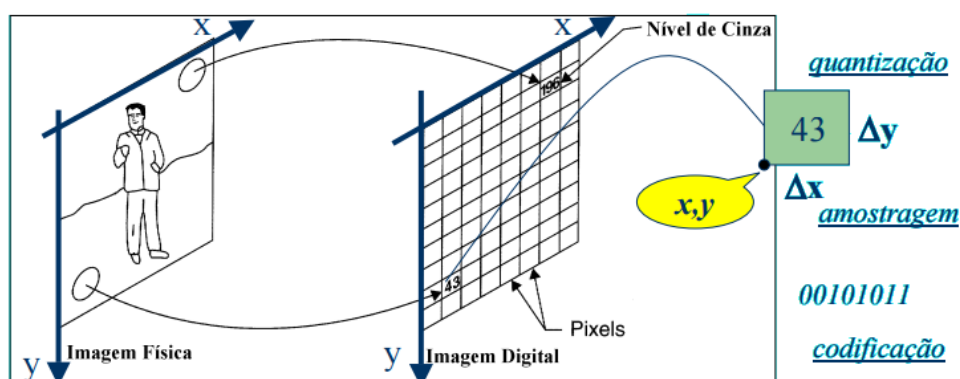


Figura 11: Ilustração do processo de aquisição de imagens digitais (THOME, 2004).

De acordo com Gonzalez (2006) o processo de aquisição de imagens digitais, consiste em converter cenas reais em imagens digitais. Para entender o que é uma aquisição de imagem, deve se saber que isto é o processo de conversão de uma cena real tridimensional em uma imagem analógica. O processo de aquisição de uma cena tridimensional em uma imagem eletrônica é um processo ótico de redução de dimensionalidade, ou seja, uma câmera digital converterá a cena 3D do ambiente em uma representação 2D eliminando a informação da profundidade de cada ponto adquirido. A Figura 11 ilustra o conceito de aquisição de imagem descrita a cima.

Assim como as câmeras convencionais, as digitais possuem uma variedade de lentes, que servem para direcionar a luz do ambiente para o sensor. Porém, ao invés de utilizar um filme fotográfico, é utilizado um sensor semicondutor, que registrar eletricamente a luz incidente sobre as lentes.

O sensor utilizado para fazer a captura da imagem é um dispositivo capaz de converter a energia luminosa de um ponto da imagem em carga elétrica, processando-a através de um circuito eletrônico específico e formando, assim, a imagem digital. A partir daí, o processo divide-se de acordo com o tipo de sensor que o aparelho possui, CCD ou CMOS (SHOPSEG, 2016).

A tecnologia CCD (*charge coupled device*) é uma matriz de células semicondutoras fotossensíveis, que trabalham como capacitores, fazendo um armazenamento da carga elétrica necessários para transformar a imagem do sensor CCD para a forma digital.

Hoje em dia o dispositivo de conversão mais utilizado usam a tecnologia CMOS (*Complementary Metal Oxide Semiconductor*). Essa tecnologia utiliza transistores alocados em cada pixel, para formar imagens digitais, sem a necessidade de conversores A/D (SHOPSEG, 2016).

Quando se quer capturar uma imagem colorida é necessário a utilização de um conjunto de prismas e filtros de cor, que tem a função de decompor a imagem colorida em RGB. Os veículos autônomos geralmente utilizam câmeras do tipo estéreo para fazer a captura da imagem da pista e do cenário que o veículo está trafegando, e ao mesmo tempo, para capturar as informações de profundidade dos objetos presentes no caminho do veículo no ambiente. Esse tipo de aquisição permite que seja levada em consideração a posição espacial relativa dos objetos presentes no pista na hora de se determinar a trajetória do veículo na pista. Um câmera estéreo é um tipo de câmera com duas ou mais lentes, espaçadas igualmente e que realizam a captura da imagem individualmente. Isso permite que a câmera simule a visão humana binocular, provendo-lhe a capacidade de capturar imagens tridimensionais do ambiente ¹.

3.3 Região de Interesse (ROI)

Após a aquisição da imagem o primeiro procedimento executado pelo sistema é a extração de uma região de interesse 1 (ROI-1) da imagem da pista. Essa região é isolada porque a parte superior da imagem (região da vista da pista que engloba os prédios, árvores e o céu) não contém qualquer informação relevante para o processo de aquisição das faixas de sinalização e a parte inferior da vista da pista sofre oclusão da parte frontal do veículo. Essa região extraída é diretamente relacionada com a posição da câmera no veículo. Nessa ótica, foram definidas regiões de interesse diferentes para cada banco de dados utilizados para validação deste trabalho.

Para o banco de dados da Caltech o ponto de fuga da imagem (*Vanishing Point*) foi

¹Esse processo também é conhecido como fotografia estéreo.

definido como sendo o limite superior da região de interesse. O ponto de fuga é um ponto do plano de visão, que representa a interseção aparente de duas, ou mais, retas paralelas, segundo um observador num dado momento² (MACHADO, 1983). Esse ponto é fixo para todas as imagens pois a posição da câmera não é alterada durante o trajeto do veículo. O limite inferior foi definido como o ponto na imagem onde a parte dianteira do veículo não aparece. A Figura 12 ilustra a região de interesse definida para o banco de dados da Caltech.



Figura 12: Ilustração da primeira região de interesse (ROI-1) que isola a região da imagem onde a pista está contida utilizada no banco de dados da Caltech. Adaptado de (ALY, 2008).



Figura 13: Ilustração da primeira região de interesse (ROI-1) que isola a região da imagem onde a pista está contida utilizada no banco de dados da UFRGS. Adaptado de (PAULA, 2015).

²O Vanishing Point utilizado nesse trabalho foi obtido através de inspeção visual nas imagens do banco de dados utilizado para o desenvolvimento e testes do sistema.

Para o banco de dados da UFRGS o ponto de fuga da imagem (*Vanishing Point*) foi definido como sendo o limite superior da região de interesse. Não foi definido um limite inferior visto que nesse banco de dados a carcaça do veículo não causa oclusão na vista da pista. A Figura 13 ilustra a região de interesse definida para o banco de dados da UFRGS.

As oscilações e vibrações devido ao movimento do veículo sobre a pista tem efeito mínimo sobre a extração das faixas de sinalização devido a posição da região de interesse (ROI-I). Entretanto, nesse trabalho não foram considerados cenários onde o veículo passa por regiões da pista com muitos buracos e lombadas.

3.4 Filtro de Mediana

A remoção do ruído em imagens digitais é uma das tarefas mais comuns em sistemas que utilizam visão computacional e processamento de imagem. Ruídos são quaisquer tipos de informações indesejáveis nas imagens. Eles podem ser causados por vários fatores (SANCHES et al., 2015), como a variação de brilho ou problemas no processo de conversão analógica-digital. Assim, é essencial tratar as imagens para que a extração de informação de contornos na imagem de forma confiável.

Os filtros passa baixas, também conhecidos como filtros de suavização, segundo Gonzalez (2006), são muito utilizados na remoção de ruídos de alta frequência espacial em sistema de visão computacional e processamento de imagem. Uma das principais dificuldades dos métodos de suavização é o borrimento dos detalhes da imagem filtrada.

Foram testados diversos tipos de filtros de suavização como: Filtro de Média, Filtro de Mediana, Filtro de Média Ponderada e Filtro Gaussiano. O filtro de Mediana foi o que apresentou o melhor resultado entre os filtros testados em critérios de custo computacional e desempenho. Embora existam muitos outros filtros de suavização disponíveis na literatura, o filtro de mediana é um meio termo em termos de complexidade de implementação e custo computacional.

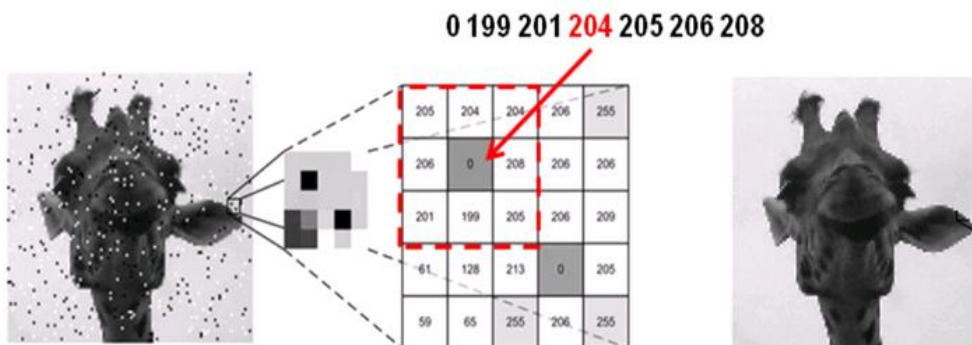


Figura 14: Ilustração do funcionamento de um filtro de mediana em uma imagem ruidosa. Adaptado de (WHATWHENHOW, 2016).

O Filtro da mediana é um filtro não-linear baseado em estatística de ordenação (*order-statistic filter*), utilizado para suavizar ruídos do tipo impulsivo (*salt and papper*) em sinais e imagens digitais. A filtragem provoca uma suavização da imagem, ao descartar pontos muito altos ou muito baixos com relação a mediana dos píxeis vizinhos. Essa operação desloca uma janela, de tamanho ajustável, através de toda a imagem, pixel a pixel, substituindo o elemento central dessa janela pelo valor da mediana das intensidades

dos píxeis que estiverem dentro da janela. Quanto maior for a janela, mais influência dos vizinhos este pixel sofrerá e maior será o efeito de *blurring*, visto que levaremos em consideração um número maior de píxeis. A Figura 14 ilustra uma imagem poluída por ruído do tipo *salt and pepper* e como o filtro de mediana elimina esse ruído através da remoção das variações abruptas de píxeis nessa imagem. Onde uma janela de tamanho 3×3 é deslocada através da imagem, os píxeis dentro dessa janela (píxeis da vizinhança) são ordenados em ordem crescente e o pixel central da janela é substituído pelo valor central da sequência ordenada. Essa operação é feita sobre todos os píxeis da imagem.

3.5 Transformação de Perspectiva

A transformação de perspectiva é um método de conversão matemático que transformar o plano espacial de uma imagem em outro. É utilizada em uma ampla variedade de sistemas de detecção de faixas de sinalização disponíveis na literatura, como os de Aly (2008), Lee (2016), Li (2012), Liu (2010). Essa operação quando aplicada sobre uma pista é conhecida como vista *Bird's Eye*, quando o plano "destino" corresponda a uma câmera de topo. Em comparação com a imagem original da câmera, a imagem no formato *Bird's Eye* proporciona diversas vantagens para os sistema de detecção de faixas de sinalização.

A primeira vantagem é que nessa perspectiva as faixas de sinalização ficam igualmente alinhadas fluindo na mesma direção do ponto de fuga da imagem, sendo uma a cópia da outra em termos de inclinação, o que traz comodidade para as técnicas processamento de imagem para extração de detalhes da imagem. A segunda vantagem é que a largura das faixas é quase invariante dentro da imagem inteira, o que simplifica as etapas de extração das linhas. Outra vantagem dessa perspectiva é que os detalhes mais distantes da imagem ficam mais visíveis e sua identificação mais acessível para o sistema. Um exemplo da vista *Bird's Eye* pode observada na Figura 15.



Figura 15: (1) A imagem da pista na escala de cinza e filtrada por um filtro de mediana e (b) A imagem de saída da transformação de perspectiva (*Bird's Eye*).

Segundo Gonzales (2006), a transformação de perspectiva (*Perspective Mapping*) é uma técnica de transformação geométrica que projeta o plano de um objeto 3D em um plano com perspectiva 2D. Ela uma aproximação da maneira como a imagem é formada através de um mundo 3-D. Matematicamente, a transformação de perspectiva pode ser descrita como uma projeção de um espaço euclidiano 3D, $\mathbf{p}_i = (x, y, z) \in E^3$ (espaço de coordenadas mundo) sobre um plano 2D, $\mathbf{p}'_i = (u, v) \in E^2$, ou seja, $\mathbf{p}_i \rightarrow \mathbf{p}'_i$ (BERTOZZ;

BROGGI; FASCIOLI, 1998; BROGGI, 1995).

Diversas técnicas de transformação de perspectiva estão disponíveis na literatura (HARTLEY; ZISSERMAN, 2000). Uma das técnicas de transformação de perspectiva mais populares é a transformação Homográfica. Ela é uma a técnica de geo-referenciamento baseada em conceitos geométricos e matemáticos de coordenadas homogêneas e planos projetivos que realiza uma transformação linear de raio equivalentemente, isto é, transforma linearmente o plano da imagem. Ela possui a vantagem de não precisar de algumas características da câmera como distancia focal, altura em relação ao solo, ângulo vertical e horizontal em relação a direção da estrada, entre outros, e por esse motivos foi escolhida para ser utilizada neste trabalho. Todas as imagens formadas por todos os planos que cortam o mesmo feixe de raios estão relacionadas por homografia (CORRMAP, 2013). A Figura 16 ilustra esse conceito.

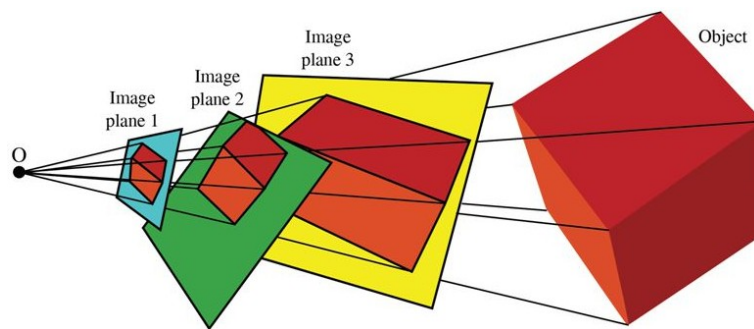


Figura 16: Ilustração dos planos da imagem que cortam o mesmo feixe de raios (PRINCE, 2011).

A homografia transforma um quadrilátero de um sistema de referência em um quadrilátero correspondente em outro sistema de referência. Na homografia entre dois planos, dado quatro pontos em um plano, sempre existe uma relação que os transforma em quatro pontos correspondentes em outro plano, em outras palavras, neste cenário cada ponto 2D pode ser projetado em qualquer outro plano no espaço. A Figura 17 ilustra a transformação planar de quatro pontos definidos em um sistema de referência para um outro sistema de referência.

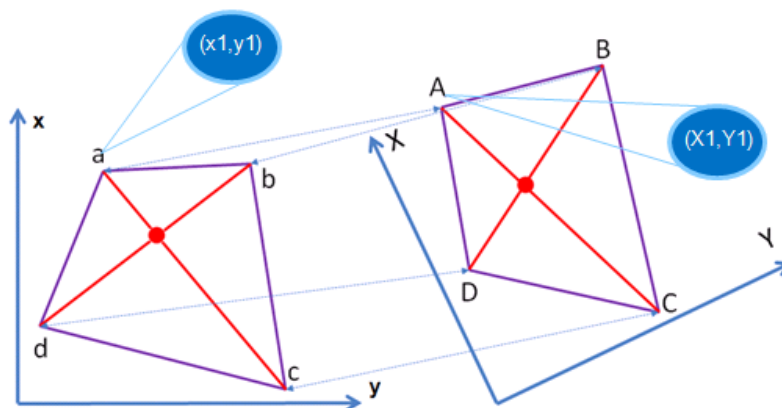


Figura 17: Ilustração da transformação de um quadrilátero de um sistema de referência em um quadrilátero correspondente em outro sistema de referência. Adaptado de (CORRMAP, 2013).

O mapeamento homográfico dos oito pontos definidos na Figura 17 é uma transformação planar projetiva, e pode ser expresso como Equação (2), e sua decomposição em (3) e (4), para uma forma homogêfica. Este mapeamento homográfico é uma transformação linear. A matriz não singular \mathbf{H} , de dimensão 3×3 , é chamada matriz de transformação homogênea, e age como um operador linear que mapeia os píxeis no espaço tridimensional para o espaço bidimensional (LIN; WANG, 2012).

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \Rightarrow \mathbf{p}'_i = \mathbf{H} \cdot \mathbf{p}_i \quad (2)$$

$$X_1 = \frac{x_1}{z_1} = \frac{h_{11}x_1 + h_{12}y_1 + h_{13}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (3)$$

$$Y_1 = \frac{y_1}{z_1} = \frac{h_{21}x_1 + h_{22}y_1 + h_{23}}{h_{31}x_1 + h_{32}y_1 + h_{33}} \quad (4)$$

Onde as variáveis X_1 e Y_1 são as coordenadas a serem calculadas no segundo sistema de referência, em função de oito parâmetros de transformação definidos $h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}$, dadas coordenadas x, y e z no primeiro sistema de referência. A variável z é atribuída com o valor 1, pois não possuímos a informação de profundidade da imagem e o parâmetro h_{33} também é atribuído com o valor 1, para utilizarmos apenas oito coordenadas (CORRMAP, 2013). Esses oito pontos são os pontos que definem o quadrilátero a, b, c, d e o quadrilátero A, B, C e D ilustrados na Figura 17. A solução das Equações (3) e (4) nos fornece os parâmetros que compõe a matriz transformação homográfica \mathbf{H} da Equação (2).

A partir das Equações (3) e (4) obtemos o sistema de equação de oito incógnitas, evidenciado na Equação (5), do qual extraímos as componentes da matriz homográfica \mathbf{H} , apresentada na equação (2). Os oito pontos utilizados para obter a matriz homográfica são escolhidos individualmente, quatro deles na imagem do plano original da imagem, onde vão limitar a região da imagem que se deseja mudar a perspectiva, e quatro no respectivo plano que se deseja projetar a imagem.

A matriz \mathbf{H} possui nove entradas e oito graus de liberdade. Baseado nos graus de liberdade de \mathbf{H} , um limite inferior n , que determina a quantidade de pontos correspondentes $\mathbf{p}_i \leftrightarrow \mathbf{p}'_i$ necessários para computar a transformação projetiva \mathbf{H} pode ser estabelecido. Cada par de pontos correspondentes tem dois graus de liberdade, pois as coordenadas de um ponto \mathbf{p}_i são determinadas por dois elementos x e y (o fator de escala homogêneo é arbitrário) e devem corresponder ao ponto mapeado $\mathbf{H} \cdot \mathbf{p}_i$. Assim, pelo menos quatro pontos correspondentes são necessários ($n = 4$) (SANTOS, 2012).

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -y_1X_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -y_2X_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3X_3 & -y_3X_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4X_4 & -y_4X_4 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1Y_1 & -y_1Y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2Y_2 & -y_2Y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -x_3Y_3 & -y_3Y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -x_4Y_4 & -y_4Y_4 \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} \quad (5)$$

A matriz de transformação homogênea obtida através desse oito pontos é então multiplicada por todos os píxeis da imagem, agindo como um operador linear que vai mapear

os píxeis distribuídos num plano tridimensional em um plano bidimensional, formando assim a vista *Bird's Eye* da pista.

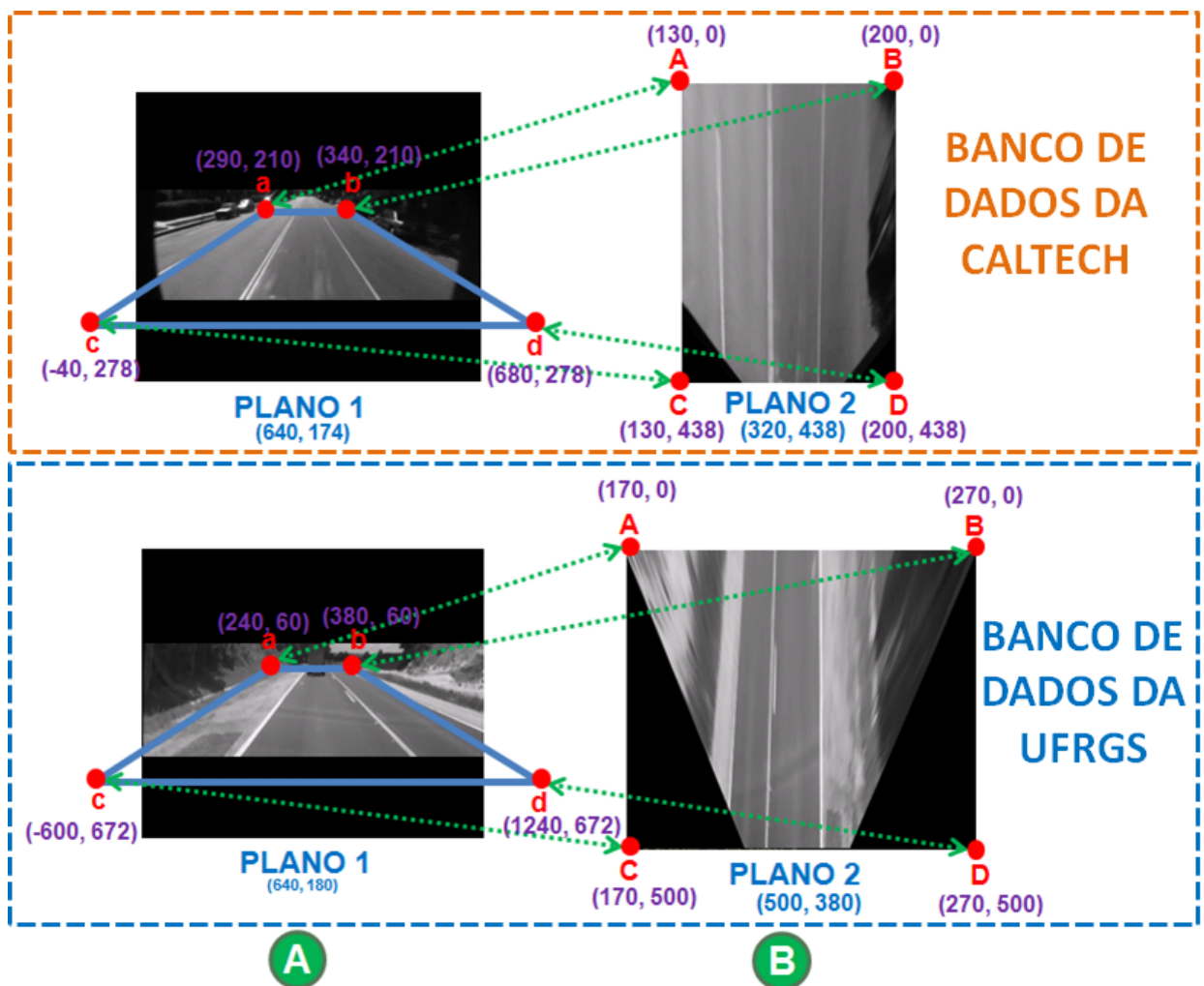


Figura 18: As figuras da coluna da esquerda (A) ilustração os 4 primeiros pontos geométricos que formam uma área trapezoidal da superfície da estrada (PLANO 1). As figuras da coluna da direita (B) ilustração os 4 últimos pontos geométricos que formam uma área retangular (PLANO 2), este pontos são utilizados para formar a matriz de transformação homográfica.

O principal problema desse método de transformação geométrica é como selecionar os pontos, isto é, os vetores $(x_1, y_1, z_1)^T$ aplicados na Equação (2), para realizar a transformação, pois diferentes pontos selecionados produzem soluções diferentes. Neste trabalho, esses pontos foram definidos experimentalmente visando se obter a melhor vista da pista. O resultado final é uma imagem que nos fornece uma vista da pista na perspectiva superior do espaço tridimensional. Os pontos definidos neste trabalho e suas respectivas coordenadas, para os dois bancos de dados utilizados para validação, estão ilustradas na Figura 18. Esses pontos foram definidos de forma manual. Foram definidos pontos diferentes par cada banco de dados utilizados para validação desse trabalho, pois a posição da câmera e sua altura em relação ao solo varia entre os bancos.

A transformação inversa de perspectiva é realizada através da multiplicação de todos os píxeis da imagem na vista *Bird's Eye* pela inversa da matriz homográfica. Contudo,

a transformação de perspectiva elimina algumas das informações referentes a geometria espacial original da imagem, o que conseqüentemente faz com que a reconstrução da imagem na sua perspectiva original fique levemente distorcida. Com isso em mente, neste trabalho, apenas aplicamos a transformada inversa nos pontos que definem os segmentos de linha que descrevem as faixas de sinalização, o que torna esse efeito irrelevante.

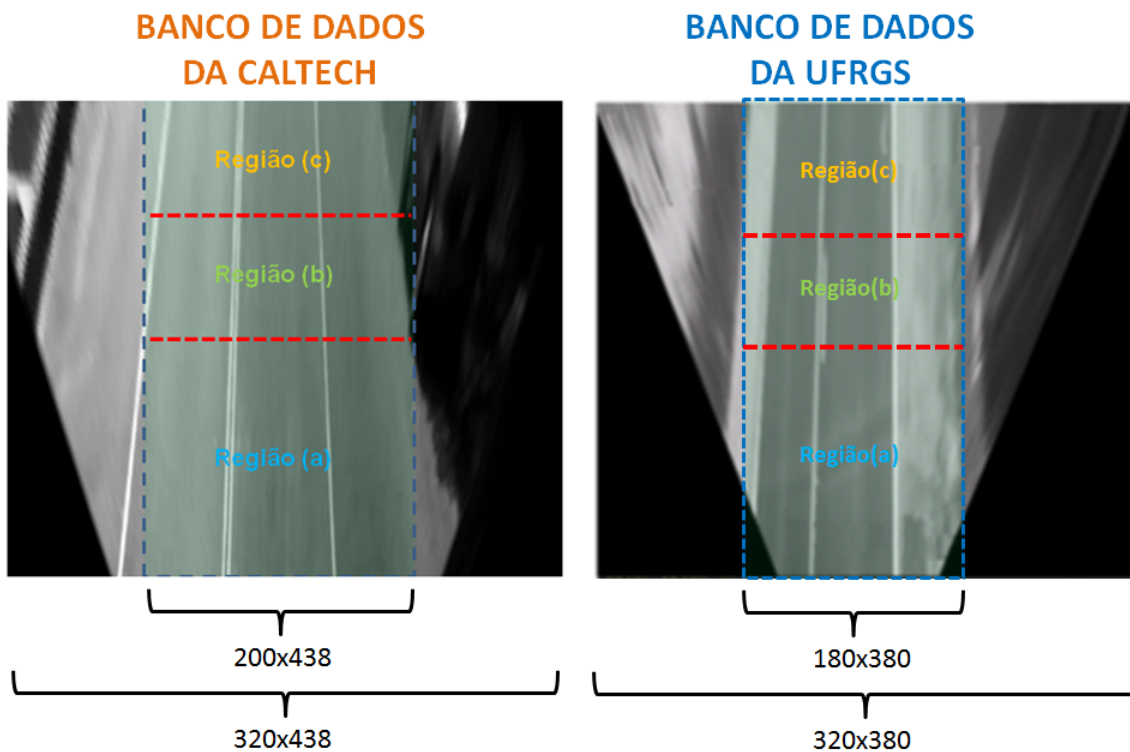


Figura 19: Ilustração da área central isolada da imagem da pista (ROI-2), onde as faixas de sinalização estão contidas (retângulo esverdeado), e da divisão realizada nessa área em três regiões menores: Região (a), Região (b) e Região (c).

Por fim, após a transformação de perspectiva é feito o isolamento da segunda região de interesse (ROI-2), que engloba somente as regiões da imagem onde as faixas de sinalização e bordas da pista estão contidas. O isolamento da região de interesse vai modificar a largura à imagem, concentrando a região da imagem onde as faixas estão contidas. A imagem é então dividida em três regiões: região (a), região (b) e região (c) para que dessa forma a detecção do sistema possa ser capaz de detectar a curvatura da pista com mais precisão. Estes procedimentos estão ilustrados na Figura 19. A altura da região (a), limitada pela primeira linha tracejada na cor vermelha, de baixo para cima, é igual a metade do tamanho total da imagem. A altura da região (b) é igual a três quartos da altura total da imagem.

3.6 Filtro de Limiar Adaptativo

A limiarização ou binarização de imagens é uma tarefa comum em aplicações gráficas e de visão computacional. Esse é o método mais simples de segmentação de imagem, método no qual é feita a classificação dos píxeis de uma imagem como "escuros" ou "claros", isto é, é uma operação não-linear que converte uma imagem de escala de cinza

em uma imagem binária onde os dois níveis são atribuídos a pixels que estão abaixo ou acima do valor limite especificado. Segundo Gonzalez (2006), a atribuição convencional é que se o pixel possui um valor maior que um valor de limite pré-definido (limiar), é atribuído ao pixel o valor 255 que representa a cor branca, caso contrário é atribuído o valor 0 que representa a cor preta.

Porém, usar um valor de limiar global pode não ser uma boa escolha para limiarizar imagens da pista de cenários urbanos, uma vez que a pista vai estar exposta a diversas condições de iluminação, sombras e pavimentação com cores diferentes tornando a utilização de um único valor de limiar inadequado para toda a imagem. Esses problemas ficam evidentes na Figura 20, que mostra uma imagem com variação de iluminação e os resultados obtidos com diferentes métodos de limiarização. Onde em (A) é apresentada uma imagem em escala de cinza; (B) é apresentada a mesma imagem limiarizada por um valor de limiar global; (C) mostra limiarizada pelo filtro de Otsu; (D) mostra a imagem limiarizada por um filtro de limiar adaptativo utilizando a média dos pixels para definir o valor de limiar e em (E) utilizando a soma ponderada dos valores dos pixels dentro de uma janela gaussiana.

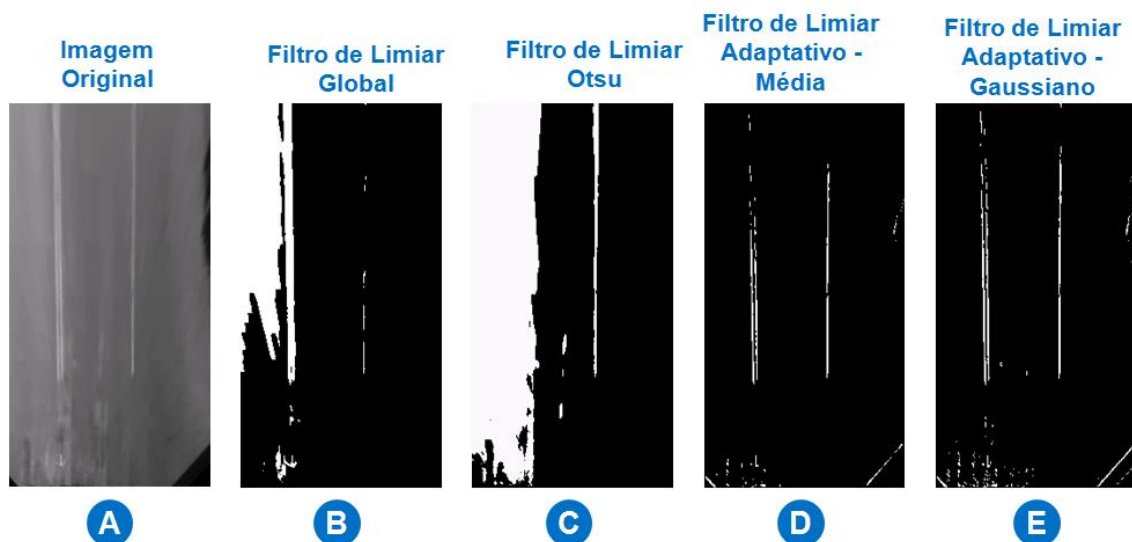


Figura 20: Ilustração dos diferentes métodos de limiarização de imagens.

Levando isso em consideração, neste trabalho, optamos por utilizar um filtro de limiar adaptativo Gaussiano. O filtro de limiar adaptativo foi publicado por Bradley (2007). Ao invés de definir um valor de limiar para toda a imagem, o algoritmo divide a imagem em várias regiões pequenas, de tamanho pré-definido ($BlockSize \times BlockSize$), e calcula o valor de limiar para cada uma dessas regiões, de modo que são obtidos diferentes limiares para diferentes regiões da mesma imagem, melhorando os resultados do processo de limiarização em imagens com diferentes condições de iluminação. O valor de limiar é definido como sendo o valor da média das intensidades dos pixels presentes dentro da pequena região. Esse processo é necessário pois a transformada de Hough utilizada para encontrar linhas, requer uma imagem binarizada como entrada. A Equação (6) representa o processo de limiarização realizado em cada região:

$$image(x, y) = \begin{cases} maxValue & \text{if } image(x, y) > T(x, y) - offset(C) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Onde $T(x, y)$ é um limiar calculado individualmente para região, igual à soma ponderada dos valores resultantes de uma correlação cruzada com uma janela gaussiana dos píxeis da imagem contidos dentro da região da imagem, subtraída de um valor de $offset(C)$. O parâmetro "maxValue" é o valor para qual o pixel na imagem é substituído caso ele seja maior que o valor de limiar da região. Os elementos da matriz Gaussiana (\mathbf{G}) (de dimensão $BlockSize \times 1$) são obtida através da Equação (7).

$$\mathbf{G}_i = \alpha e^{-\frac{(i - \frac{BlockSize-1}{2})^2}{4\sigma^2}} \quad (7)$$

As componentes da matriz Gaussiana segundo Bradley (2007) são definidos da seguinte forma: $i = 0 \dots BlockSize-1$, α é o fator escalar escolhido tal que $\sum_i \mathbf{G}_i = 1$ e $\sigma = 0.3((BlockSize - 1)0.5 - 1) + 0.8$. Devido à distorção causada pelo processo de transformação de perspectiva que forma a vista *Bird's Eye*, adotamos configurações diferentes do filtro de limiar adaptativo para cada uma das três regiões em que a imagem é dividida. O valor de $offset(C)$ para a região (a) utilizado foi de -13 , para a região (b) de -9 e para a região (c) -4 . O parâmetro $BlockSize$ foi definido igualmente para todos como 7.

Neste trabalho a binarização da imagem ocorre somente após a transformação de perspectiva. Foram realizados experimentos binalizando-se a imagem antes de realizar a transformação de perspectiva e não forma obtidos bons resultados.

3.7 Transformada de Hough

A transformada de Hough é o método mais utilizados para a detecção de estruturas geométricas em imagens digitais que podem ser parametrizados, como linhas, círculos, elipses, além de poder ser aplicado no reconhecimento de padrões de curvas que possam ser descritas na forma paramétrica (DUAN et al., 2010; LEI; WONG, 1999; CHEN; CHUNG, 2001) e por este motivo foi escolhido como método para extrair as linhas que descrevem as faixas de sinalização neste trabalho. Foi criado por Paul Hough em 1962, que o chamou de "transformada Hough generalizada" (HOUGH, 1962). Neste trabalho a transformada de Hough é utilizada para detectar as linhas que descreverem as faixas de sinalização nas áreas de busca na imagem. Normalmente, para a aplicação desta transformada, é realizado um pré-processamento na imagem com o objetivo de identificar claramente os contornos dos elementos que a compõem, a fim de que sejam considerados apenas os píxeis das bordas que se deseja identificar.

Em síntese, o objetivo principal da transformada de Hough é transformar a imagem do espaço digital (x, y) para uma representação no respectivo espaço de parâmetros, utilizando as coordenadas polares ρ e θ para definir o segmento de reta, ao invés de inclinação e intersecção. Essa transformação é aplicada de modo que todos os pontos pertencentes a uma mesma reta sejam mapeados em um único ponto no espaço dos parâmetros usando a Equação (8).

$$\rho = x \cos(\theta) + y \sin(\theta). \quad (8)$$

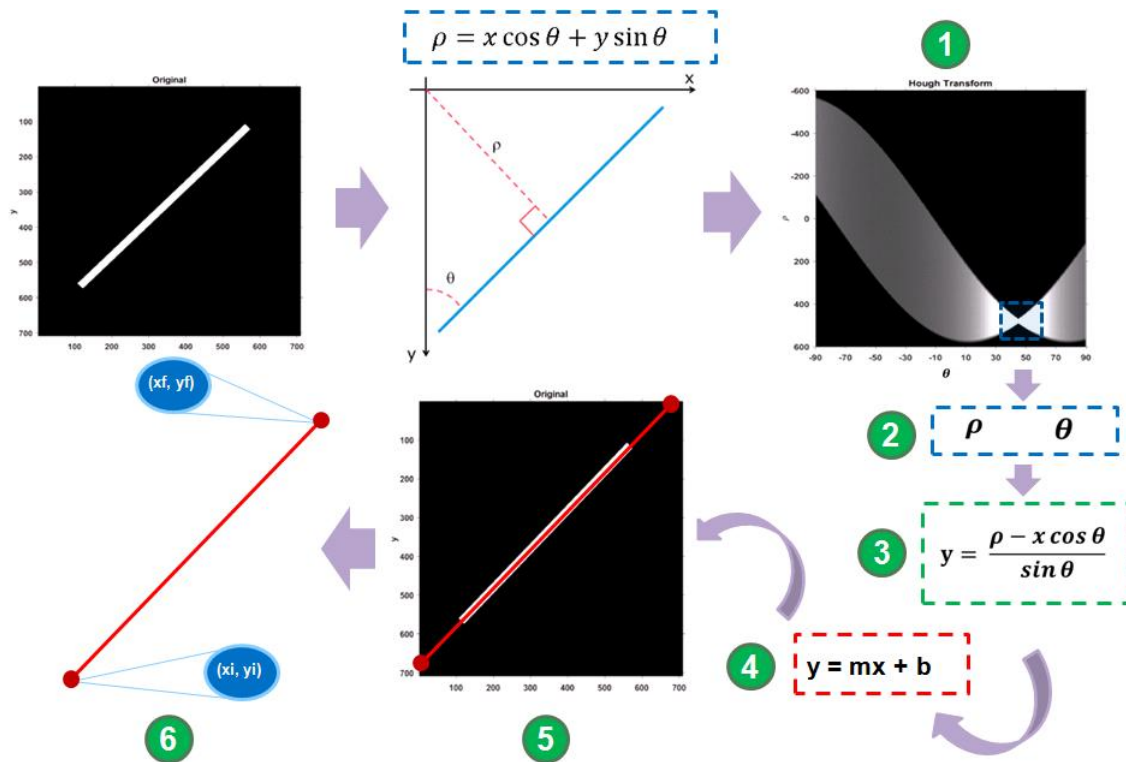


Figura 21: Ilustração do procedimento de detecção de linhas da transformada de Hough.

Onde, ρ denota a distância da origem do plano (x,y) ao longo de um vetor perpendicular à linha, e θ denota o ângulo entre o eixo x e este vetor. Essa equação converte todos os pixels da imagem para o espaço de parâmetros, cujas linhas e colunas correspondem aos valores ρ e θ , respectivamente.

Neste trabalho atribuímos o valor de resolução para o eixo ρ do espaço de parâmetros igual a 1 e para eixo θ igual a 20. A saída da transformada de Hough são os valores de ρ e θ de todas as linhas encontradas pela transformada. Esses valores são utilizados para extraímos a equação da reta de cada linha encontrada. Figura 21 ilustra a ideia fundamental de como a transformada de Hough extrai as linhas contidas em uma imagem.

Nessa figura, (1) representa o espaço de parâmetros para o qual a imagem limiarizada que contém uma linha foi transformada, (2) são os valores de ρ e θ do pico de maior concentração de pixels (indicado pelo retângulo tracejado azul) que foi identificado no espaço de parâmetros, na etapa (3) reescrevemos a equação de Hough em função de y e x e em (4) utilizando os valores de ρ e θ identificados nessa equação para obtemos a equação da reta que descreve a linha na imagem limiarizada ilustrada em (5). Por fim, em (6) extraímos as coordenadas de início e fim dessa linha. Se houvessem mais linhas na imagem limiarizada, seriam identificados outros picos de concentração de pontos, que gerariam outras equações da reta (GONZALEZ; WOODS, 2006).

Neste trabalho, a extração das coordenadas de início e fim das linhas encontradas pela transformada de Hough, a partir dos valores de ρ e θ extraídos do espaço de parâmetros, é feita através das seguintes equações:

$$x_1 = \frac{\rho}{\cos(\theta)}x - h \cdot \frac{\sin(\theta)}{\cos(\theta)} \quad (9)$$

$$y_1 = h \quad (10)$$

$$x_2 = \frac{\rho}{\cos(\theta)} \quad (11)$$

$$y_2 = 0 \quad (12)$$

Nessas equações, (x_1, y_1) é a coordenada de início da linha encontrada e (x_2, y_2) é a coordenada do fim da linha, h é a altura da imagem onde estamos procurando pelas linhas e w é a largura. Logo, utilizando as equações a cima, encontramos as coordenadas de início e fim das linhas identificadas. Porém, muitas vezes vamos obter coordenadas horizontais menores que 0 ou maiores que w . Para deixar todas as coordenadas de início e fim das linhas dentro do tamanho da imagem utilizamos a seguinte equações:

$$x_1 \geq w = \begin{cases} y_1 = y_1 - |(x_1 - w) \cdot \frac{\sin(\theta)}{\cos(\theta)}| \\ x_1 = w \end{cases} \quad (13)$$

$$x_2 \geq w = \begin{cases} y_2 = y_2 - |(x_2 - w) \cdot \frac{\sin(\theta)}{\cos(\theta)}| \\ x_2 = w \end{cases} \quad (14)$$

$$x_1 \leq 0 = \begin{cases} y_1 = y_1 - |x_1 \cdot \frac{\sin(\theta)}{\cos(\theta)}| \\ x_1 = 0 \end{cases} \quad (15)$$

$$x_2 \leq 0 = \begin{cases} y_2 = y_2 - |x_2 \cdot \frac{\sin(\theta)}{\cos(\theta)}| \\ x_2 = 0 \end{cases} \quad (16)$$

Uma vez que, as marcas de pista não estão próximas da origem e não são horizontais na imagem, apenas detectamos as linhas retas com restrição na faixa de ângulos de 0 a 40 graus e 140 a 210 graus, para reduzir o número de linhas falso-positivas e reduzir o custo no tempo de processamento. Mesmo com essa restrição nos ângulos, a transformada de Hough retorna, quase sempre, mais de uma linha na área de busca, devido ao ruído presente na imagem e a deterioração na sua forma após a utilização do filtro de limiar adaptativo para binarizar a imagem. O algoritmo MSLC é utilizado para determinar qual dessas linhas obtidas é a que possui a maior probabilidade de ser a linha que descreve a faixa de sinalização na área de busca, baseado na distribuição e no número de píxeis que essas linhas englobam.

3.8 Método de Seleção de Linha Candidata (MSLC)

O Método de Seleção de Linha Candidata (MSLC) utilizado neste trabalho foi definido de forma semelhante ao algoritmo RANSAC (*R*andom *S*ample *A*nd *C*onsensus). O algoritmo RANSAC é um método iterativo não determinístico, baseado no princípio de geração e verificação de hipóteses, para estimar parâmetros de um modelo matemático a partir de um conjunto de dados observados que contém *outliers*³, utilizado para produzir

³Um dado é considerado um *outliers* se ele não se enquadra no modelo delimitado pelo conjunto de parâmetros dentro de algum limite de erro que define o desvio máximo atribuível ao efeito do ruído (ZULIANI, 2012)

um resultado razoável apenas com uma certa probabilidade, que aumenta com o número de iterações realizadas (ZULIANI, 2012).

Foi proposto por Fischler (1981) que descrevem uma abordagem geral de estimação de parâmetros projetada para lidar com uma grande proporção de *outliers* nos dados de entrada. A porcentagem de *outliers* que podem ser manipulados pela RANSAC pode ser maior do que 50% de todo o conjunto de dados (ZULIANI, 2012). Essa porcentagem, também conhecida como ponto de ruptura, é comumente assumida como o limite prático para muitas outras técnicas comumente usadas para estimação de parâmetros (HUBER; RONCHETTI, 2009; ROUSSEEUW; LEROY, 1987; MENDEL; SHYNK, 1995).

Neste trabalho, o algoritmo RANSAC foi escolhido como base para o MSLC por apresentar métricas de seleção de parâmetros baseadas na concentração e quantidade de pontos de uma estrutura que podem ser utilizados para avaliar a sua qualidade, ou seja, foi escolhido devido a sua capacidade de analisar um universo de pontos e determinar a localização de uma estrutura baseada em um modelo matemático e atribuir um valor relacionado a qualidade desta estrutura. O algoritmo RANSAC utilizado como base para o MSLC está descrito no Algoritmo 1:

Algorithm 1: RANdom Sample Consensus (RANSAC) algorithm

- 1 Defina um modelo matemático, por exemplo a equação da reta.
 - 2 Selecione aleatoriamente o número mínimo de pontos necessário para determinar os parâmetros do modelo.
 - 3 Resolva o modelo matemático utilizando os pontos selecionados.
 - 4 Determine a distância ortogonal de todos os pontos do conjunto de pontos, que estão dentro da área de tolerância predefinida ε . E para cada iteração é atribuído um peso relacionado ao número de pontos dentro da área limitada por ε .
 - 5 Se a pequena região do número de *inliers* sobre o número total de pontos no conjunto exceder um limiar predefinido τ , uma re-estimação dos parâmetros do modelo deve ser feita usando todos os *inliers* identificados e o algoritmo é finalizado.
 - 6 Caso contrário, repita os passos 1 a 4 (máximo de M vezes). Ao final das iterações o conjunto com o maior peso é a saída do algoritmo.
-

A Figura 22 ilustra as etapas (de forma geral) como o algoritmo RANSAC é executado para encontrar uma linha em um ambiente ruidoso. Conforme é apresentado na Figura, inicialmente as hipóteses aleatórias são geradas a partir de amostras de dados. Para realizar a validação do conjunto de dados selecionado dois pontos são escolhidos aleatoriamente do conjunto de amostras, uma linha imaginária é delimitada entre esses dois pontos e a distância perpendicular de todos os outros pontos que estiverem dentro da distância de tolerância ε definida é calculada. Caso a amostra esteja dentro do limite ε pré-definido, a amostra é considerada uma *inlier*, caso contrário uma *outlier*. Para cada iteração realizada é calculado um peso em relação à quantidade de *inliers* encontrados. Caso o peso encontrado seja menor ou igual a um limiar τ pré-determinado o algoritmo para, caso contrário, executa-se uma nova iteração para novas hipóteses aleatórias até chegar no valor de limiar ou executar o número máximo de iterações (M) pré-definido.

O número de iterações M , é escolhido de forma a garantir que a probabilidade p (geralmente definida como 0,99) dos conjuntos de amostras aleatórias não inclua um *outlier*. Seja m o número de pontos necessários para validar o modelo de estimação, u a probabilidade de que qualquer ponto dentre os m pontos do conjunto de amostras seja um

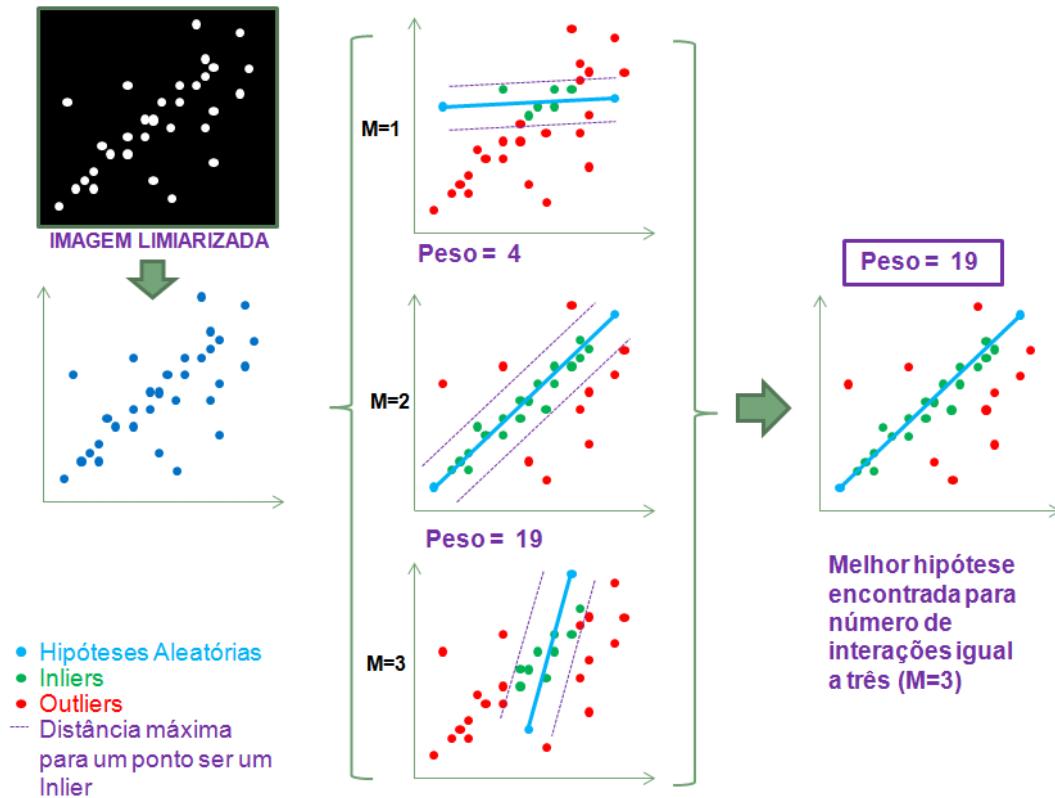


Figura 22: Ilustração do procedimento de estimação RANSAC de uma reta.

inlier, e seja $v = 1 - u$ a probabilidade de que ao menos um dos pontos selecionados seja um *outlier*. Logo, podemos calcular o número mínimo de iteração M , a partir da Equação (17), necessárias para estimar o conjunto de amostras que apresenta a maior probabilidade de ser uma linha (DERPANIS, 2010).

$$1 - p = (1 - u^m)^M \quad (17)$$

Onde $(1 - u^m)^M$ representa a probabilidade de que nunca seja escolhido um conjunto de amostras onde todas as amostras são *inliers* (que deve ser a mesma probabilidade que $1 - p$). Reorganizando a equação (17) obtemos a Equação (18) que fornece o número mínimo de iteração são necessárias para a execução do algoritmo de estimação apresentado,

$$M = \frac{\log(1 - p)}{\log(1 - (1 - v)^m)} \quad (18)$$

O algoritmo do MSLC utilizado neste trabalho é definido de forma quase idêntica ao algoritmo de RANSAC ilustrado no *Algorithm 1*. A única alteração realiza no algoritmo foi na forma como ele escolhe as hipóteses que vão ser testadas (Etapa 2), isto é, ao invés de escolher aleatoriamente alguns pontos do universo de pontos que vão ser analisados, usamos os pontos de início e fim das linhas fornecidas pela transformação de Hough, para que seja identificada qual dessas linhas é a que possui a maior probabilidade de ser a linha que descreve a faixa de sinalização na imagem da pista, utilizando as métricas do algoritmo RANSAC para fazer essa verificação. O número máximo de iterações M vai ser igual ao número total de linhas fornecidas pela transformadas de Hough. Logo, o MSLC é usado para selecionar qual das linhas encontradas pela transformada de Hough

é a que vai ser utilizada pelo sistema. O algoritmo MSLC está descrito no Algoritmo 2 e sua dinâmica de funcionamento está ilustrado na Figura 23:

Algorithm 2: Método de Seleção de Linha Candidata (MSLC)

- 1 Defina um modelo matemático
 - 2 Selecione os pontos de início e fim de uma das linhas fornecidas pela transformada de Hough.
 - 3 Resolva o modelo matemático utilizando os pontos selecionados.
 - 4 Determine a distância ortogonal de todos os pontos do conjunto de pontos, que estão dentro da área de tolerância predefinida ε . E para cada iteração é atribuído um peso relacionado ao número de pontos dentro da área limitada por ε .
 - 5 Se a pequena região do número de *inliers* sobre o número total de pontos no conjunto exceder um limiar predefinido τ , uma re-estimação dos parâmetros do modelo deve ser feita usando todos os *inliers* identificados e o algoritmo é finalizado.
 - 6 Caso contrário, repita os passos 1 a 4 (máximo de M vezes). Ao final das iterações o conjunto com o maior peso é a saída do algoritmo.
-

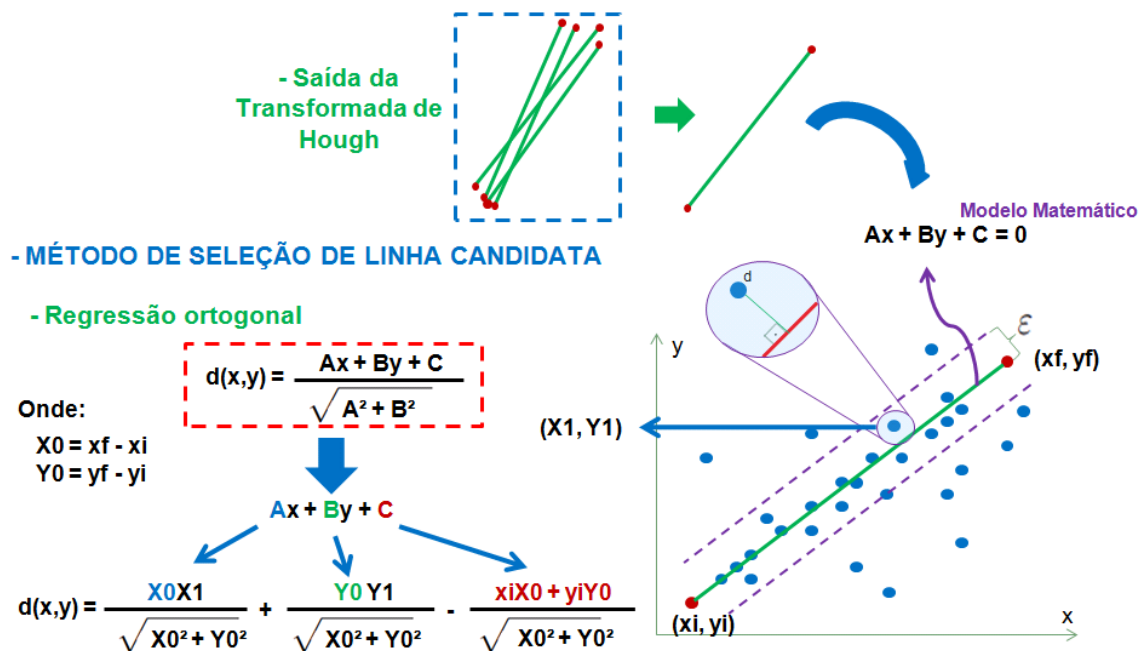


Figura 23: Ilustração do funcionamento geral do algoritmo MSLC.

3.9 Filtro de Kalman

O filtro de Kalman é uma ferramenta estatística utilizada para estimar os estados e minimizar o erro quadrático de um sistema linear dinâmico poluído por ruído gaussiano. Desde sua publicação por Rudolph E. Kalman (1960) o filtro Kalman foi uma componente fundamental dentro de milhares de sistemas de navegação tanto militares como civis, principalmente para integrar (ou fundir) os dados de sensores para atingir um maior

rendimento de todo o sistema (BISHOP; WELCH, 2006). O filtro de Kalman é ideal para sistemas que estão em constante mudança e que possuem pouca memória (só é necessário armazenar o estado anterior), além de ser executado rapidamente, tornando-o uma escolha promissora para atingir requisitos de tempo real e sistemas embarcados, e por estes motivos foi utilizado neste trabalho.

Neste trabalho, o filtro de Kalman é utilizado para corrigir as coordenadas de início e fim de todos os seis segmentos de linha que descrevem as duas faixas de sinalização na imagem da pista. Esses pontos de início e fim são afetados por fontes de incerteza na imagem e pela corrosão causada pela operação de limiarização. A segunda função do filtro de Kalman é prever a posição dos seis segmentos de linha na próxima imagem da pista, essas previsões são utilizadas para definir as áreas de busca 1 na próxima imagem da pista.

O filtro de Kalman é essencialmente um estimador de controle de tipo preditor-corretor retroalimentado, ele estima o processo em algum momento e obtém a retroalimentação medindo o sistema ruidoso (BISHOP; WELCH, 2006). Dessa forma, o filtro de Kalman funciona em duas etapas: uma de previsão e outra de correção.

Na etapa de previsão, definida pelas Equações (19) e (20), inicialmente calculamos o próximo estado do sistema ($\mathbf{X}_{k|k-1}$) a partir do estado atual ($\mathbf{X}_{k-1|k-1}$). Essa previsão é chamada de estado a priori e representa uma previsão feita pelo filtro de Kalman de qual vai ser o próximo estado do sistema. O filtro de Kalman gera no total seis previsões, uma para cada segmento de linha que descreve as faixas de sinalização, entorno dessas previsões definimos as áreas de busca 1, com largura de 20 píxeis e altura igual ao do segmento, que vão ser utilizadas na próxima imagem da pista.

$$\mathbf{X}_{k|k-1} = \mathbf{F}_{k|k} \mathbf{X}_{k-1|k-1} + \mathbf{B}_{k|k} \mathbf{U}_k + \mathbf{w}_k \quad (19)$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}_{k|k} \mathbf{P}_{k-1|k-1} \mathbf{F}_{k|k}^T + \mathbf{Q}_k \quad (20)$$

As matrizes $\mathbf{F}_{k|k}$ e $\mathbf{B}_{k|k}$ representam a relação entre o estado anterior, medido e o estado previsto, ou seja, é o modelo matemático do sistema a ser previsto na forma de espaço de estados que leve o sistema do estado $\mathbf{X}_{k-1|k-1}$ para o estado $\mathbf{X}_{k|k-1}$.

Os ruídos nas equações de estado \mathbf{w}_k e \mathbf{e}_k são chamados de ruído de processo e de medição, respectivamente e neste trabalho serão considerados como ruídos Gaussianos, não correlacionados e com distribuição normal. Portanto, as matrizes de covariância para estes processos aleatórios vetoriais são constantes e diagonais. Assim a distribuição é dada conforme (21) e (22):

$$p(\mathbf{w}_k) = N(0, \mathbf{R}_k) \quad (21)$$

$$p(\mathbf{e}_k) = N(0, \mathbf{Q}_k) \quad (22)$$

Onde \mathbf{R}_k é covariância do ruído de medição e \mathbf{Q}_k a covariância do ruído do processo. Ambas covariâncias podem variar durante o tempo, porém elas serão consideradas constantes para nosso processo. A matriz \mathbf{Q}_k foi definida como uma matriz nula e a matriz \mathbf{R}_k foi definida como sendo uma matriz diagonal, de dimensão 2×2 , onde o primeiro elemento da diagonal é a variância da coordenada horizontal do ponto de início dos segmento de linha que o filtro de Kalman recebeu, considerando todos os pontos obtidos nas imagens passadas da pista. O segundo elemento da matriz diagonal é a variância do ponto de fim do segmento de linha. Neste trabalho, \mathbf{w}_k e \mathbf{e}_k foram definidos com o valor 0.

Depois da etapa de previsão do próximo estado, o filtro realiza a correção conforme é definido por Lim (2012). Nessa etapa, o valor medido é incorporado ao sistema para melhorar o estado a priori, criando um estado aprimorado a posteriori. A diferença entre o valor previsto e o valor medido é utilizada para aperfeiçoar as variáveis do filtro, diminuindo, assim, o erro de predição. Essa diminuição do erro é feita através do ganho de Kalman \mathbf{K}_k ⁴, que gera as estimativas de mínimo erro quadrático (TESCHL, 2012). Este ganho é obtido através da Equação (23), que é calculado a partir da covariância do erro $\mathbf{P}_{k|k-1}$ e da covariância do ruído de medição (\mathbf{R}_k). Esse ganho é aplicado sobre $\mathbf{y}_k - \mathbf{H}_k \mathbf{X}_{k|k-1}$, que é a diferença entre o estado estimado ($\mathbf{H}_k \mathbf{X}_{k|k-1}$) e o medido (\mathbf{y}_k) (obtido através da transformada de Hough e selecionado utilizando o algoritmo MSLC). As equações abaixo mostram a etapa de correção:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k (\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (23)$$

$$\mathbf{X}_{k|k} = \mathbf{X}_{k|k-1} + \mathbf{K}_k (\mathbf{y}_k - \mathbf{H}_k \mathbf{X}_{k|k-1}) \quad (24)$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (25)$$

Onde \mathbf{I} é uma matriz identidade de dimensão 4×4 ; $\mathbf{P}_{k|k-1}$ é uma covariância do erro da estimativa a priori; $\mathbf{P}_{k|k}$ é uma covariância de erro de estimativa posterior; \mathbf{K}_k é o ganho de Kalman (LIM et al., 2012). Na etapa de correção, calcula-se primeiramente o ganho de Kalman \mathbf{K}_k , descrito na Equação (23). Em seguida, mede-se o sistema para obter o seu estado atual \mathbf{y}_k e gerar uma estimativa da posição da linha a posteriori $\mathbf{X}_{k|k}$, incorporando a medição, como mostra a Equação (24). Por fim, se obtém a estimativa a posteriori da covariância do erro através da Equação (25). A cada iteração de cálculo, são utilizados os valores calculados a posteriori para prever os novos dados a priori. Essa propriedade recursiva torna o filtro de fácil implementação (BISHOP; WELCH, 2006; BORKAR; HAYES; SMITH, 2009).

As grandezas de entrada utilizadas no filtro de Kalman são as coordenadas horizontais (eixo x) dos pontos de início e fim de cada segmento de linha extraídos da imagem da pista. O filtro de Kalman é utilizado em cada segmento de linha detectado de forma individual. O vetor de entrada do filtro de Kalman, que contém as coordenadas horizontais de início e fim de um segmento de linha está descrito abaixo:

$$\mathbf{X}_{k|k-1} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (26)$$

Nesse vetor, (x, y) representa os pontos de início e fim de um dos segmentos de linha que descrevem as faixas de sinalização que vão ser corrigidos pelo filtro de Kalman, (\dot{x}, \dot{y}) representam a velocidade desses segmentos de linha. A velocidade inicial foi definida com o valor 0. A cada mudança de *frame*, uma medição ruidosa da posição dos segmentos de linhas das faixas e sinalização é realizada. Supondo que o ruído de medição \mathbf{e}_k também é normalmente distribuído com média zero e distribuição normal, a Equação (27) descreve a medida de entrada do filtro de Kalman considerando o ruído de medição.

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{X}_{k|k-1} + \mathbf{e}_k \quad (27)$$

⁴Também conhecida como equação de Riccati.

Na Equação (27), a matriz \mathbf{H}_k , de dimensão 2×4 , representa a relação entre o estado previsto $\mathbf{X}_{k|k-1}$ e o valor medido \mathbf{y}_k do ponto de vista do comportamento do sistema, ou seja, é o modelo de observação, que mapeia o espaço de estados real no espaço de estados observado. As matrizes \mathbf{I} , \mathbf{H}_k , \mathbf{R}_k e \mathbf{Q}_k utilizadas neste trabalho estão descritas abaixo:

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (29)$$

$$\mathbf{R}_k = \begin{bmatrix} sx_1^2 & 0 \\ 0 & sx_2^2 \end{bmatrix} \quad (30)$$

$$\mathbf{Q}_k = 0 \quad (31)$$

Nessas equações, sx_1^2 é a variância da variável x_1 , que é a coordenada horizontal do início do segmento de linha que está sendo analisado e sx_2^2 é a variância da variável x_2 , que é a coordenada horizontal do fim do segmento de linha. Essas variâncias são calculadas considerando-se as 20 últimas coordenadas de início e fim obtidas referente ao segmento de linhas que está sendo analisado pelo filtro de Kalman, ou seja, cada um dos seis segmentos de linhas vai possuir uma matriz \mathbf{R}_k .

Para prever a posição da linha, foi utilizado um modelo de primeira ordem, que descreve as equações de movimento retilíneo uniformemente variado (MRUV), que vão ser utilizadas para modelar o descolamento dos pontos de início e fim de cada segmento de linha encontrados⁵. As equações do MRUV utilizadas no filtro de Kalman neste trabalho estão descritas a seguir:

$$y = y_0 + \dot{y}t + 0.5adt^2 \quad (32)$$

$$\dot{y} = \dot{y}_0 + at \quad (33)$$

Onde y é a posição, \dot{y} é a velocidade, y_0 é a posição inicial, \dot{y}_0 é a velocidade inicial e a é a aceleração. Como não conhecemos a aceleração do veículo foi atribuído o valor zero para a aceleração a e para a matriz \mathbf{U}_k , caracterizando esse sistema de equações como Movimento Retilíneo Uniforme (MRU). Logo, as equações do modelo matemático utilizado no filtro de Kalman neste trabalho na forma de espaço de estados estão descritas a seguir:

$$\mathbf{F}_{k|k} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (34)$$

$$\mathbf{B}_{k|k} = \begin{bmatrix} 0.5adt^2 \\ 0.5adt^2 \\ dt \\ dt \end{bmatrix} \quad (35)$$

⁵Considerando-se o espaço coordenadas da imagem.

A Figura 24 ilustra o efeito da aplicação do filtro de Kalman em dois dos seis pontos que definem o segmento de linha da direita e da esquerda do sistema implementado neste trabalho. A curva na posição superior de cor vermelha (*line 4*) e a curva da posição inferior na cor azul (*line 3*) representam os pontos de início do segmento de linha da região (a) da faixa da esquerda e da direita obtidas ao longo de 60 primeiros *frames* do banco de dados Cordova 1 da CALTECH utilizado neste trabalho. A curva na posição superior de cor verde (*line 2*) e a curva da posição inferior na cor roxa (*line 1*) representam os mesmos 60 pontos, mas corrigidos pelo filtro de Kalman. É fácil de se observar que o filtro de Kalman propicia uma estabilização dos pontos que descrevem as faixas de sinalização resultando em uma detecção mais suave.

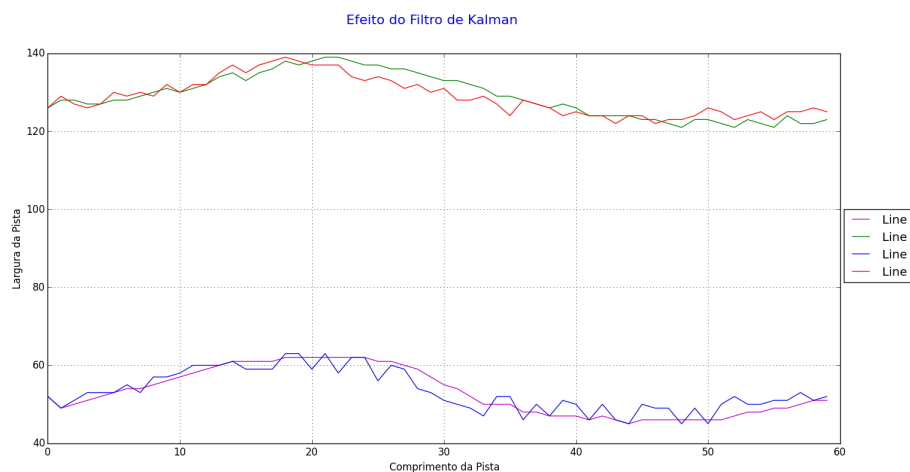


Figura 24: Ilustração do efeito do filtro de Kalman em dois pontos que definem os segmentos de linha mais próximos do veículo de ambas faixas detectadas. As linhas verde (*line 1*) e roxa (*line 2*) corrigidas pelo filtro de Kalman. As linhas azul (*line 3*) e vermelha (*line 4*) são os pontos sem correção.

3.10 Áreas de Busca

Nesta etapa, são delimitadas as áreas onde serão feitas as buscas pelas faixas de sinalização ou bordas da pista na imagem. Após a divisão da imagem nas três regiões é feito o isolamento das áreas de busca na imagem. O sistema utiliza quatro áreas de busca a todo, para realizar a extração das linhas nas imagem. A área de busca 1 é área de busca delimitada a partir da previsão feita pelo filtro de Kalman, a área de busca 2 é obtida partir da interpretação de características extraídas da imagem, as áreas de busca 3 e 4 são regiões da imagem fixas, utilizadas pelo sistema quando nenhum segmento de linha é encontrado nas áreas de busca 1 e 2 e como chute inicial para o Método de Extensão de Linha (MEL). Na região (a) são utilizadas as áreas de busca 1, 3 e 4 e nas regiões (b) e (c) são utilizadas as áreas de busca 1, 2 e 4. A disposição das áreas de buscas relativas a cada região estão ilustradas na Figura 25.

A área de busca 1 é obtida através das previsões feitas pelo filtro de Kalman implementado. O emprego do filtro de Kalman neste trabalho só foi possível porque o deslocamento das faixas de sinalização da pista na imagem é gradual e linear, ou seja, a posição das faixas de sinalização ou bordas da pista na imagem não mudam de forma abrupta.

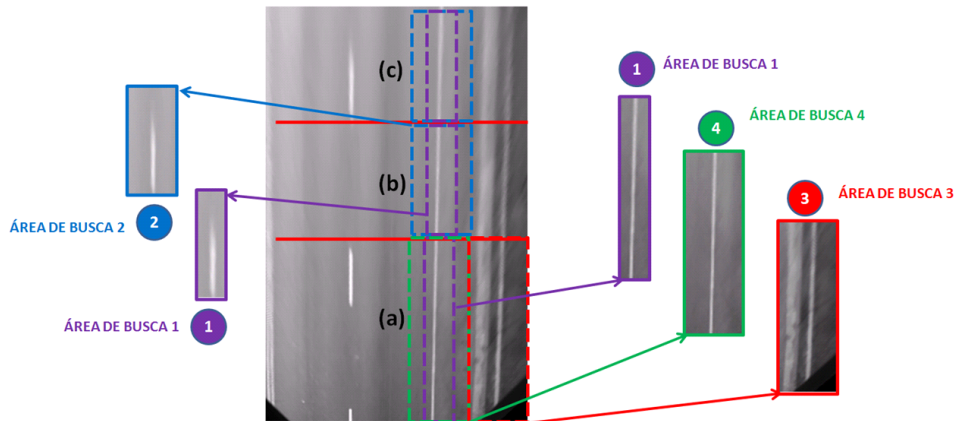


Figura 25: Ilustração das áreas de buscas utilizadas pelo sistema para extrair os segmentos de linha que descrevem as faixas de sinalização.

No início de cada iteração do sistema todas as seis áreas de busca, fornecidas pelo filtro de Kalman, já estão definidas. Porém, o MEL sempre é executado em paralelo ao filtro de Kalman, para garantir pelo menos uma área de busca para cada segmento de linha, caso não tenha sido possível definir a área de busca 1 de algum segmento. A dinâmica do sistema utilizando as áreas fornecidas pelo filtro de Kalman está ilustrada na Figura 26, onde a sequência definida pelos círculos verdes fazem referência a detecção da faixa da direita e as em azul a faixa da esquerda. Nessa imagem, primeiramente em (1) o sistema isola a área de busca 1 (retângulo amarelo tracejado) na região (a), e um filtro de limiar adaptativo é utilizado para binarizar a imagem em (2), seguido da transformada de Hough e do algoritmo MSLC para encontrar as linhas contidas na área em (3). O mesmo procedimento é aplicado em (4) e (5) para encontrar os demais segmentos de linha das regiões (b) e (c).

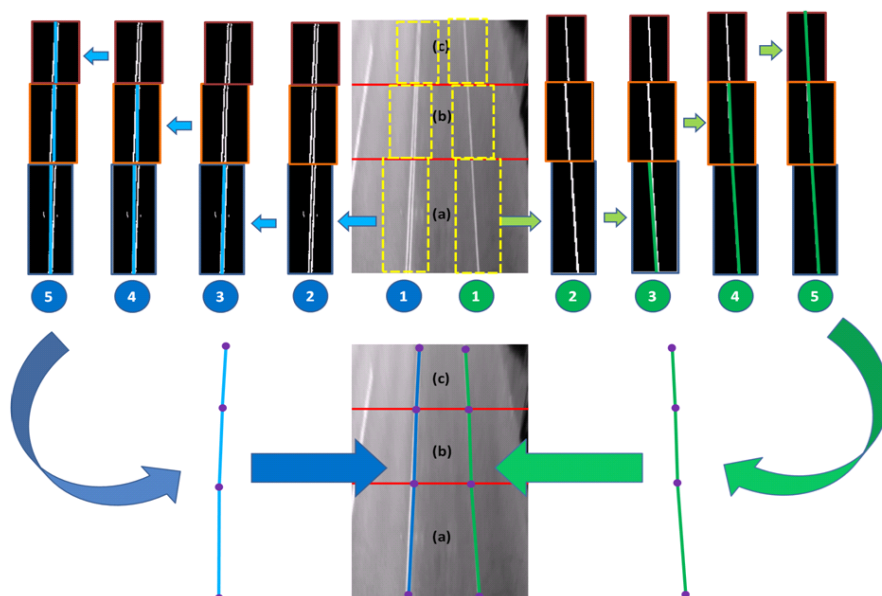


Figura 26: Ilustração do processo de extração dos segmentos de linha utilizando as áreas de busca obtidas através das previsões feitas pelo filtro de Kalman.

Quando o veículo começar a mudar de pista, eventualmente a faixa referente ao lado oposto da pista para qual o veículo está se deslocando vai sair da área de visão do sistema e não vai mais ser detectada. Nesse situação, não haverá áreas de busca 1 fornecidas pelo filtro de Kalman referentes a faixa que desapareceu do campo de visão e o sistema vai utilizar somente as áreas de busca 2 pra extrair as faixas da imagem.

A área de busca 2 é obtida através do MEL implementado. Este método é utilizado principalmente como apoio ao filtro de Kalman que gera as área de busca 1, em situações onde filtro de Kalman não conseguiu fornecer previsões referentes a algum dos seis segmentos de linha ou não tenha sido possível extrair nenhum segmento de linha da área de busca 1 fornecida. Todas as áreas de busca fornecidas pelo MEL são utilizadas na primeira vez que o sistema é acionado, pois não podemos utilizar o filtro de Kalman por não termos nenhuma informação prévia da pista.

A dinâmica de funcionamento do MEL está ilustrado na Figura 27, onde a sequência definida pelos círculos verdes fazem referência a detecção da faixa da direita e as em azul a faixa da esquerda. Inicialmente, na etapa (1) o sistema faz um chute inicial da posição do primeiro segmento de linha na região (a), esse chute inicial está indicado pelo retângulo tracejado amarelo. Em (2) o filtro de limiar adaptativo é aplicado na área dentro do retângulo tracejado amarelo, seguido da etapa (3), onde a transformada de Hough e o algoritmo MLSC são utilizados para encontrar a linha que descreve o segmento da marcação na pista. Ainda em (3), estendemos a linha encontrada na pequena região (a) ao longo da região (b). Em (4) criamos outro retângulo tracejado (representado na cor laranja) ao redor desta linha estendida, com largura fixa de 20 píxeis. Na etapa (5) utilizamos novamente o filtro limiar adaptativo, transformada de Hough e MSLC sobre essa nova área, para extraímos a linha que descreve a marcação na pista. Em (6) estendemos a linha encontrada na região (b) repetimos o processo para extrair a linha que descrevem a marcação na região (c). Finalmente, temos três segmentos de linha que descrevem a marcação da faixa da esquerda na estrada. Este mesmo processo é realizado para encontrar a linha da direita.

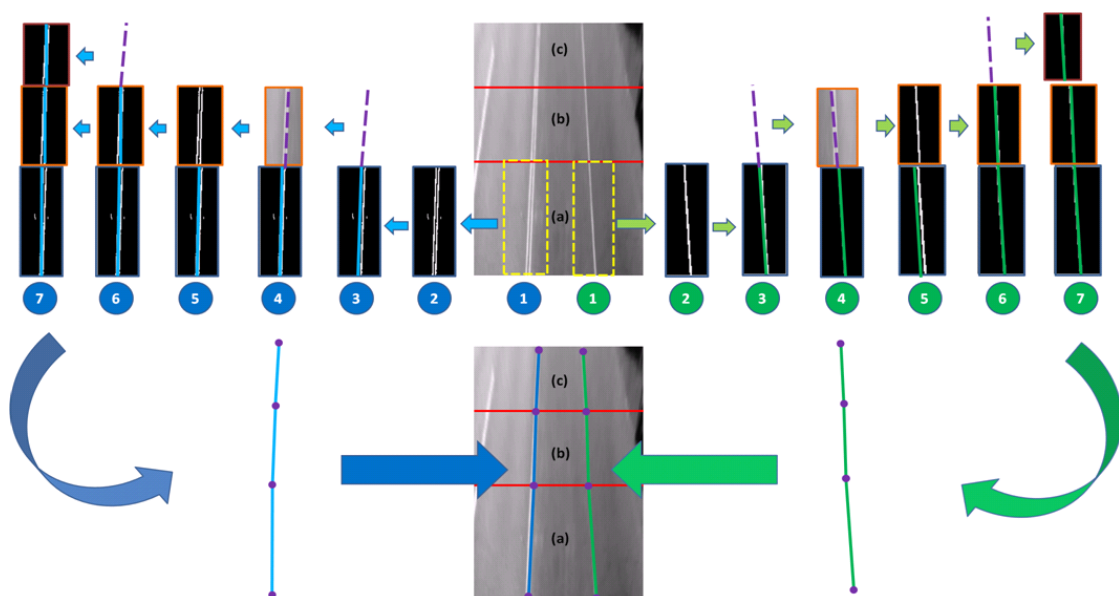


Figura 27: Ilustração do processo de extração dos segmentos de linha utilizando as área de busca obtidas pelo MEL.

O chute inicial utilizado pelo MEL é dado a partir das áreas de busca 3 e 4. A área de busca 4 foi definida conforme a ilustração 1 da Figura 28. Ela foi definida com base na premissa de que o motorista sempre tende a posicionar o veículo na parte central da pista. Para o banco de dados da CALTECH, os limites que definem a área de busca 4 para a faixa da esquerda foram definidos no intervalo dos píxeis 35 a 75 do eixo horizontal e os referentes a faixa da direita foram definidos no intervalo de 105 a 145. Para o banco de dados da UFRGS, os limites que definem a área de busca 4 para a faixa da esquerda foram definidos no intervalo dos píxeis 40 a 80 do eixo horizontal e os referentes a faixa da direita foram definidos no intervalo de 90 a 130. Quando o veículo está passando de uma pista sem faixas de sinalização para uma pista com faixas de sinalização, utilizamos a área de busca 4 nas regiões (b) e (c) na imagem para extrairmos os segmentos de linha que descrevem as faixas de sinalização até que as faixas atingiam a região (a).

A área de busca 3 foi definida conforme a ilustração 2 da Figura 28. Ela somente é utilizada na região (a) como chute inicial para o MEL, pois é a área utilizada para realizar as buscas pelas bordas da pista quando o sistema não encontra nenhuma faixa de sinalização pintada na pista. Para o banco de dados da CALTECH, os limites que definem a área de busca 3 para a faixa da esquerda foram definidos no intervalo dos píxeis 0 a 40 do eixo horizontal e os referentes a faixa da direita foram definidos no intervalo de 140 a 180. Para o banco de dados da UFRGS, os limites que definem a área de busca 3 para a faixa da esquerda foram definidos no intervalo dos píxeis 0 a 45 do eixo horizontal e os referentes a faixa da direita foram definidos no intervalo de 130 a 190.

A posição das áreas 3 e 4 muda de acordo com a posição da câmera no veículo, com os pontos utilizados para realizar a transformação de perspectiva e de acordo com a posição das faixas regulamentada pelas leis de cada país.

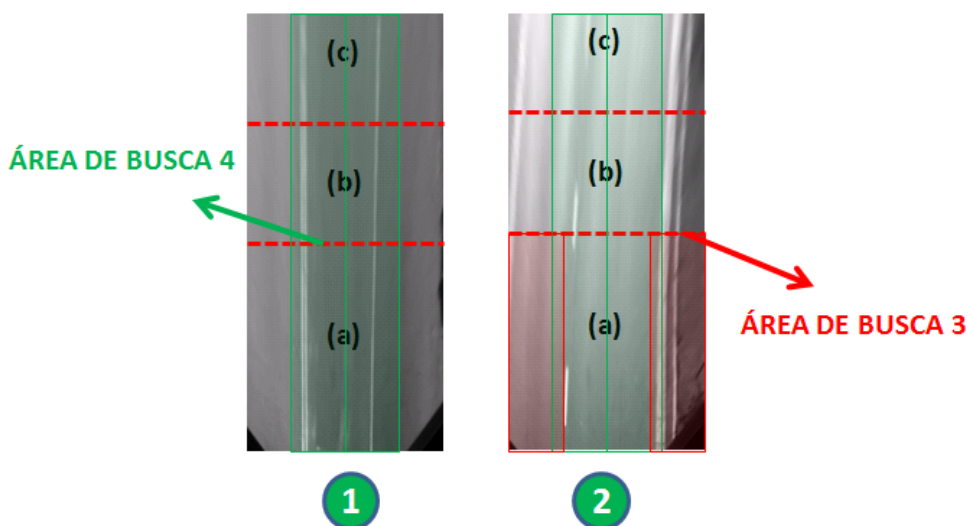


Figura 28: Ilustração das regiões de busca fixa 3 e 4. A área de busca 3 está delimitada pelos retângulos na cor vermelha e área de busca 4 pelos retângulos na cor verde.

3.11 Correção de Erros

Na etapa de Correção de Erro é feita uma análise das linhas encontradas com o objetivo de identificar as detecções falso-positivas e completar as lacunas entre as linhas. O

primeiro problema que tentamos eliminar é uma consequência devido à divisão da imagem em três regiões (Regiões (a), (b) e (c)). Em certos trechos da pista o segmento de linha da região (b) não é preenchido, pois pode não haver uma marcação contida dentro dessa região. Isto acontece principalmente quando as faixas de sinalização na pista estão na forma tracejada. Para contornar esse problema e fornecer ao condutor do veículo ou sistema de guia autônomo uma informação mais concreta e homogênea, isolamos os pontos extremos dos segmentos de linha das regiões (a) e (c) e criamos uma linha que liga esses dois pontos. Esse processo é ilustrado pela Figura 29.

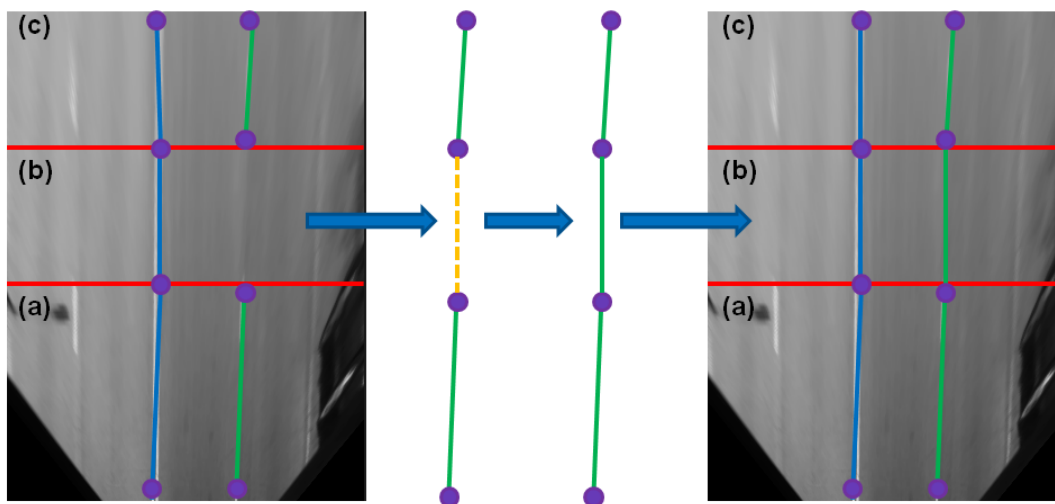


Figura 29: Ilustração do procedimento de preenchimento de lacunas na detecção quando as linhas na pista estão na forma tracejada.

Dois dos casos de detecção de falso-positivos mais comuns é quando um outro veículo entra sobre a área de detecção do sistema ou quando outras formas de sinalização horizontal estão presentes na pista, como palavras de PARE e setas de sentido de conversão, que acabam sendo detectadas pelo sistema erroneamente, conforme ilustra a Figura 30.

Para tentar eliminar esses problemas foi definido um limiar de distância mínima entre as linhas e foi criada uma faixa de ângulos que os segmentos de linha encontrados devem apresentar. Esses valores foram definidos experimentalmente e são relacionados à posição da câmera instalada no veículo e dos pontos utilizados para realizar a transformação de perspectiva da imagem. Outro fator que pode ser relativo é a legislação de cada país referente a posição das faixas de sinalização na pista.

Para o banco de dados da CALTECH o limite mínimo foi definido como 30 píxeis e o limite máximo como sendo de 135 píxeis. Para o banco de dados da UFRGS o limite mínimo foi definido como 20 píxeis e o limite máximo como sendo de 90 píxeis. As faixas de ângulos de 40 graus até 95 graus e de 220 graus até 310 graus foram definidas igualmente para ambos os bancos de dados. Novamente, esses valores são relativos a posição da câmera do veículo e aos pontos utilizados para realizar a transformação de perspectiva.

Outra problema bem comum é o comprimento variado dos segmentos de linha detectados. Para garantir que o método utilizado para definir as áreas de busca e o filtro de Kalman possam fornecer resultados mais precisos, convertemos o comprimento de todas as linhas para comprimentos padrões, ou seja, alongamos todos os segmentos de linha até encontrar as linhas horizontais vermelhas. Esses comprimentos são iguais as alturas

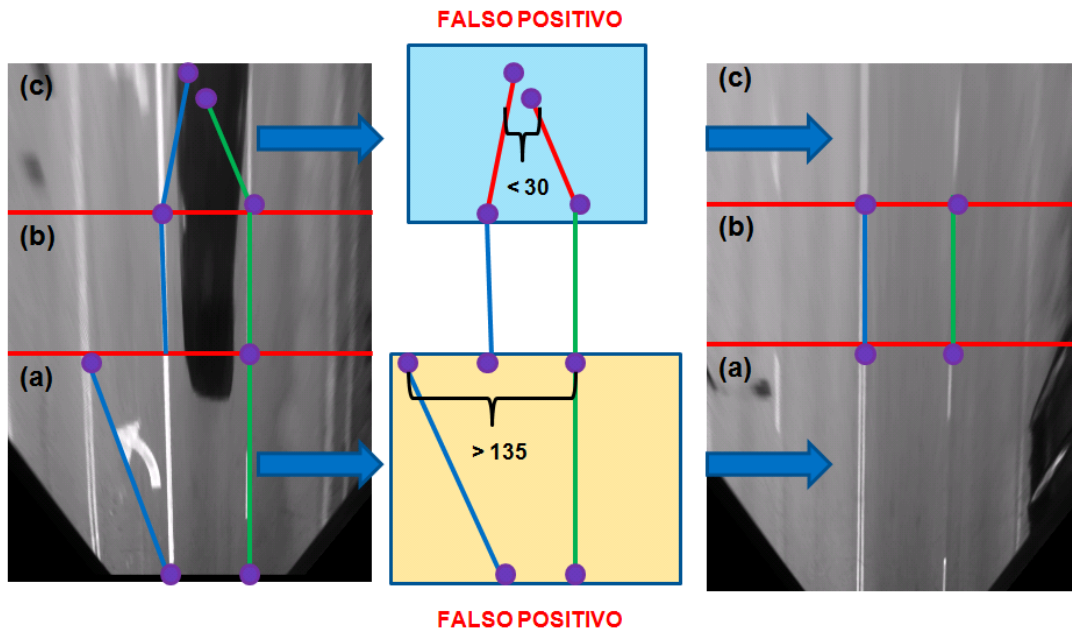


Figura 30: Ilustração do procedimento de detecção de linhas falso-positivas devido a interferência de veículos e ruído na pista.

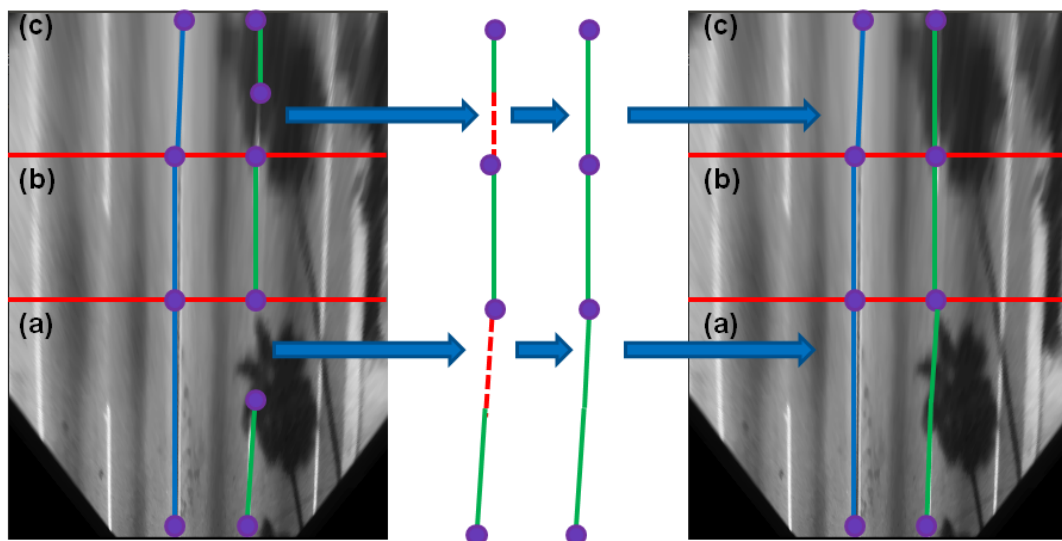


Figura 31: Ilustração do procedimento de correção do comprimento das linhas detectadas.

das regiões (a), (b) e (c). Dessa forma todas as alturas são conhecidas, e nosso sistema trabalha somente com as informações de deslocamento longitudinal (variação ao longo do eixo x da imagem). A Figura 31 ilustra este processo.

3.12 Metodologia

O algoritmo implementado nesse trabalho está ilustrado de forma geral na Figura 32. Ele é dividido em quinze etapas, classificadas em três grupos: Pré-Processamento, Extração de Característica e Interpretação.

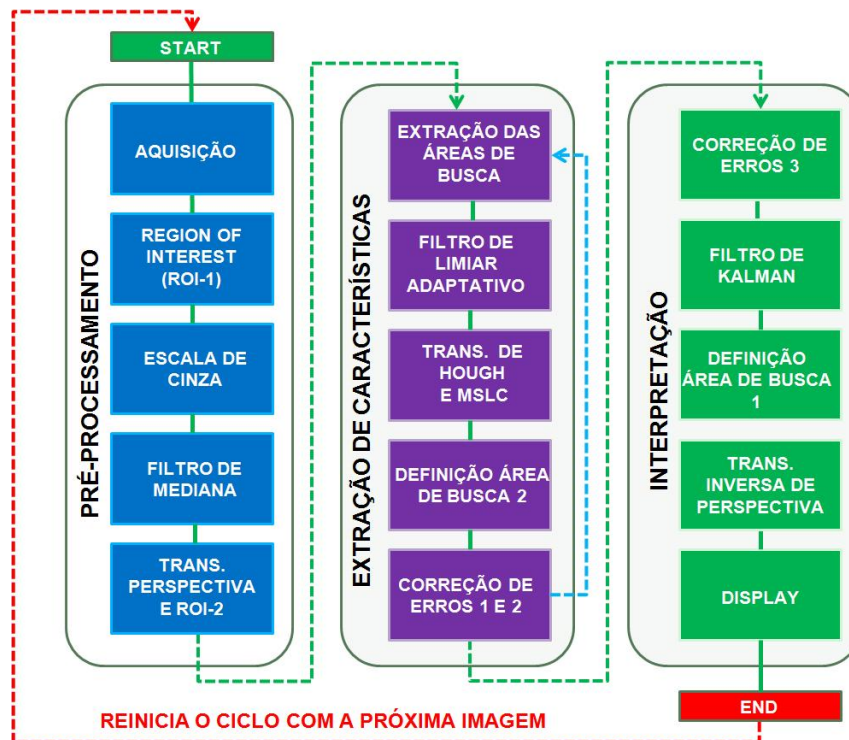


Figura 32: Ilustração das doze etapas do sistema de detecção de faixas de sinalização desenvolvido neste trabalho.

Na etapa de Aquisição, as imagens da pista são obtidas e armazenadas na memória do sistema. Foram utilizados dois bancos de dados para validação. As imagens do banco de dado da CALTECH foram obtidas a partir de duas câmeras fixadas na parte superior-frontal do veículo, nas extremidades do teto e para o banco de dados da UFRGS a partir de uma câmera, instalada na parte superior-frontal do veículo, um pouco a cima do para-brisas. Na segunda etapa, identificamos e extraímos a região de interesse 1 (ROI-1) da imagem adquirida. Nessa região estão contidas as informações relevantes para as demais etapas do sistema, ou seja, a região na imagem onde a pista está contida é a nossa região de interesse. Essa região é delimitada por dois limites verticais, o limite inferior é o ponto vertical da imagem, definido de baixo para cima, onde a carcaça do veículo não causa nenhuma oclusão da vista da pista e o limite superior, definido de cima para baixo, é o ponto de fuga da imagem ou *vanishing point*.

Na terceira etapa, transformamos o espaço de cores da imagem para o escala de cinza, uma vez que a informação referente as cores presentes na imagem não é utilizada em nenhuma etapa do sistema. Em seguida, aplicamos um filtro de mediana para suavizar a estrutura dos objetos presentes na imagem. O filtro de mediana é utilizado para remover pequenos detalhes da imagem antes da extração de estruturas e para conexão de pequenas discontinuidades em linhas ou curvas. Ele foi utilizado neste trabalho para homogenizar as estruturas das faixas de sinalização da pista e tornar sua extração mais eficiente.

Na quinta etapa é realizada a transformação de perspectiva da imagem para a vista *Bird's Eye*. Para realizar essa transformação foi utilizado uma Transformada Homográfica de Perspectiva, que realiza uma transformação linear de raio equivalente entre dois planos euclidianos distintos, através do mapeamento dos píxeis da imagem por um operador linear. Na transformação homográfica não é necessário a calibração da câmera,

somente a definição de oito pontos espaciais, quatro deles em cada plano que se deseja realizar a transformação de perspectiva. A imagem da pista nessa perspectiva nos permitirá extrair as características da estrada com mais precisão e confiabilidade, principalmente na detecção da curvatura da pista e das faixas de sinalização que estão mais distantes do veículo. Após a mudança de perspectiva, uma nova região de interesse é isolada na imagem (ROI-2), e na sequência ela é dividida em três regiões menores (regiões (a), (b) e (c)) e cada região é analisada individualmente.

Na sexta etapa (primeira etapa do grupo de extração de características), duas áreas de busca são extraídas em cada uma das três regiões. A contribuição deste trabalho está centrada na utilização destas áreas de busca, que foram desenvolvidas com o objetivo de isolar as regiões da imagem, em pequenas áreas, onde as faixas estão contidas e aplicar as técnicas de extração de linhas somente nestas pequenas regiões. Uma vez que o custo computacional de se utilizar essas técnicas é diretamente proporcional ao tamanho das imagens onde elas são aplicadas, vamos obter uma redução do custo computacional de uma forma bem significativa utilizando essas áreas de busca otimizadas.

As áreas de busca são definidas de três maneiras: Através de um filtro de Kalman (etapa 12), usamos as informações da posição das faixas de sinalização obtidas nas imagens passadas da pista para prever onde elas estarão na imagem atual e as usamos para definir as seis áreas de busca.

A segunda maneira é o Método de Extensão de Linha (MEL) que é utilizado quando não foram encontradas nenhuma marcação da pista nas áreas de busca fornecidas pelo filtro de Kalman. Neste método, são utilizadas as informações presentes na imagem atual para delimitar as regiões onde serão realizadas as buscas pelas faixas de sinalização, isto é, são utilizadas as linhas encontradas na região (a), para prever as áreas de busca na região (b) e são utilizadas as linhas encontradas na região (b) para prever as áreas de busca na região (c).

O terceiro método utiliza regiões de buscas fixas na imagem, que são regiões onde as faixas de sinalização convergem em função da tendência do motorista do veículo de sempre buscar a região central da via destinada a ele e regiões onde as bordas da pista provavelmente estão. Essas regiões fixas só são utilizadas em último caso, quando não for possível extrair nenhum segmento de linha nas outras regiões de busca.

Para realizar a extração das linhas que descrevem as faixas de sinalização nas áreas de busca na pista aplicamos três procedimentos, que correspondem as etapas 7, 8 e 9. No primeiro, binarizamos a imagem utilizando um filtro de limiar adaptativo (etapa 7), seguido da aplicação da transformada de Hough (etapa 8) para extrair as linhas contidas na imagem. Quase sempre a transformada de Hough irá encontrar mais de uma linha na área de busca. Dessa forma, para escolhermos qual dessas linhas é a mais promissora, utilizamos o Método de Seleção de Linha Candida (MSLC) que faz essa seleção com base na concentração e posição dos píxeis na imagem. Por fim, definimos a área de busca 2, através do MEL, para o próximo segmento de linha que vamos detectar.

Uma vez que as linhas que descrevem as faixas de sinalização na pista foram obtidas, elas são analisadas na etapa 10 para buscar erros, ajustar o comprimento dos segmentos de linha para que tenham sempre o mesmo comprimento e detectar falso-positivos. As etapas do grupo de Extração de Características são executadas seis vezes, uma para cada segmento de linha que descrevem as duas faixas de sinalização que desejamos detectar.

Na etapa 11 (início do grupo de Interpretação) realizamos um último procedimento de correção de erros para tentar eliminar os buracos na nossa detecção, caso algum dos segmentos de linha da região (b) não tenha sido detectado, completamos as lacunas através

de uma ligação simples, entre o final do segmento de linha da região (a) e o início do segmentos de linha da região (c).

Na etapa 12 aplicamos o filtro de Kalman sobre as linhas encontradas, com o objetivo de reduzir o efeito do ruído e da deformação da estrutura da linha causada no processo de limiarização, de forma a garantir a estabilidade da linha apresentada ao condutor do veículo ou sistema autônomo. A segunda utilidade do filtro de Kalman é fornecer as áreas de busca que serão utilizadas para extrair as faixas na próxima imagem da pista. Essas áreas são definidas utilizando as previsões que o filtro de Kalman faz da posição dos seis segmentos de linha na próxima imagem da pista (etapa 13).

Por fim, na etapa 14, os segmentos de linha extraídos da imagem na pista na vista *Bird's Eye* são trazidas de volta para o plano tridimensional, utilizando a inversa do operador \mathbf{H} e são apresentadas na tela para o usuário (etapa 15), ou a posição das linhas são informadas para o sistema autônomo. Essas linhas podem ser utilizadas para monitorar o comportamento do veículo sobre a pista e gerar alertas,0 caso o veículo saía da área de tráfego seguro destinado a ele, ao mesmo tempo que pode ser utilizado para refinar a trajetória de um veículo autônomo na pista.

4 IMPLEMENTAÇÃO

O sistema desenvolvido neste trabalho foi implementado utilizando a linguagem de programação de alto nível Python. A linguagem Python foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional, isto é, ela prioriza produtividade e legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara e sua principal característica é permitir a fácil leitura do código e compactação das linhas de código em comparação a outras linguagens. Embora a linguagem Python não seja a linguagem mais indicada para sistemas que operam em tempo real, por ser uma linguagem lenta em termos de execução, ela foi escolhida para o desenvolvimento desse trabalho por proporcionar uma forma mais fácil e dinâmica de manipulação de imagens do que outras linguagens. A principal biblioteca utilizada para o desenvolvimento do sistema foi o OpenCV (*Open Source Computer Vision Library*).

O OpenCV é uma biblioteca de software de visão computacional e aprendizagem de máquinas de código aberto. Ele foi desenvolvido para fornecer uma infra-estrutura comum para aplicações de visão computacional e para acelerar o uso da percepção da máquina nos produtos comerciais. Possui interfaces C++, C, Python, Java e MATLAB e suporta Windows, Linux, Android e Mac OS. O OpenCV é destinado principalmente para aplicações de visão em tempo real (OPENCV, 2016).

As aplicações do OpenCV abrangem desde a agrupamento de imagens do streetview, detecção intrusos em vídeos de vigilância em Israel, monitorando equipamentos de minas na China, ajudando robôs a navegar e pegar objetos, detecção de acidentes em piscinas (OPENCV, 2016). O trabalho foi inteiramente desenvolvido utilizando IDE Pycharm da JetBrains. Essa IDE fornece uma estratégia inteligente de compactação e alinhamento de códigos de programação, revisão do código, realce e ferramentas de manipulação dinâmicas ao mesmo tempo que fornece recursos de navegação rápidos.

4.1 Sistema

O diagrama de funcionamento geral do sistema e suas respectivas divisões foram implementadas tal como apresentado na Figura 32 do capítulo Metodologia. A Figura 33 ilustra de forma geral como o sistema foi implementado, suas componentes, ordenamento e interações. O funcionamento e sequência de execução do diagrama vão ser contemplados nas próximas seções.

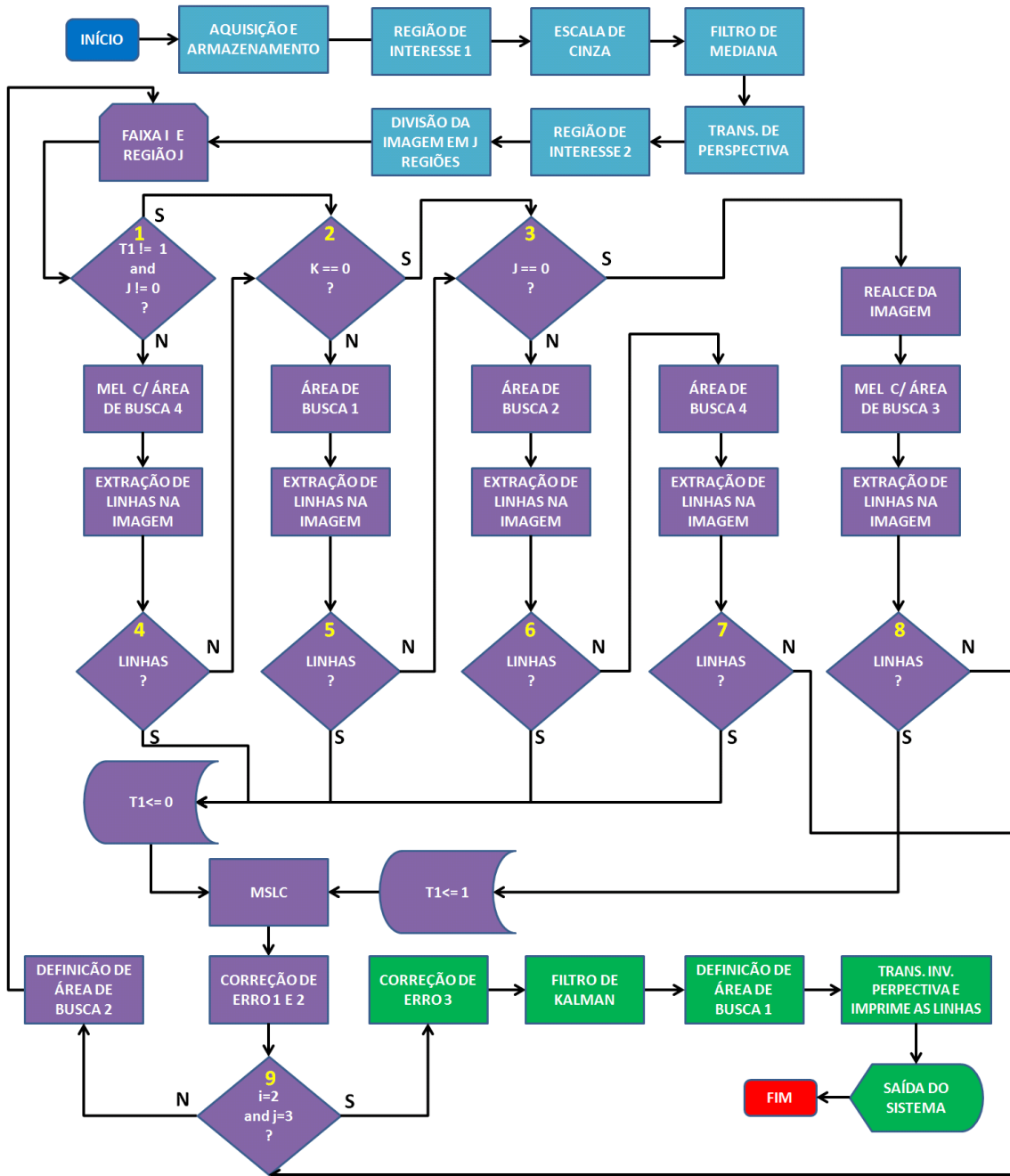


Figura 33: Ilustração do sistema implementado, suas componentes, ordenamento e interações.

4.1.1 Inicialização

Na inicialização do sistema todas as variáveis do sistema são inicializadas. Os seis segmentos de linha são inicializados com o valor 0. Os demais parâmetros de inicialização foram definidos conforme os valores descritos dos capítulos de descrição dos métodos utilizados no sistema, descritos nas seções anteriores. O operador linear utilizado para fazer a transformação de perspectiva já está definido quando o sistema é inicializado e seus valores não são modificados.

Quando o sistema é ligado, ele não possui nenhuma informação sobre a estrada, isto

é, ele não possui nenhuma informação referente a posição das faixas de sinalização da pista das imagens passadas. Sem essa informação não podemos fazer nenhuma suposição confiável sobre todas as áreas onde as marcações da pista estarão na imagem atual da estrada. Logo, ele não vai utilizar o filtro de Kalman, pois não possuímos informações prévias da pista. Nessa situação, somente o MEL é utilizado para extrair os seis segmentos de linha que descrevem as faixas de sinalização. As linhas detectadas vão ser utilizadas no filtro de Kalman na imagens futuras da pista.

Caso o sistema esteja sendo executado em um computador a aquisição da imagem da pista vai ser realizada utilizando imagens contidas em um diretório no computador. Caso o sistema esteja embarcado em alguma plataforma de hardware as imagens vão ser obtidas a partir de uma câmera instalada no veículo e ligada ao hardware como periférico.

4.1.2 Pré-Processamento

A etapa de Pré-Processamento executada pelo sistema é constituída por sete passos que vão preparar a imagem para o procedimento de extração dos segmentos de linha. Esta etapa e suas componentes estão ilustrada na Figura 34.

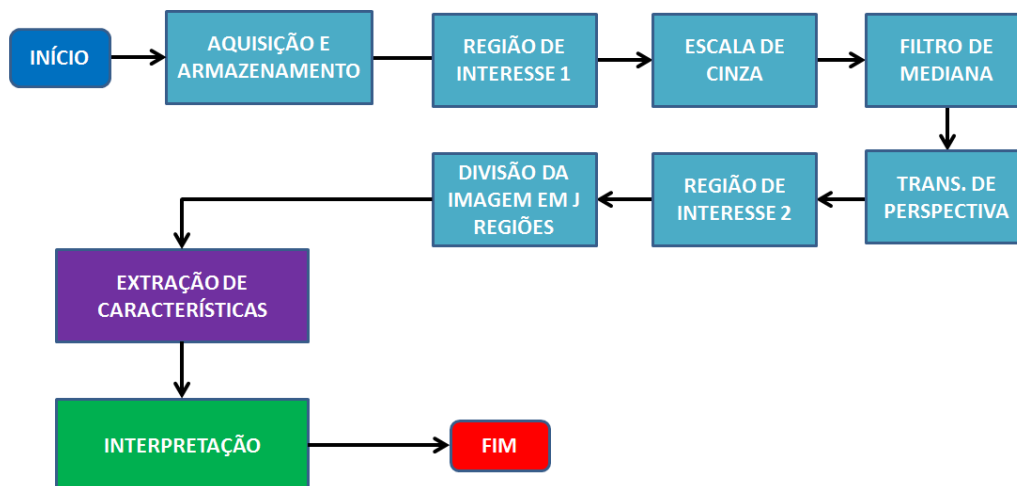


Figura 34: Ilustração das etapas de pré-processamento executados pelo sistema.

O primeiro procedimento realizado pelo sistema é aquisição da imagem da pista. Conforme foi apresentado no subcapítulo 3.2, as imagens são obtidas através de uma câmera instalada na parte superior-frontal do veículo. A Figura 35 ilustra o procedimento de aquisição da imagem da pista idealizada neste trabalho. Onde (1) mostra uma imagem do veículo autônomo Alice da CALTECH (ALY et al., 2007). Em (2) as câmeras estéreo do veículo capturam a imagem da pista. Em (3) a imagem é digitalizada e armazenada em uma unidade de memória seguido da etapa (4) onde a imagem é enviada para o sistema de detecção de faixas de sinalização para ser processada.

O desenvolvimento do sistema implementado neste trabalho foi realizado utilizando imagens de um banco de dados obtidas através do veículo autônomo Alice do Instituto de Tecnologia da Califórnia (CALTECH) (ALY et al., 2007). As imagens são carregadas e armazenadas individualmente simulando em tempo real a aquisição feita pela câmera do veículo.

Na segunda etapa, identificamos e extraímos a primeira região de interesse da imagem armazenada. Essa região vai conter somente as informações relevantes para as demais

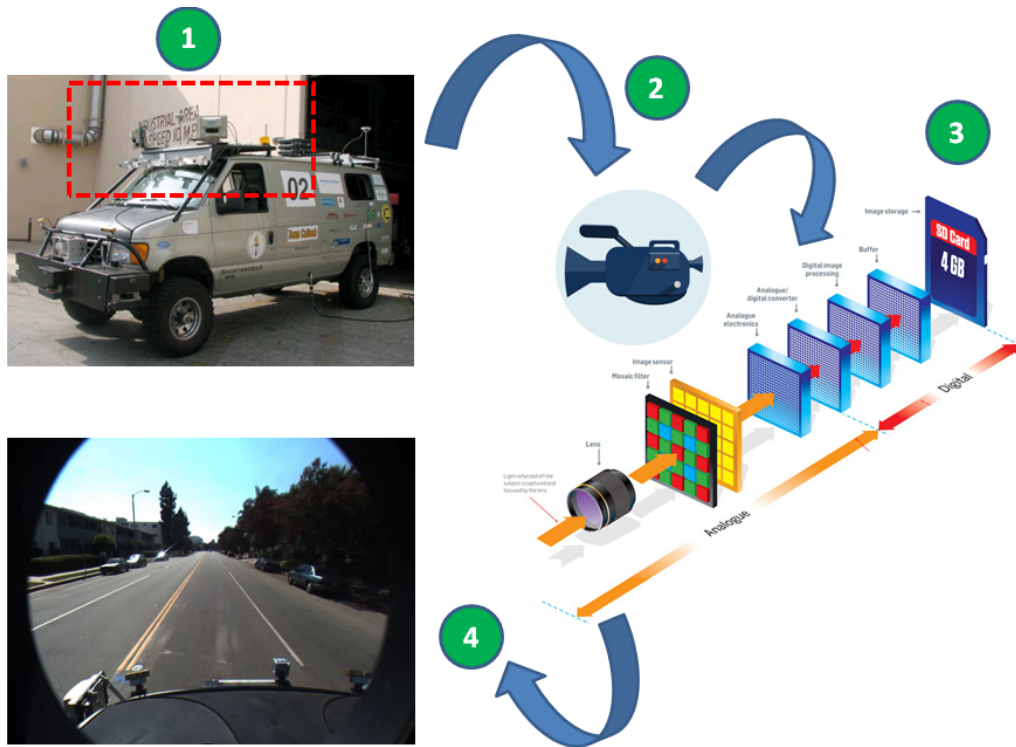


Figura 35: Ilustração do procedimento de aquisição da imagem da pista utilizando as câmeras do veículo autônomo Alice.

etapas do sistema, ou seja, a região da imagem onde a pista está contida é a nossa região de interesse. Esse procedimento está ilustrado na Figura 36. Os limites de corte superior e inferior da região de interesse 1 foram definidos como sendo o ponto de fuga da imagem (*Vanishing Point*) e como o ponto onde a parte frontal do veículo não causa nenhuma oclusão na vista da pista, conforme discutido no subcapítulo 3.3.



Figura 36: Ilustração do procedimento separação da imagem de interesse 1 (ROI-1). A imagem (1) representa a imagem da pista obtida pelas câmeras, a imagem (2) delimita a região de interesse 1 e a imagem (3) representa a região de interesse extraída.

Na terceira etapa, transformamos o espaço de cores da imagem para o escala de cinza, uma vez que a informação referente as cores presentes na imagem não é utilizada em nenhuma etapa do sistema. Em seguida, aplicamos um filtro de mediana para remover os ruídos presentes na imagem para tornar o processo de extração de informação relevante

mais eficaz, conforme discutido no subcapítulo 3.4. A Figura 37 ilustra as etapas de processo. Onde a imagem (1) ilustra a imagem colorida da região de interesse, (2) representa a imagem após o procedimento de conversão para a escala de cinza, e por fim, a imagem (3) representa a imagem após a filtragem pelo filtro de mediana. As funções de conversão de escala RGB para escala de cinza e filtro de mediana utilizado neste trabalho foram obtidas através da biblioteca OpenCV.

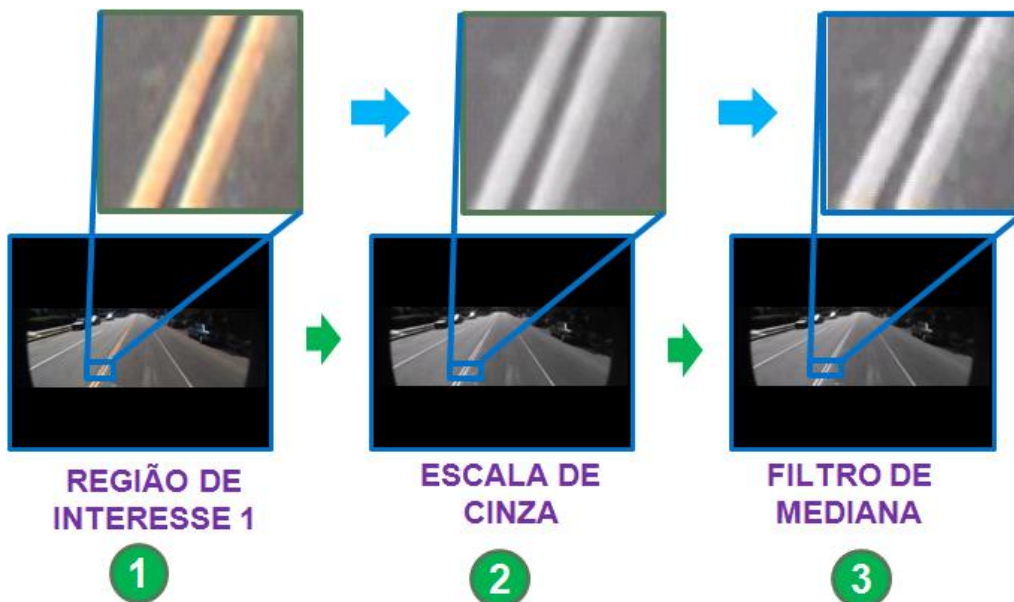


Figura 37: Ilustração das etapas de conversão da escala de cores e da aplicação do filtro de mediana.

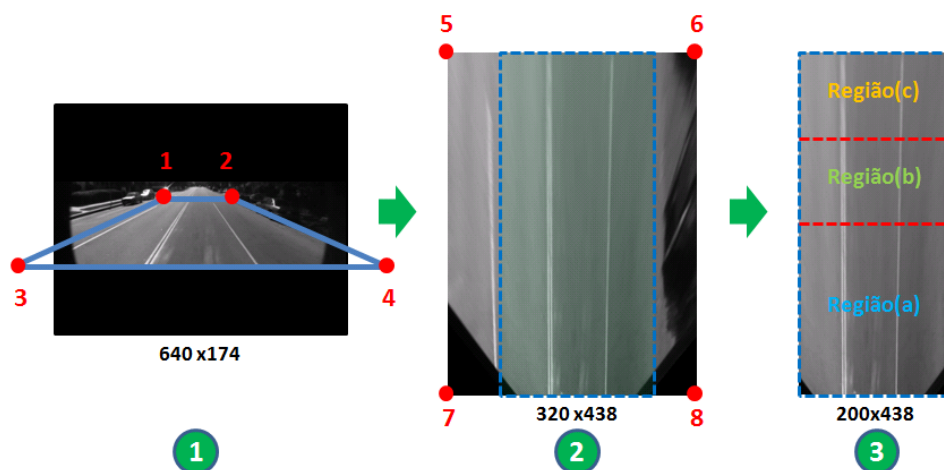


Figura 38: Ilustração das etapas de transformação de perspectiva e extração da segunda região de interesse.

Após realizar as etapas de pré-processamento da imagem o sistema executa a operação de transformação de perspectiva da imagem da pista, para obter a vista *Bird's Eye*. Esse procedimento é realizado através da multiplicação da matriz homográfica¹ por toda

¹O procedimento para se obter a matriz homográfica é descrito em detalhes no subcapítulo 3.5.

a imagem, que vai agir como um operador linear e remapear os píxeis da imagem em um outro espaço dimensional. A imagem na perspectiva tridimensional com dimensões 640×174 é mapeada em uma imagem com dimensões 320×438 . Em seguida, isolamos a segunda região de interesse (ROI-2), que é a região onde somente as faixas de sinalização da pista destinadas ao veículo estão contidas, conforme ilustra a Figura 38. Nessa imagem, (1) ilustra a imagem da pista após a etapa de pré-processamento e os quatro primeiros pontos utilizados para se obter a matriz de transformação homográfica, a imagem (2) representa a imagem após a transformação de perspectiva e os quatros últimos pontos utilizados para se obter a matriz homográfica. Por fim, (3) ilustra a imagem após a extração da região de interesse 2 e as respectivas delimitações das três regiões que o sistema divide a imagem. O resultado final dessa procedimento gera uma imagem com dimensões 200×438 na escala de cinza. Após a mudança de perspectiva a imagem é então dividida em três regiões (regiões (a), (b) e (c)) e cada região é analisada individualmente.

4.1.3 Extração de características

Na etapa seguinte, o sistema executa um *loop* para extrair os seis segmentos de linha que descrevem as faixas de sinalização ou bordas da pista nas três regiões que a imagem foi dividida. A ordem de execução do Loop começa com o segmento de linha que descreve a faixa da esquerda da região (a), seguido do segmento de linha da esquerda da região (b), e por fim, da região (c). Em seguida, a mesma ordem de execução é feita nos segmentos de linha da direita. No diagrama, essas seis iterações são comandadas pela variável "i", de tamanho 2, que indica se o sistema esta analisando a faixa da esquerda ou da direita e pela variável "j", que indica se o sistema esta analisando os segmentos de linha da região (a), (b) ou (c).

A sequência de execução inicial do sistema, indicado pelo caminho tracejado na cor verde e pelos balões laranjas, é ilustrado na Figura 39. Inicialmente a variável T1 é testada para determinar se o sistema está rastreando uma faixa de sinalização ou a borda da pista.

Quando T1=0, significa que o sistema está rastreando uma faixa de sinalização e deve passar para o segundo teste (indicado pelo caminho de bolinhas vermelhas). Caso T1=1, significa que o sistema está rastreando a borda da pista e deve utilizar a área de busca 4 como chute inicial para o Método de Extensão de Linha (MEL), apresentado no subcapítulo 3.10, para definir as demais áreas de busca. ²

A extração das linhas da imagem é um procedimento onde a imagem é limiarizada por um filtro de limiar adaptativo apresentado no subcapítulo 3.6, seguido da aplicação da transformada de Hough que foi apresentada no subcapítulo 3.7, para encontrar as linhas presentes na área que está sendo analisada pelo sistema. Se nenhum segmento de linha for encontrado o sistema passa para o segundo teste (indicado pelo caminho traço e ponto na cor azul). Caso contrário, os segmentos de linha encontrados pelo procedimento de extração de linhas, são enviados para o algoritmo MSLC para escolher qual das linhas extraídas vai ser utilizada.

O algoritmo MSLC (apresentado no subcapítulo 3.8), analisa a distribuição dos píxeis presentes nas linhas encontradas pela transformada de Hough e determinar qual, dentre essas linhas, é a mais promissora. A Figura 40 mostra a entrada e a saída do algoritmo

²O sistema proposto nesse trabalho foi desenvolvido dando preferência a detecção das faixas de sinalização pintadas na pista, somente na ausência delas o sistema deve utilizar a borda da pista pra indicação da área de tráfego do veículo. Com esse critério de prioridade, o sistema sempre busca primeiramente por uma faixa de sinalização, mesmo quando estiver rastreando a borda da pista e por este motivo sempre é feito este primeiro teste.

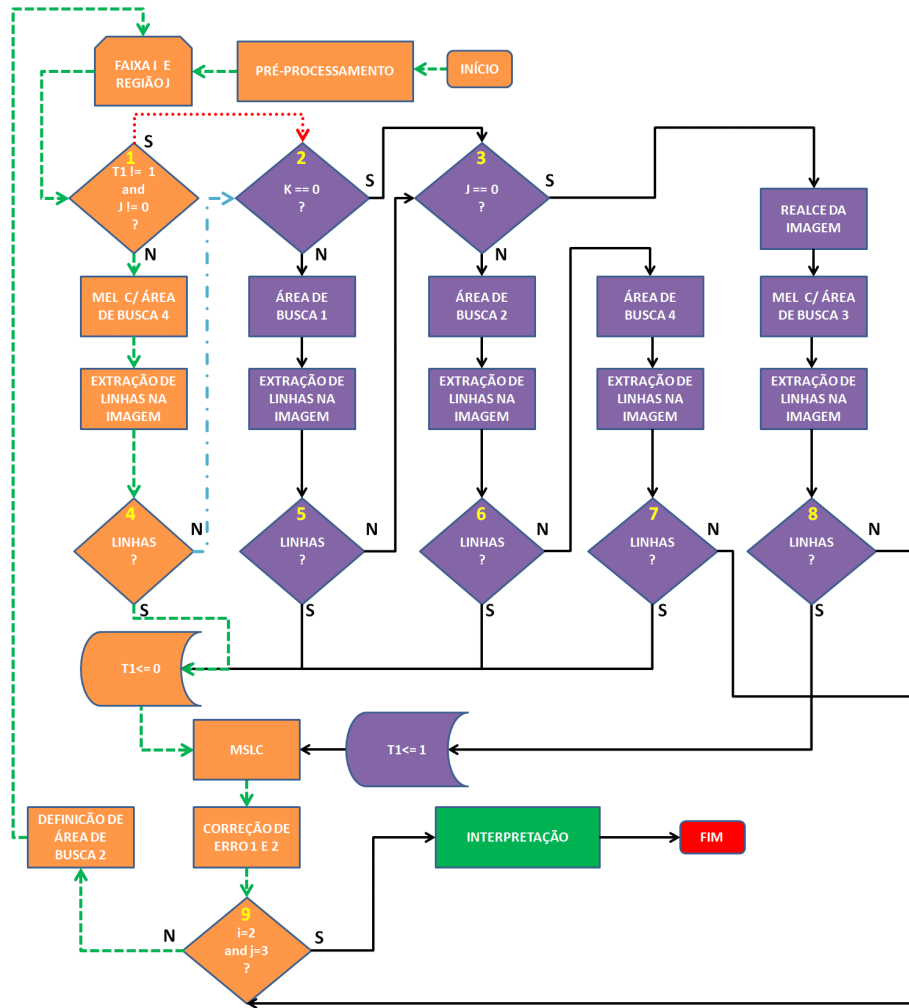


Figura 39: Ilustração da primeira possibilidade de sequência de execução do sistema.

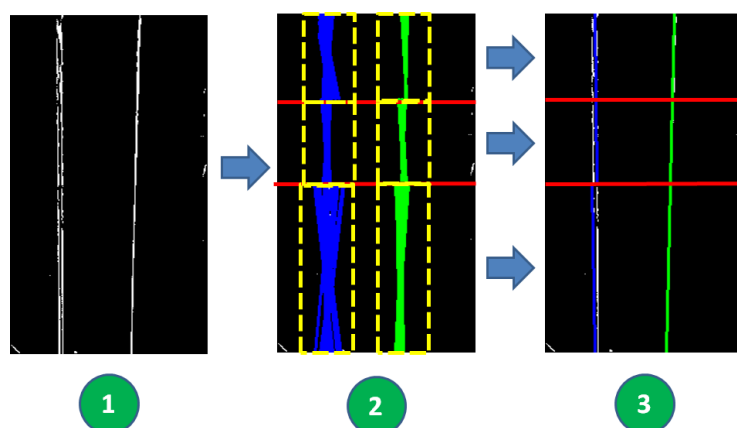


Figura 40: Ilustração dos resultados obtido através da execução do algoritmo de MSLC sobre as linhas encontradas pela transformada de Hough.

MSLC executado pelo sistema. Onde em (1) temos o cenário onde é feita a extração das faixas de sinalização. Os segmentos de linhas obtidas através do processo de Extração de

Características são ilustradas em (2). Já em (3) é apresentada o resultado da aplicação do algoritmo MSLC sobre as linhas encontradas pela transformada de Hough.

Em seguida, o sistema executa o procedimento de Correção de Erro 1 e 2 (apresentado no subcapítulo 3.11), onde na Correção de Erro 1, os comprimentos dos segmentos de linha encontrados são verificados para que todos os segmentos tenham sempre o mesmo comprimento. O comprimento dos segmentos referentes as regiões (b) e (c) possuem o comprimento de um quarto da altura total da imagem da pista no formato *Bird's Eye* e o comprimento dos segmentos de linha da região (a) devem possuir o comprimento igual a metade do tamanho total da imagem no formato *Bird's Eye*. Na Correção de Erros 2 analisamos os ângulos e a distância entre os segmentos de linha da faixa da direita e seus correspondente da faixa da esquerda para eliminar linhas muito próximas ou muito distantes uma da outra.

Em seguida, o sistema atualiza a área de busca 2, que é a área de busca fornecida pelo MEL, que utiliza as linhas encontradas na região (a) para determinar as áreas de busca para a região (b), e as linhas encontradas na região (b) para determinar as áreas de busca para a região (c), conforme foi apresentado no subcapítulo 3.10.

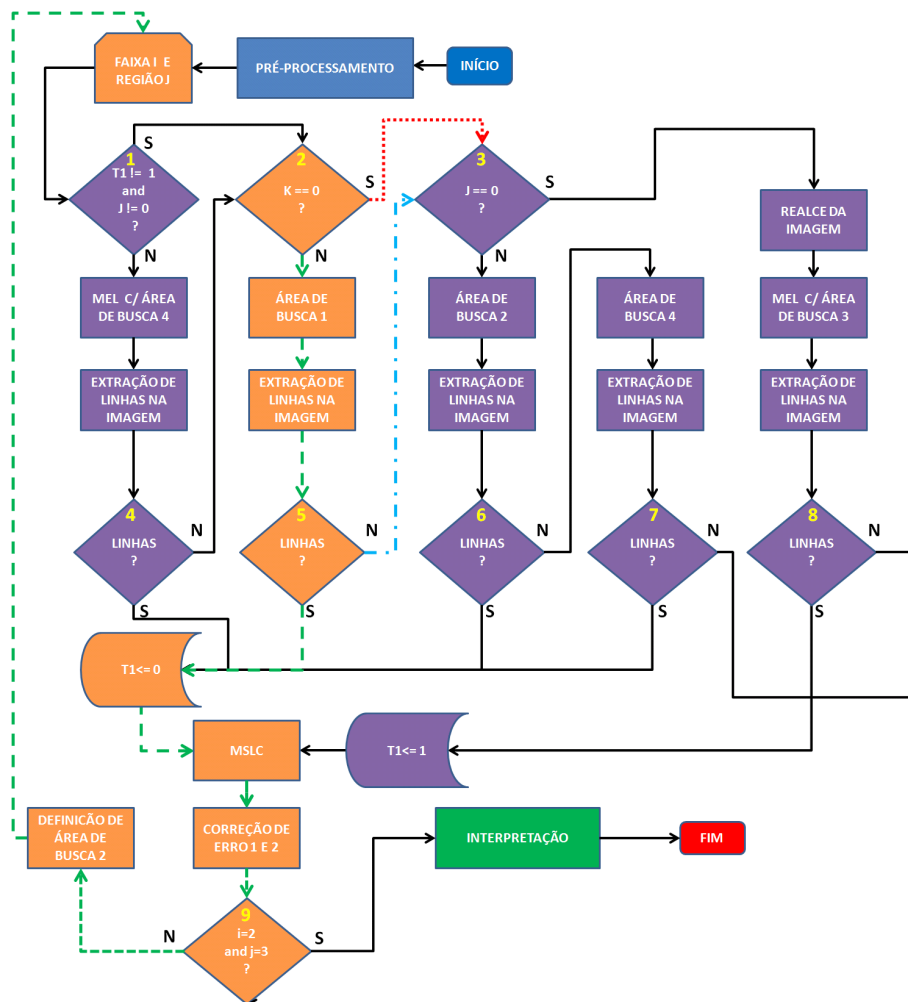


Figura 41: Ilustração da segunda possibilidade de sequência de execução do sistema.

A segunda possibilidade de sequência de execução do sistema é ilustrado na Figura 41. Quando $k=0$, significa que não existem uma área de busca fornecida pelo filtro de Kal-

man e o sistema deve passar para o teste número 3 (indicado pelo caminho de bolinhas vermelhas).

Caso $k=1$, significa que existe uma área de busca para esse segmento de linha fornecido pelo filtro de Kalman. Logo, o sistema aplica os procedimentos de extração de linhas na área de busca 1 e verifica se foram encontradas linhas nessa área. Caso o sistema tenha encontrando linhas elas são enviadas para o algoritmo MSLC, Correção de Erro 1 e 2 e definição da área de busca 2. Caso contrário o sistema passa para o terceiro teste (indicado pelo caminho traço e ponto na cor azul).

A próxima sequência de execução do sistema, é ilustrado na Figura 42. Caso $j=1$ significa que os sistema está realizando as buscas pelos segmentos de linha nas regiões (b), logo, o sistema vai utilizar a área de busca 2 para aplicar o procedimento de extração de linhas, que é a área de busca obtida através do MEL na iteração passada ($j=0$). Caso $j=2$ significa que os sistema está realizando as buscas pelos segmentos de linha nas regiões (c), logo, o sistema vai utilizar a área de busca 2 para aplicar o procedimento de extração de linhas, que é a área de busca obtida através do MEL na iteração passada ($j=1$). As linhas obtidas são enviadas para o algoritmo MSLC, Correção de Erro 1 e 2 e definição da área de busca 2.

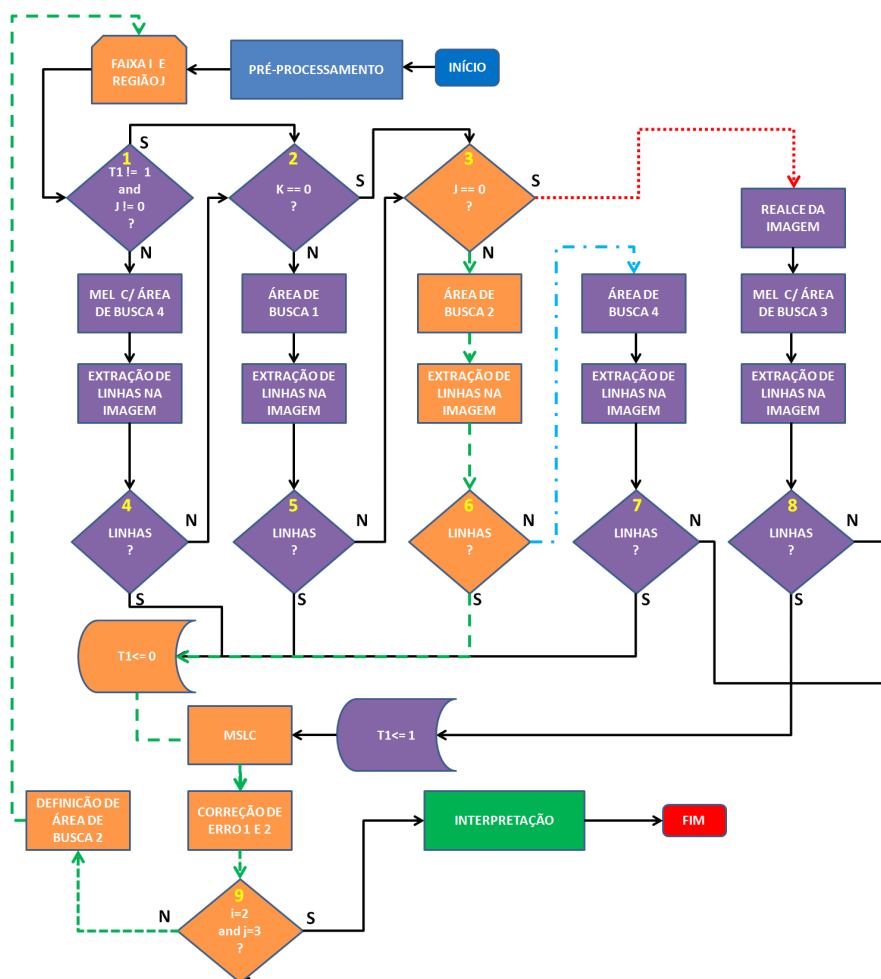


Figura 42: Ilustração da terceira possibilidade de sequência de execução do sistema.

Se nenhuma linha foi encontrada, o sistema faz uma última tentativa de extração, utilizando a área de busca 4 como chute para o MEL (indicado pelo caminho traço e

ponto na cor azul), que é uma região de busca fixa, obtida experimentalmente, baseada na posição relativa das faixas de sinalização em relação ao veículo, ela é utilizada em último caso, nas regiões (b) e (c) quando o sistema não encontra nenhum segmento de linha na região (a), por um período maior que 3 imagens da pista consecutivas, pode significar que o veículo está saindo de uma pista sem faixas de sinalização ou bordas da pista detectáveis para uma região de início demarcações na pista, logo para detectarmos as faixas nas regiões (b) ou (c) utilizamos a área de busca (4) nessas regiões para extrairmos as faixas de sinalização que estão distantes do veículo. Essa sequência de execução é ilustrada pela Figura 43. Caso o sistema não tenha encontrado nenhuma linha a etapa de busca é finalizada e inicializada a busca pelo próximo segmento de linha (indicado pelo caminho de bolinhas vermelhas).

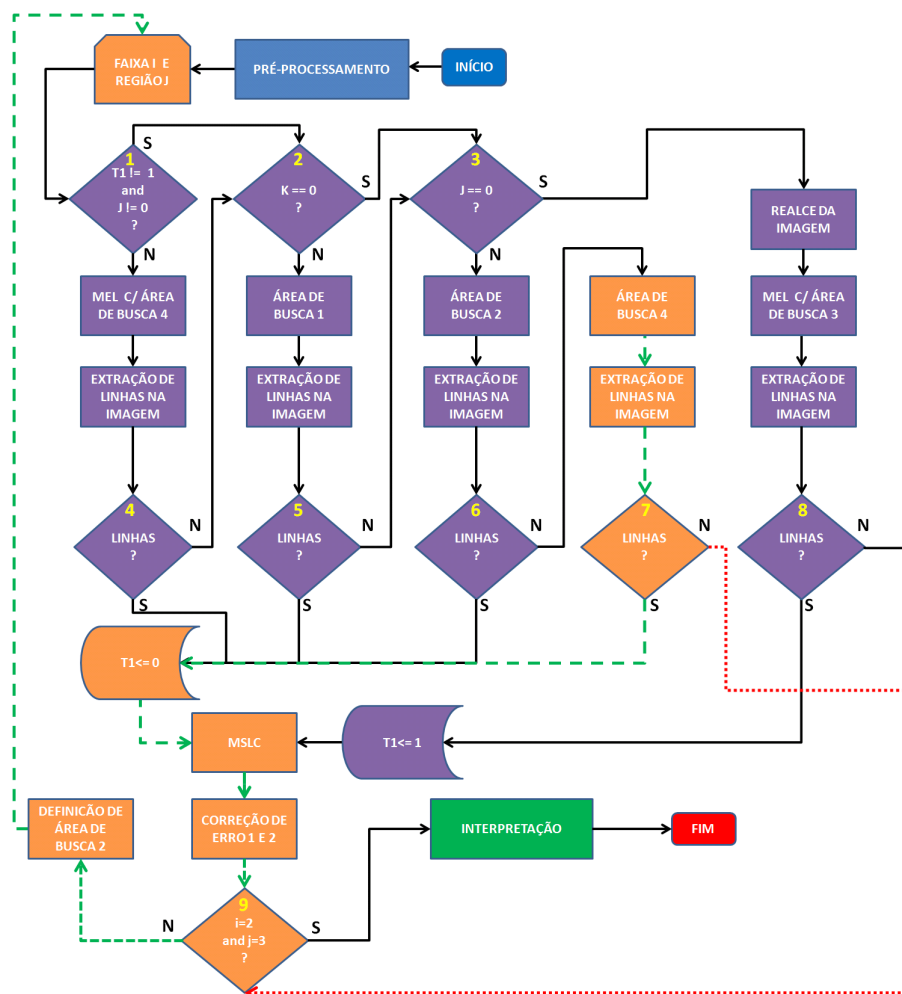


Figura 43: Ilustração da quarta possibilidade de sequência de execução do sistema.

Por fim, a Figura 44 ilustra a sequência de passo que o sistema segue quando $J=0$. Quando $J=0$ significa que o sistema está realizando as buscas pelos segmentos de linha na região (a), logo, o sistema vai realizar a busca na área de busca 3 (área onde se espera encontrar a borda da pista). Porém, antes de realiza a busca na área 3 o sistema aplica um realce no contraste da imagem, para obter melhores resultados na etapa de extração de linhas da imagem. Caso o sistema tenha tido existo em encontrar linhas a variável $T1$ é atribuída com o valor 1, indicando que o sistema está rastreando a linha que descreve a

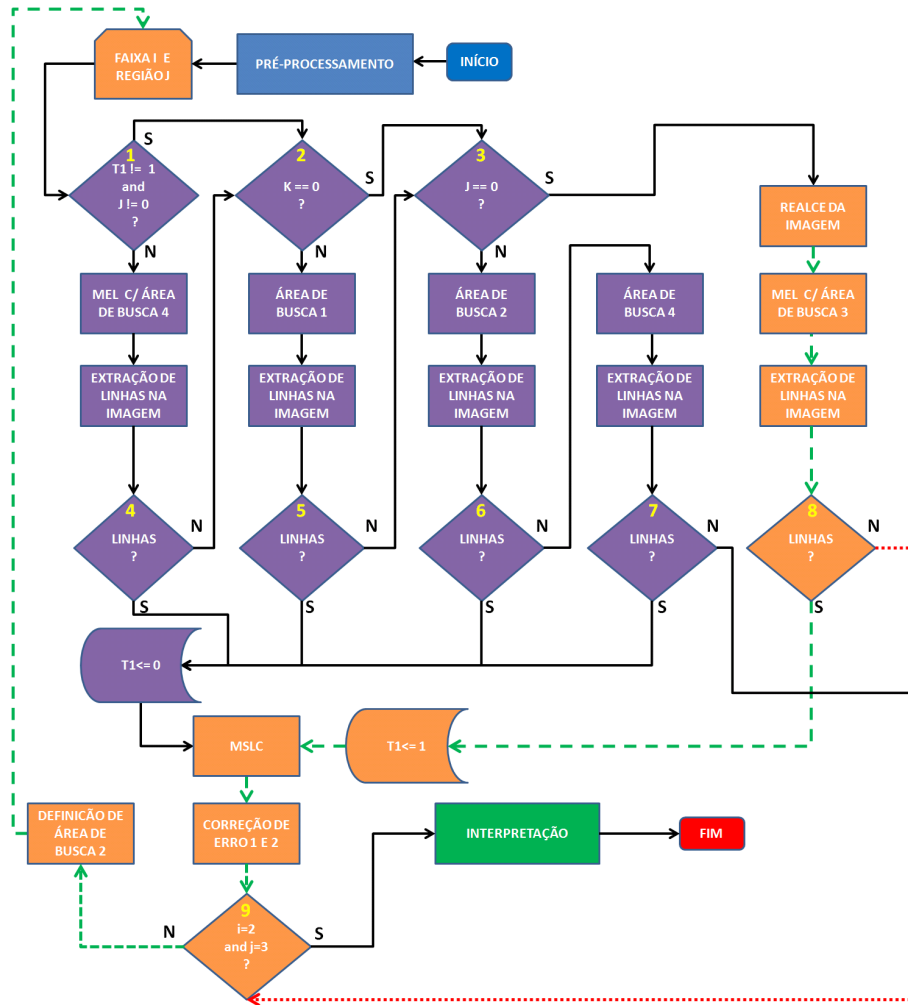


Figura 44: Ilustração da quinta possibilidade de sequência de execução do sistema.

borda da pista e as linhas são enviadas para o algoritmo MSLC, Correção de Erro 1 e 2 e definição da área de busca 2. Caso contrário significa que o sistema não conseguiu extrair nenhuma linha nas regiões de busca e deve inicializa a busca pelo próximo segmento de linha (indicado pelo caminho de bolinhas vermelhas). Essas sequências de execução são executadas seis vezes, uma para cada segmento de linha que desejamos extrair da imagem da pista. Ao final das seis iterações o sistema parte para a etapa de Interpretação.

4.1.4 Interpretação

A última etapa executada pelo sistema é a etapa de Interpretação. Na etapa de interpretação são analisados os segmentos de linha encontrados na etapa de Extração de Característica, são preenchidas lacunas, erros são identificados e corrigidos quando possível, parâmetros são atualizados, e por fim, é realizar a transformação inversa de perspectiva das linhas encontradas para apresentá-las ao usuário ou para o sistema de direção autônomo.

O último procedimento de correção de erros (Correção de Erros 3) é utilizado para corrigir as lacunas na detecção dos segmentos de linha. Quando o veículo está passando por um trecho da pista que possui faixas de sinalização do tipo tracejada, podem haver lacunas na detecção como a ilustrada na Figura 46. Esse tipo de lacuna pode prejudicar

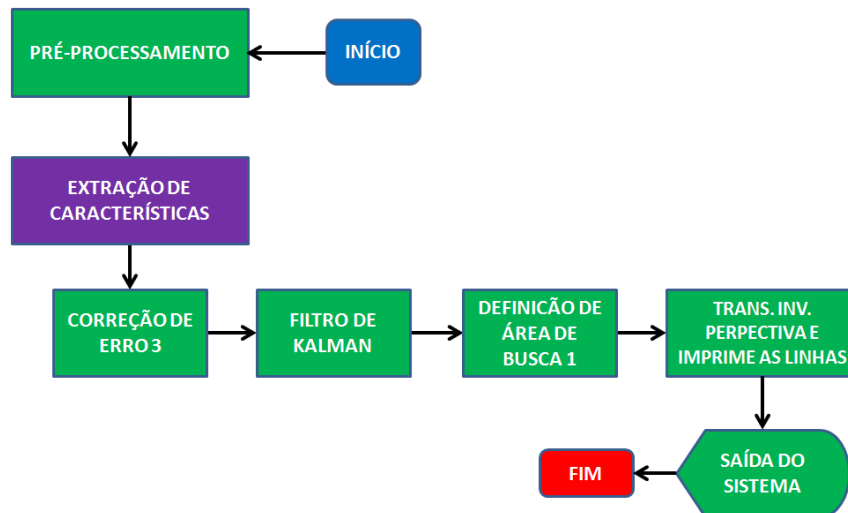


Figura 45: Ilustração das etapas de Interpretação executadas pelo sistema.

a correção e previsão feitas pelo filtro de Kalman e tornar a delimitação da região de tráfego na saída do sistema menos eficaz em prover o caminho pelo qual o veículo deve trafegar. Para eliminar essas lacunas na detecção da região (b), definimos uma linha do final do segmento de linha da região (a) até o início do segmento de linha da região (c) para eliminar essas lacunas na detecção.

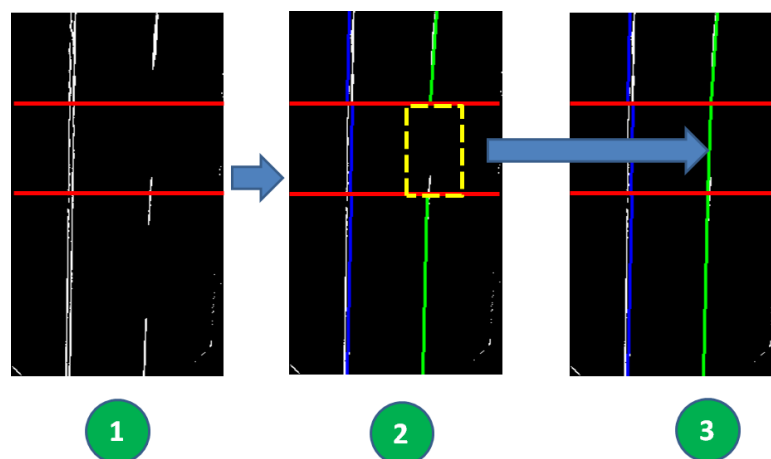


Figura 46: Ilustração da correção das lacunas entre os segmentos de linha, devido as faixas de sinalização tracejadas.

O próximo procedimento executado é o filtro de Kalman. O filtro de Kalman apresentado no subcapítulo 3.9 é utilizado para corrigir as linhas, de forma a garantir uma transição mais suave das linhas encontradas na imagem da pista passada para atual. Além da correção, o filtro de Kalman gera uma previsão dos segmentos de linha que ele espera encontra na próxima imagem da pista, esses segmentos de linha previstos são utilizadas para criar a área de busca 1. Para definir a área de busca 1, cria-se uma retângulo de largura 20 píxeis no centro da previsão e altura igual ao da região para onde a previsão foi feita.

Em seguida, o sistema executa o procedimento de transformação inversa de perspectiva nos segmentos de linha encontrados, que é realizada multiplicando-se todos os pontos de início e fim dos segmentos de linha encontrados pela matriz homográfica inversa (\mathbf{H}^{-1}). Por fim, o sistema adiciona as linhas detectadas na imagem da pista ao mesmo tempo que delimita a região de tráfego entre elas. A Figura 47 ilustra a saída do sistema de detecção de faixa de sinalização desenvolvida neste trabalho. Nessa Figura, (A) ilustra os segmentos de linha obtidos pelo sistema na perspectiva *Bird's Eye* da imagem da pista. Após a transformação inversa de perspectiva, as faixas detectadas e a região de tráfego são apresentadas ao condutor em (B).

Esse sistema foi executado em um PC intel core i5 com 8 Gb de memória raw em um ambiente linux, através da IDE Pycharm. Nessas condições, foi possível executar o algoritmo em tempo real, em um tempo médio de operação da ordem de 13,3 ms.

Por fim, ressaltamos que este trabalho foi desenvolvido com finalidade de reduzir o custo computacional associado as técnicas de processamento de imagens, de formar a permitir que o sistema seja embarcado em plataformas de hardware de baixa performance.

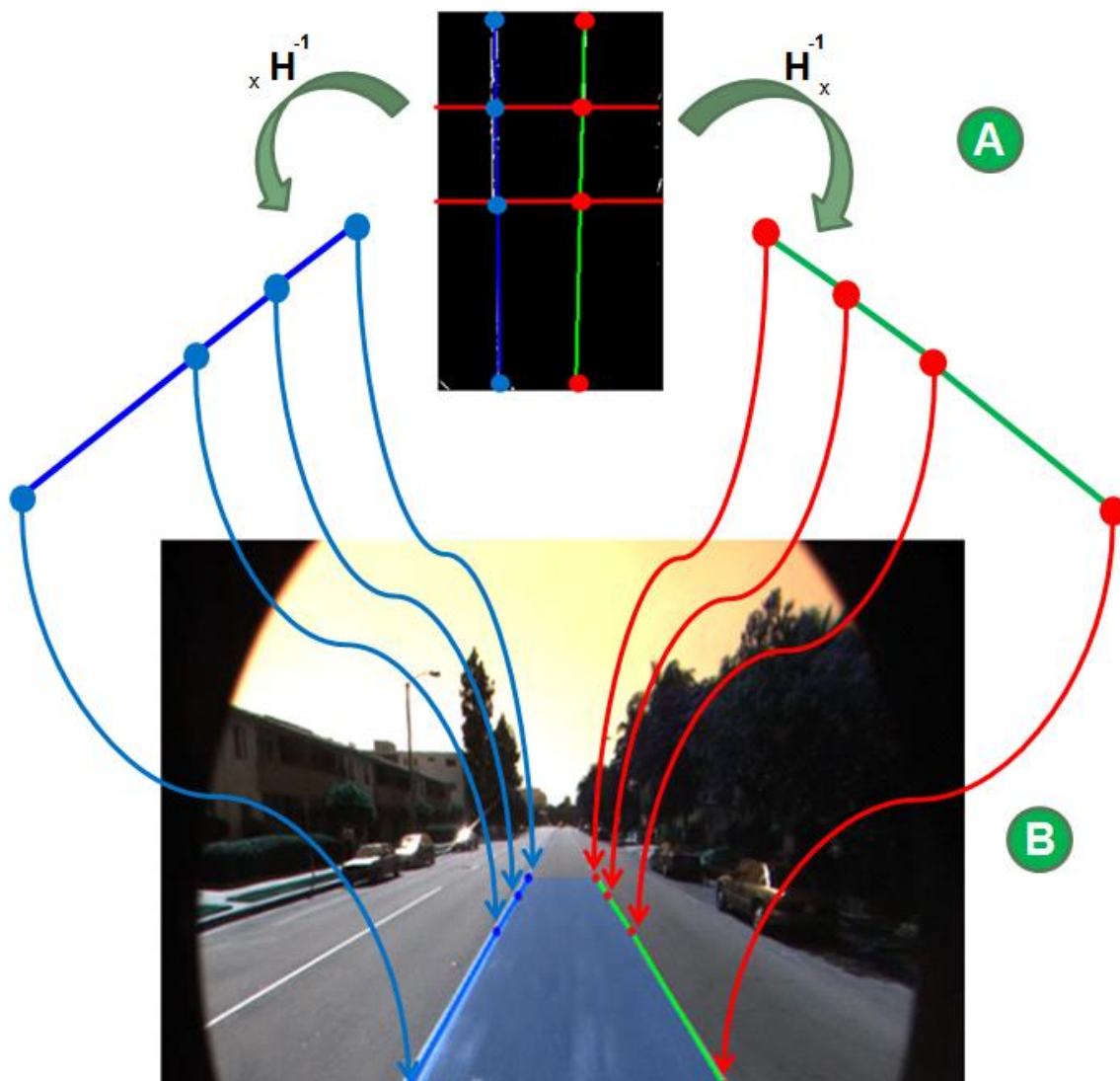


Figura 47: Ilustração da saída do sistema de detecção de faixas de sinalização desenvolvido nesse trabalho.

5 EXPERIMENTOS E RESULTADOS

Nos capítulos anteriores foi proposto um sistema de detecção em tempo real de faixas de sinalização de trânsito horizontal, capaz de operar tanto em cenários intermunicipais como em cenários urbanos. Através da utilização de áreas de busca, foi proposto um método de otimizar as regiões da imagem onde as técnicas processamento de imagem que são utilizadas para extrair as faixas de sinalização da imagem da pista, resultando em uma diminuição do custo de processamento associado a essas técnicas.

Os experimentos e validação desse trabalho foram implementados em Python 2.7, usando bibliotecas do OpenCV versão 2.4.11 em computador LINUX Intel i5 3.3 GHz. Utilizando imagens do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH), que foi obtido utilizando o veículo autônomo Alice (ALY, 2008) e do Instituto de Informática da Universidade Federal do Rio Grande do SUL (UFRGS) (PAULA, 2015)

O banco de dados da CALTECH foi escolhido por ter sido feito em cenários urbanos complexos, contemplando diversas ruas na cidade de Pasadena, Califórnia. As imagens desse banco de dados foram obtidas em diferentes momentos do dia, em diversos formatos de pista (retas e curvas), e sobre diferentes condições de iluminação e sombreamento da pista. Ele é dividido em quatro bancos de dados: Cordova 1, Cordova 2, Washington 1 e Washington 2. As informações referentes a este banco de dados são mostrados na Tabela 2.

Tabela 2: BANCO DE DADOS - CALTECH

Cenário	Nome	Imagens	Faixas	Faixas Pintadas	Bordas
1	Cordova1	250	480	480	0
2	Cordova 2	406	738	264	274
3	Washington 1	336	670	645	25
4	Washington 2	232	461	461	0
	Total	1224	2349	2050	299

Na Tabela 2 a coluna "Nome" identifica o cenário. A coluna "Imagens" mostra o número total de imagens no cenário; A coluna "Faixa" mostra o número total de marcações horizontais presentes no cenário; A coluna "Faixas Pintadas" mostra o número total de faixas de sinalização do tipo pintada na pista; A coluna "Bordas" mostra o número total de bordas da pista considerados com limitadores da região de tráfego do veículo sobre a pista na ausência das faixas de sinalização na pista.

Para obter uma avaliação quantitativa precisa do algoritmo, identificamos todas as faixas visíveis de interesse em todas as imagens do banco de dados individualmente. O banco de dados da CALTECH engloba um total de 1224 imagens de dimensões de $680 \times$

480, contendo 2349 faixas de sinalização horizontal, sendo 2050 faixas de sinalização pintadas sobre a pista (Faixas Pintadas) e 299 borda da pista (Bordas).

Esses quatro cenários apresentam uma série de condições desafiadoras para sistemas de detecção de faixas de sinalização. No cenário Cordova 1, possui um total de 250 imagens, contando 471 marcações de sinalização de interesse, contendo com diversas faixas de travessia de pedestres e algumas curvas.

No Cordova 2, temos um total de 406 imagens, totalizando 720 marcações horizontais, onde 257 dessa total são bordas da pista, que na ausência da faixas pintadas na pista, é utilizada como limite da área de tráfego do veículo sobre a pista. Nesse cenário, uma intensa luz solar está incidindo sobre a pista e contem vários veículos trafegando e estacionados.

O início do cenário Washington 1 é o trajeto mais desafiador dentre todos, pois esse trecho de pista possui uma cor de pavimento variado, os cordões da calçada possuem cores fora do padrão e devido a grande quantidade de sombras de árvores, somado, a forte incidência da luz solar, elevando de forma significativa o nível de dificuldade para extrair os segmentos e linha que descrevem as faixa de sinalização na pista. O último cenário Washington 2, possuem faixas de travessia de pedestres, e muita sinalização horizontal na forma escrita.

O banco de dados da UFRGS foi escolhido por contemplar diversos cenários de rodovias Brasileiras, para que dessa forma seja constatada a efetividade do sistema implementado neste trabalho, não somente em áreas urbanas, mas em rodovias e auto-estradas. Ele é dividido em quatro bancos de dados: clip_i5s_0085, clip_i5s_0094, clip_i5s_0096, clip_i5s_0104.

O banco de dados da UFRGS engloba um total de 10831 imagens de dimensões de 680x480, das quais contem um total de 21571 faixas de sinalização horizontal, todas faixas de sinalização pintadas sobre a pista. As informações referentes a este banco de dados são mostrados na Tabela 3 e suas respectivas divisões são idênticas a da Tabela 2.

Tabela 3: BANCO DE DADOS - UFRGS

Cenário	Nome	Imagens	Faixas	Faixas Pintadas	Bordas
1	clip_i5s_0085	1006	2012	2012	0
2	clip_i5s_0094	2160	4320	4320	0
3	clip_i5s_0096	6192	12357	12357	0
4	clip_i5s_0104	1473	2882	2882	0
	Total	10831	21571	21571	0

Estes cenários contemplam de forma geral os principais desafios enfrentados por sistemas de detecção de faixas de sinalização em rodovias. O primeiro cenário possui 1006 imagens e 2012 faixas de sinalização brancas e amarelas. Nesse cenário o sistema deve lidar com as variações de iluminação incidente sobre a vista devido a oclusões causadas por nuvem.

O segundo cenário possui 210 imagem e contêm 4320 faixas de sinalização de interesse. Este cenário apresenta variações na cor do pavimento, faixas desbotadas, auto fluxo de veículos nos dois sentidos do trajeto e a pista passa por diversas elevações e declínios.

O terceiro cenário é o maior de todos, apresenta um total de 6192 imagens e possui 12357 faixas de sinalização de interesse. A incidência de luz solar sobre este cenário é bastante intensa, tornando a detecção das faixas na cor amarela mais desafiadora. O cenário ainda apresenta muitas curvas, alto fluxo de veículos e trechos de oclusão causado por areia e faixas deterioradas pelo tempo.

O último cenário apresenta uma típica via de tráfego de três faixas e auto fluxo que tramita entre municípios brasileiros. Este cenário é constituído, na sua maior parte, por faixas tracejadas e possui pouca incidência de luz solar devido a nuvens.

Para realizar os experimentos, em ambos bancos de dados apresentados, todas as imagens foram analisados individualmente e os critérios de seleção adotados foram similares aos de Aly (2008). Eles são os seguintes: Para que uma linha detectada seja considerada correta, a maior distância entre a linha detectada e a faixa de sinalização verdadeira não pode ser maior que 15 píxeis.

A técnica de mudança de perspectiva, embora, facilite a detecção das faixas e bordas, pode causar problemas na detecção das linhas mais distantes do veículo devido à distorção causada pelo procedimento. Com isso em mente, definimos um limite, levando em conta o ponto em que a distorção pode diminuir a confiança da detecção das faixas de sinalização mais distantes do veículo, esses limites estão ilustrados na Figura 48. Esses limites podem variar dependendo de como o procedimento de mudança de perspectiva é executado. As linhas detectadas acima dessa distância foram consideradas falso-positivas.



Figura 48: Ilustração dos limites usados para definir a região da pista em que as faixas de sinalização e borda da pista detectadas são consideradas corretas.

A detecção de qualquer marcação transversal (em relação ao veículo), faixa de travessia de pedestres e outras formas de sinalização horizontal presente na pista é considerada um falso-positivo. Em regiões onde as faixas de sinalização pintadas na pista serão interrompidas ou são introduzidas, a detecção tanto da borda da pista quanto da faixa pintada na pista são aceitas como corretas. A maioria das métricas definidas a cima foram baseadas nas métricas utilizadas por Aly (2008), de forma a garantir que a comparação entre os dois trabalhos fosse a mais confiável possível.

Nessas condições, a Tabela 4 ilustra os resultados obtidos nos quatro cenários do banco de dados da CALTECH. Com base nos resultados ilustrados na Tabela, fica evidente a eficácia do sistema implementado neste trabalho. Foi obtido uma taxa média de acerto de 97.94%, evidenciando a robustez do sistema em cenários urbanos complexo. A taxa de detecção de Falso Positivo (FP) foi de 3.51%, essa é uma taxa relativamente alta, porém, o sistema foi idealizado para detectar tanto faixas de sinalização quanto as bordas da pistas, e consequentemente o número de linhas que o sistema deve obter e analisar cresce

significativamente em tamanho e complexidade. A taxa de FP por imagem obtida foi mais balanceada, variando bastante nos cenários com menos faixas de sinalização pintadas e mais bordas da pista. As Figuras 49, 50, 51 e 52 ilustram alguns dos resultados obtidos nos quatro cenários do banco de dados da CALTECH.

Tabela 4: RESULTADOS - LAPSI

Cenário	Faixas	Detectado	Taxa de acerto	Falso Positivo	FP/Imagem
1	480	484	99.38%	1.45%	0.01458
2	738	768	96.61%	7.45%	0.07452
3	670	672	97.76%	2.53%	0.02537
4	461	464	97.39%	3.25%	0.03675
		Média	97.78%	3.67%	0.03519

Na Tabela 4 a coluna "Faixas" mostra o número total de faixas de sinalização em cada cenário; A coluna "Detectado" mostra o número de faixas e bordas da pista detectadas; A coluna "Taxa de Acerto" mostra a taxa de detecção correta em cada cenário; A coluna "Falso Positivo" mostra a taxa de falsos positivos; A última coluna mostra a relação Falso Positivo e Número de imagens no cenário.

De forma a possibilitar a comparação do sistema desenvolvido nesse trabalhos com outro trabalho relacionado, disponível na literatura, foi necessário reavaliar os resultados obtidos por Aly (2008) da CALTECH, sobre os mesmo critérios de validação definidos nesse capítulo. Essa reavaliação foi necessária pois, pelos critérios adotados por Aly, a detecção de bordas da pista são consideradas falso-positivas e devido à não ter sido possível totalizar o mesmo número de faixas encontradas por Aly no banco de dados da CALTECH. Dessa forma, a Tabela 5 ilustra os resultados obtidos pelo sistema desenvolvido por Aly, sobre os nossos critérios de validação.

Tabela 5: RESULTADOS - CALTECH

Cenário	Faixas	Detectado	Taxa de Acerto	Falso Positivo	FP/Imagem
1	480	467	95.20%	2.29%	0.02008
2	738	630	82.11%	3.25%	0.03252
3	670	647	96.44%	1.29%	0.01940
4	461	440	93.92%	1.51%	0.015184
		Média	91.46%	2.19%	0.02179

Uma vez que, a detecção das bordas da pista é classificado como um falso-positivo por Aly, utilizamos somente os cenários constituídos na sua maior parte ou completamente por faixas de sinalização pintadas na pista, de forma a realizar uma comparação justa entre os dois trabalhos. Os cenários considerados nessa comparação foram: Cordova 1, Washington 1¹ e Washington 2. Por fim, a Tabela 6 ilustra a comparação entre os dois trabalhos, considerando somente as imagens que possuem faixas de sinalização.

Os resultados obtido provam a superioridade do nosso sistema, onde obtivemos uma diferença positiva de 3.61% na taxa de acerto, totalizando 162 faixas detectadas a mais do que o sistema implementado por Aly. Entretanto, a taxa de falso-positivo obtida pelo

¹A 25 primeiras imagens do cenário Washington 1 não possuem faixas de sinalização do lado direito do veículos, dessa forma, elas foram retiradas e somente as demais imagens do cenário foram utilizadas para realizar avaliação do desempenho de ambos os sistemas.

Tabela 6: COMPARAÇÃO - CALTECH Vs. LAPSI

	Cenários	Faixas	Detecção Correta	Taxa de Acerto	Falso Positivo
LAPSI	1,3,4	1561	2251	98.60%	2.43%
CALTECH	1,3,4	1561	2092	95.19%	1.70%
	Diferença		159	3.41%	-0.73%

nosso sistema foi 0.52% maior. Essa taxa superior pode ser atribuída a busca pelas bordas da pista realizadas pelo sistema na ausência de faixas de sinalização, resultando numa detecção maior de linhas e conseqüentemente uma taxa de detecção de falso-positivo maior.

Outra comprovação da efetividade do sistema desenvolvido nesse trabalho, utilizando o banco de dados da CALTECH, pode ser obtida em comparação com o trabalho de Guo (2015) que obteve uma taxa de detecção correta de 95.17% e de Hoang (2016), que alcançou uma taxa de detecção correta de 94%. Nesses trabalhos, os autores seguiram métricas de avaliação diferentes das utilizadas neste trabalho, devido a isso não foi possível realizar uma comparação detalhada com esses trabalhos.

A validação realizada utilizando o banco de dados da UFRGS foi feita sobre os mesmos critérios de avaliação, isto é, respeitando as mesmas métricas utilizadas na validação com o banco de dados da CALTECH. Os resultados obtidos nos quatro cenários estão evidenciados Tabela 7.

Tabela 7: RESULTADOS - UFRGS

Cenário	Faixas	Detectado	Taxa de acerto	Falso Positivo	FP/Imagem
1	2012	2012	100%	0%	0
2	4320	4308	98.17%	1.55%	0.015509
3	12357	12331	97.63%	2.15%	0.021526
4	2882	2882	98.05%	1.94%	0.019431
		Média	98.4%	1.41%	0.014116

Foi obtido uma taxa média de detecção correta de 98.4% sobre os quatro cenários, uma taxa relativamente maior do que as obtidas no cenário urbano, isto já era esperado, pois as faixas de sinalização presentes em rodovias tendem a sofrer menos oclusões causadas por sombras de árvores ou outros veículos, e a cor do pavimento é na maioria das vezes constante. Esse tipo de cenário também teve uma influência positiva na detecção de falso-positivos, resultando em uma queda de 2.10% na taxa de detecção de falso-positivo e na metade da distribuição de falso-positivos por cenário do banco de imagens.

Esses resultados comprovam a eficiência e competência do sistema implementado nesse trabalho, não somente em áreas urbanas como comprovado na validação no banco de dados da CALTECH, mas também em cenários intermunicipais constituídos de rodovias e auto-estradas. As Figuras 53, 54, 55 e 56 ilustram alguns dos resultados obtidos nos quatro cenários do banco de dados da UFRGS.

Os resultados mostram a eficácia do sistema proposto neste trabalho na detecção de faixas de sinalização e bordas da pista em ambientes urbanos e intermunicipais, imersos sobre condições variáveis de iluminação, sombras, curvas e trechos sem todas faixas de sinalização pintadas. Os experimentos mostram um tempo médio execução de 13.9 ms por imagem em cenários urbanos, e 12.7 ms em cenários com rodovias e auto-estradas.

A taxa de falso-positivo obtida na maioria dos cenários foi relativamente alta. A maioria deles devido a detecção incorreta das faixas de cruzamento de pedestres e devido a trechos muito claros de pista (devido ao pavimento possuir diversas cores), que refletem muito a luz do sol em regiões sobreposta por grandes quantidades de sombras, ou sombras que formam linhas retas contínuas em um ângulo similar ao das faixas de sinalização são as principais fontes de detecção de falso-positivo. Porém, a maior deficiência constatada do nosso sistema é a influência dos veículos na precisão da detecção do nosso sistema. Esse foi a causa da maioria dos casos de detecção falso-positiva durante a validação utilizando o banco de dados da UFRGS. A Figura 57 ilustra alguns dos principais casos de detecção de falso-positivo.

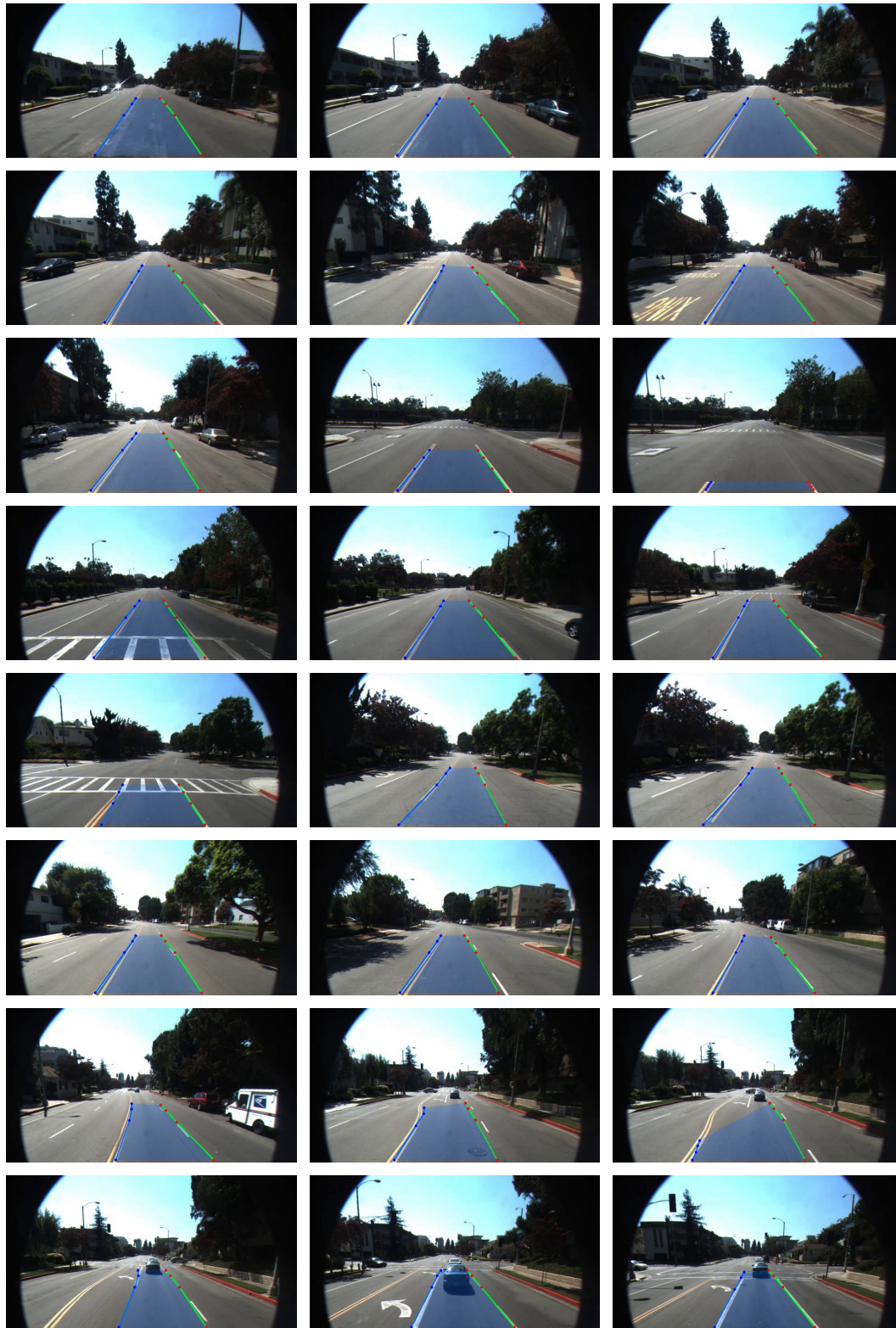


Figura 49: Ilustração dos resultados obtidos no cenário Cordova 1 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).



Figura 50: Ilustração dos resultados obtidos no cenário Cordova 2 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).



Figura 51: Ilustração dos resultados obtidos no cenário Washington 1 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).



Figura 52: Ilustração dos resultados obtidos no cenário Washington 2 do banco de dados do Instituto de Tecnologia da Califórnia (CALTECH).

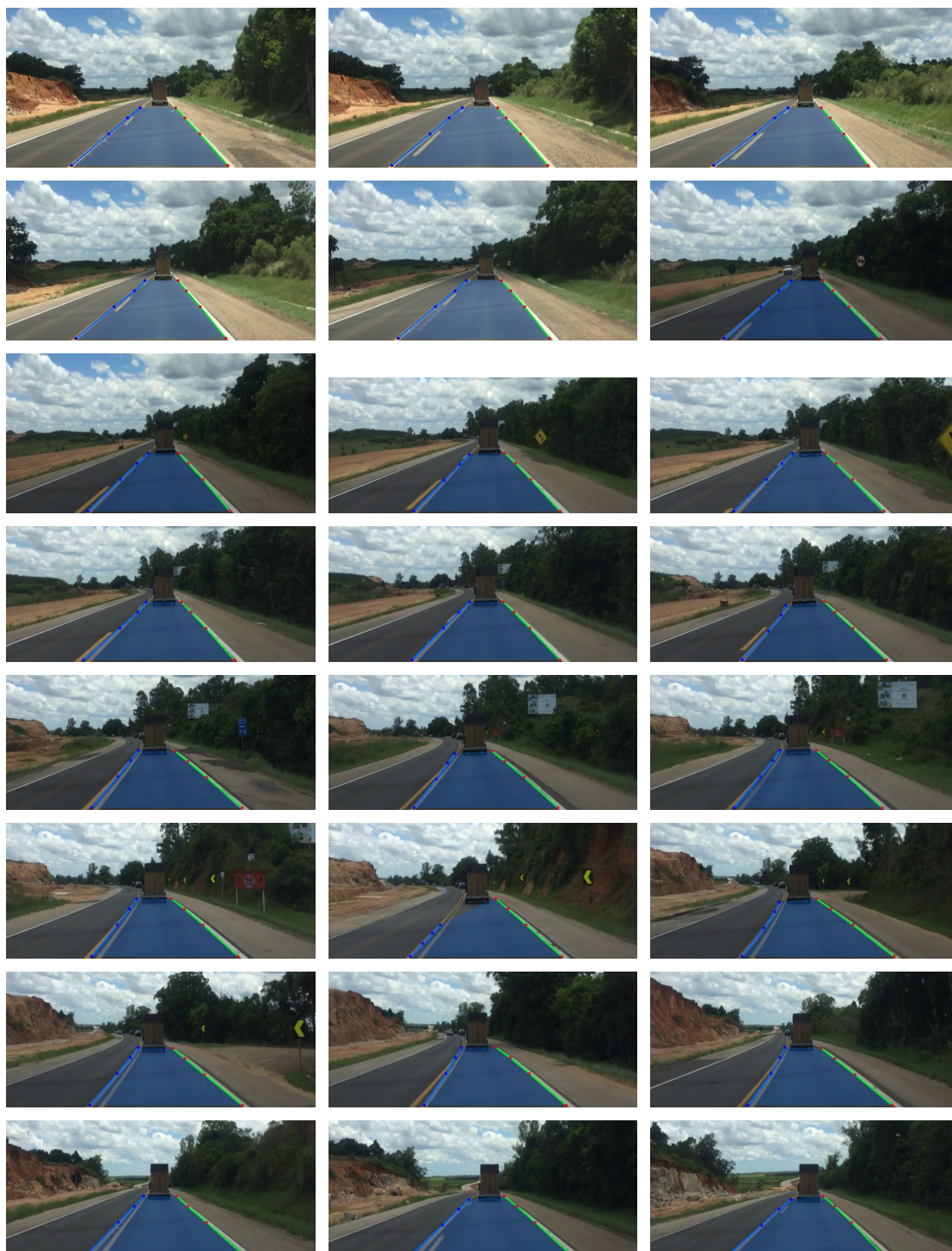


Figura 53: Ilustração dos resultados obtidos no cenário clip_i5s_0085 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

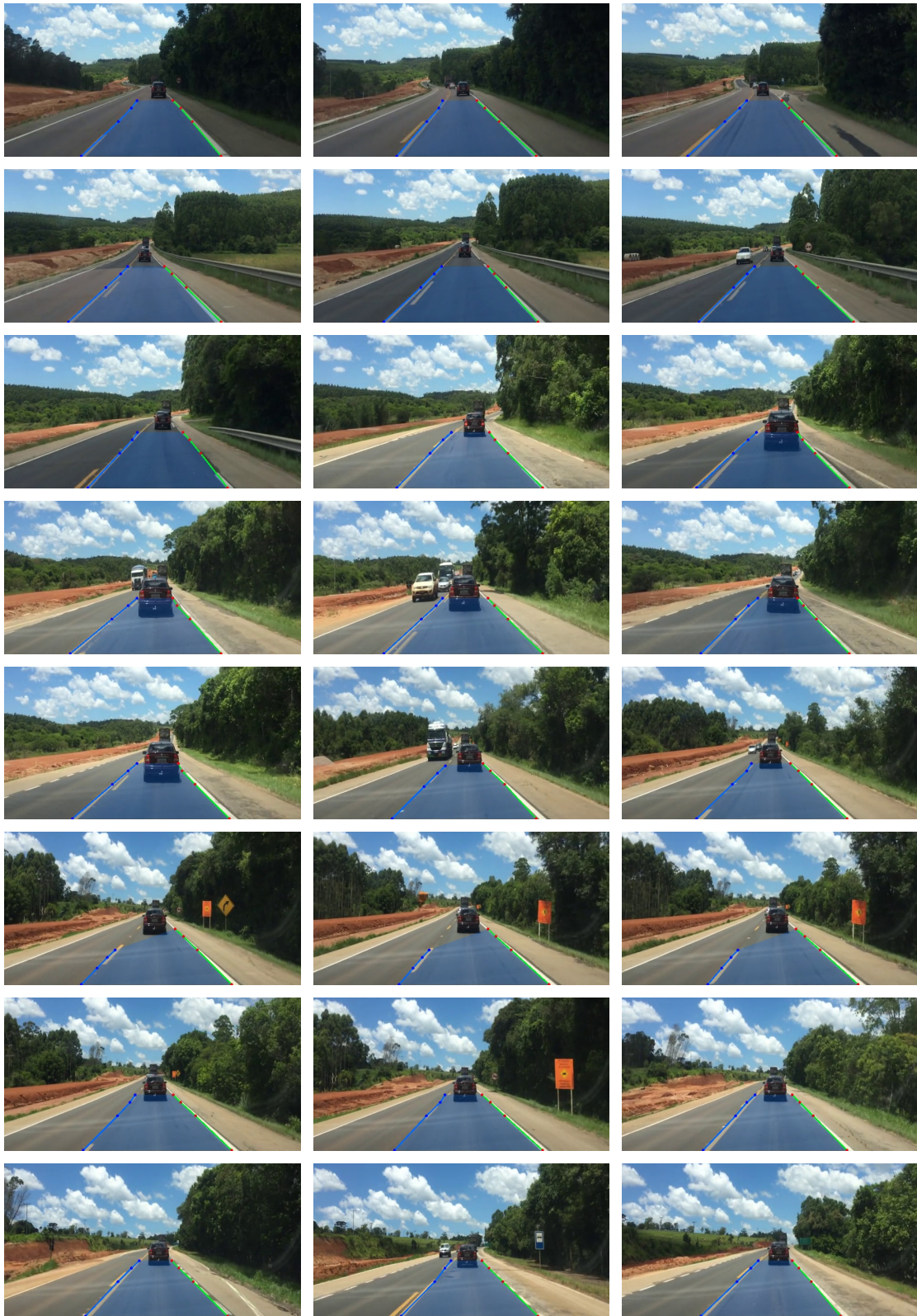


Figura 54: Ilustração dos resultados obtidos no cenário clip_i5s_0094 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

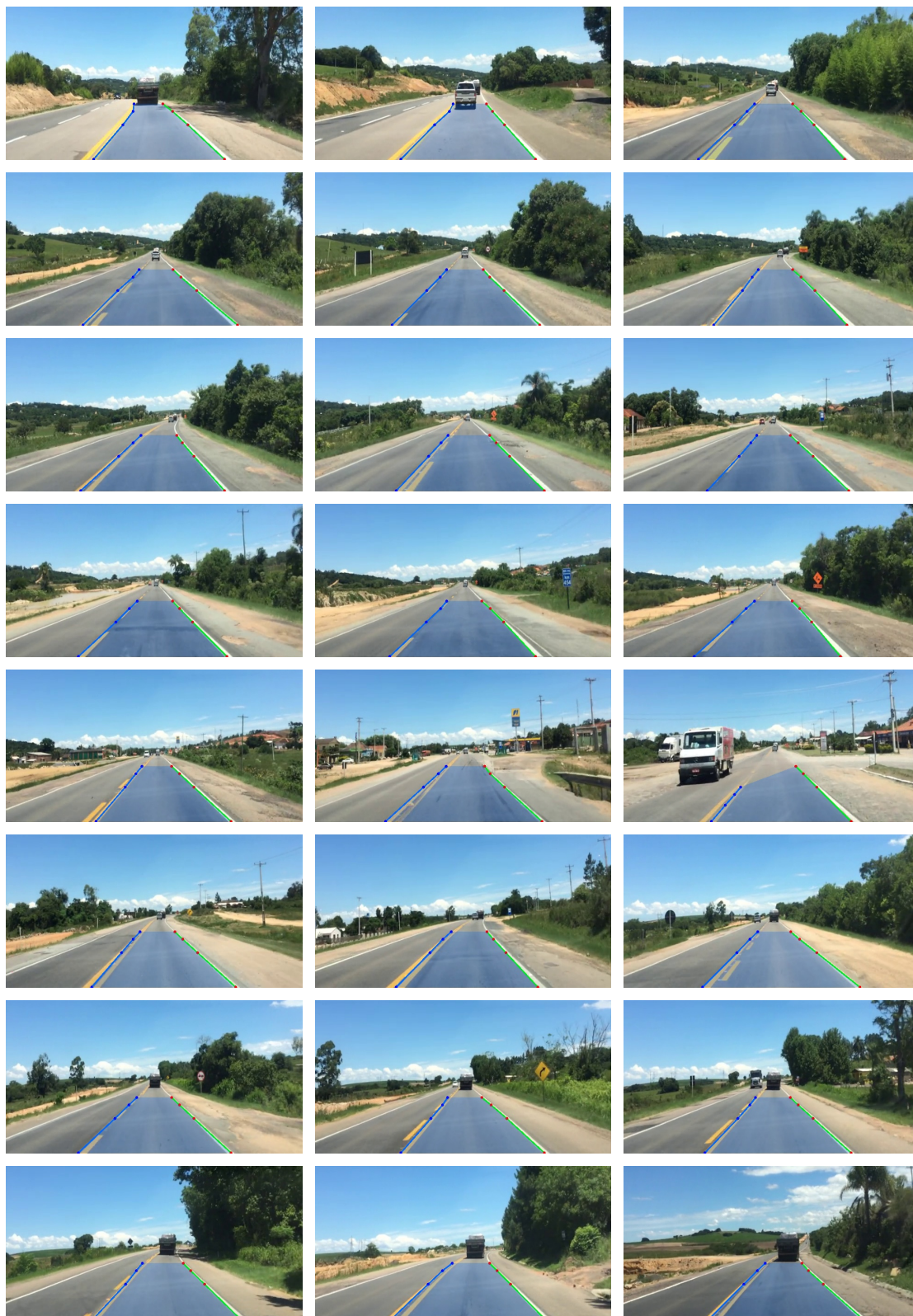


Figura 55: Ilustração dos resultados obtidos no cenário clip_i5s_0096 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).



Figura 56: Ilustração dos resultados obtidos no cenário clip_i5s_0104 do banco de dados do Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

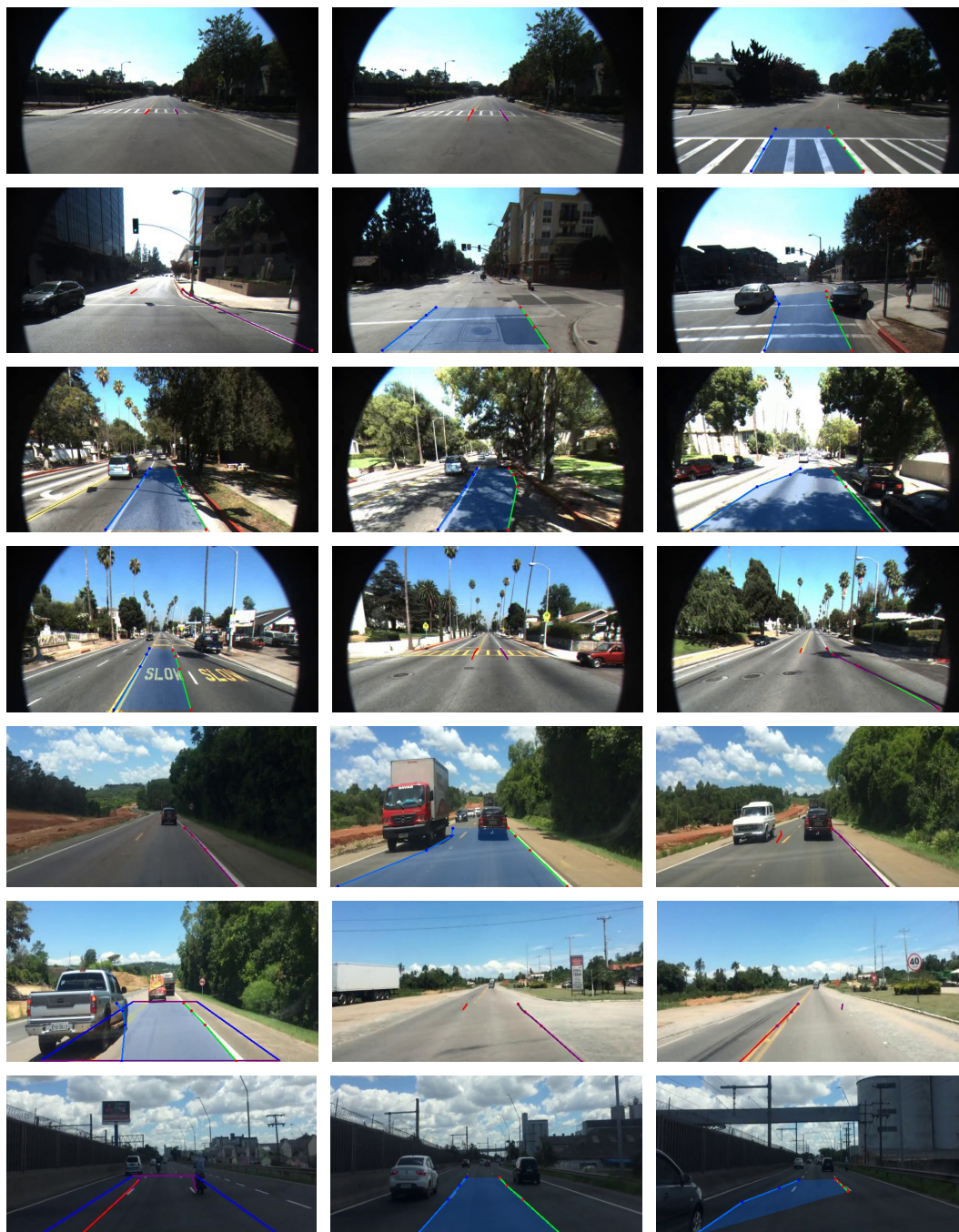


Figura 57: Ilustração de algumas detecções falso-positivas, obtidas em ambos os bancos de dados utilizados para validação neste trabalho.

6 CONCLUSÕES

Nesta dissertação, foi proposto um algoritmo de visão computacional eficiente e robusto para detectar as faixas de sinalização em cenários urbanos e intermunicipais, destinadas a aplicações em tempo real. A principal contribuição deste trabalho foi a criação das áreas de busca que isolam a região da imagem da pista onde as faixas sinalização estão contidas. Essas áreas foram criadas com o objetivo de reduzir o alto custo computacional associado a utilização da transformada de Hough, do filtro de limiar adaptativo e do algoritmo MSLC no sistema .

Através do Método de Extensão de Linha (MEL) proposto neste trabalho e pelas previsões feitas pelo filtro de Kalman foi possível reduzir em até 75% a área de busca total da imagem, resultando em uma significativa diminuição do tempo de processamento necessário para realizar a detecção das faixas de sinalização. Embora, a extração das faixas de sinalização seja uma tarefa bem difícil de ser realizada em ambientes urbanos complexos, os resultados obtidos nos experimentos foram bastante satisfatórios e apresentaram uma grande precisão na detecção de faixas de sinalização e bordas da pista ao mesmo tempo que reduziu o custo computacional do sistema. A taxa média de detecção correta obtida em ambos bancos de dados utilizados para validação nesse trabalho foi de 98,1% com um tempo médio de operação de 13 ms em um PC core i5 com 8 GB de memória.

A precisão e robustez do sistema, constatados nos experimentos apresentados, se mostraram superiores aos resultados obtidos por Aly (2008), principalmente na taxa de detecção correta das faixas de sinalização. Em comparação aparente com os trabalhos de Guo (2015) e Hoang (2016) os resultados obtidos também se mostraram superiores. Embora, a taxa de falso-positivo obtida nos cenários urbanos tenha sido alta em relação aos outros trabalhos mencionados, ela pode ser argumentada como uma consequência da estratégia de detecção adotada neste trabalho para encontrar as bordas da pista na ausência das faixas de sinalização, o qual os trabalhos da literatura mencionados a cima não possuem. Essa taxa é considerada a maior deficiência do sistema implementado neste trabalho.

A principal causa de detecção de falso-positivo identificada foi a incapacidade do sistema de evitar a interferência dos demais veículos presentes na pista. A detecção incorreta devido a presença de sombras com alinhamento muito similar aos das faixas de sinalização ou da borda da pista, provenientes de árvores, postes e de outros veículos também contribuiu de forma significativa na taxa de falso-positivos. Outros fatores que contribuíram para essa taxa, mas de forma não tão significativa, foram a incidência de luz solar muito forte no horizonte da pista que pode dificultar a detecção dos segmentos de linha mais distantes do veículo, a detecção errônea das faixas de cruzamento de pedestres e pavimentos multicoloridos com diversas fissuras de junção entre pavimentos.

O sistema desenvolvido neste trabalho foi desenvolvido tentando abranger a maioria das condições ambientais e obstáculos encontrados tanto pistas de cenários urbanos

quanto em cenários intermunicipais. Porém, ele possui algumas limitações, a primeira delas é relacionada ao modo como a transformação de perspectiva utilizada neste trabalho é realizada, essa técnica depende da posição onde a câmera, que obtém as imagens da pista, é instalada no veículo. Logo, seria necessário ter conhecimento prévio de onde a câmera seria instalada no veículo, para ser possível ajustar os oito pontos utilizados para definir o operador linear de mudança de perspectiva. A mudança da posição da câmera também acarretaria na mudança de outros parâmetros, como na posição das áreas de busca fixa 3 e 4, nos distâncias de limite inferior e superior utilizadas para eliminar falso-positivos, entre outros.

Em trabalhos futuros podemos explorar as informações de cor e textura presentes na imagem da pista, somado, as informações de contornos para ser capaz de melhorar a taxa de detecção correta do sistema ao mesmo tempo que reduz a taxa de falso-positivos. De forma complementar, uma verificação do comportamento do sistema sobre condições ambientais extremas, como neblina densa, chuva intensa e regiões com pouca iluminação disponível durante a noite deveria ser feita, com o objetivo de propor soluções de possíveis deficiências encontradas nessa verificação e melhorar o desempenho do sistema sobre esses cenários. Outra otimização seria incluir técnicas de odometria visual para realizar uma análise das estruturas tridimensionais da pista para melhorar a precisão e robustez do sistema.

REFERÊNCIAS

ALY, M. Real time detection of lane markers in urban streets. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 1., 2008, Eindhoven. **Anais...** [S.l.]: IEEE, 2008. p.7–12.

ALY, M. et al. **Sensing , Navigation and Reasoning Technologies for the DARPA Urban Challenge 1 Introduction and Overview**. Disponível em: <<http://users.cms.caltech.edu/murray/preprints/bur+07-dgc.pdf>>. Acesso em: 25 jan. 2016.

A.TAKAHASHI, Y. N. Model-based lane recognition. In: CONFERENCE ON INTELLIGENT VEHICLES, 1996, Tokyo. **Proceedings...** [S.l.]: IEEE, 1996. p.201–206.

AUDI. **Audi lane assist**. Disponível em: <<https://www.audi.co.uk/glossary/a/audi-lane-assist.html>>. Acesso em: 10 Jul. 2016.

AUDIWORLD. **Honda Announces a Full Model Change for the Inspire**. Disponível em: <<http://world.honda.com/news/2003/4030618-2.html>>. Acesso em: 11 jul. 2016.

AUNG, T.; ZAW, M. Detection of Lane Departure using 2-D FIR and Hough Transform. **International Journal of Science, Engineering and Technology Research**, [S.l.], v.3, n.5, p.1302–1306, Maio 2014.

BARROS, M. **Frota brasileira cresce onze vezes mais rápido do que a população e já é suficiente para cobrir Barbados, no Caribe**. Disponível em: <<http://veja.abril.com.br/blog/cidades-sem-fronteiras/transporte-e-transito/frota/>>. Acesso em: 30 Jun. 2016.

BERRIEL, R. F. et al. A Particle Filter-Based Lane Marker Tracking Approach Using a Cubic Spline Model. In: SIBGRAPI CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 28., 2015, Salvador. **Anais...** [S.l.]: IEEE, 2015. p.149–156.

BERTOZZ, M.; BROGGI, A.; FASCIOLI, A. Stereo inverse perspective mapping: theory and applications. **Image and Vision Computing**, Amsterdam, v.16, p.585–590, Jun. 1998.

BIBULI, M.; CACCIA, M.; LAPIERRE, L. Path-following algorithms and experiments for an autonomous surface vehicle. **IFAC Proceedings Volumes (IFAC-PapersOnline)**, [S.l.], v.7, n.1, p.81–86, Mar. 2007.

BISHOP, G.; WELCH, G. **An Introduction to the Kalman Filter**. Disponível em: <http://www.cs.unc.edu/tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf>. Acesso em: 25 Jan. 2017.

BMW. Lane Departure Warning. Disponível em: <http://www.bmw.com/com/en/insights/technology/technology_guide/articles/lane_departure_warning.html>. Acesso em: 10 Jul. 2016.

BORKAR, A.; HAYES, M.; SMITH, M. T. Robust lane detection and tracking with ransac and Kalman filter. In: IEEE INTERNATIONAL CONFERENCE ON IMAGE PROCESSING (ICIP), 16., 2009, Cairo. **Anais...** [S.l.]: IEEE, 2009. p.3261–3264.

BORKAR, A.; HAYES, M.; SMITH, M. T. A template matching and ellipse modeling approach to detecting lane markers. In: TALON, J. B. et al. (Ed.). **Advanced Concepts for Intelligent Vision Systems**. Sydney: Springer Berlin Heidelberg, 2010. v.2, p.179–190.

BROGGI, A. An image reorganization procedure for automotive road following systems. In: INTERNATIONAL CONFERENCE ON IMAGE PROCESSING, 2., 1995, Washington D.C. **Proceedings...** [S.l.]: IEEE, 1995. v.3, p.532–535.

CALMON, F. **Tecnologia contra acidentes é estudada em carros mais baratos.** Disponível em: <<http://carros.uol.com.br/colunas/alta-roda/2015/09/15/tecnologia-que-preve-acidentes-ja-e-estudada-em-carros-mais-baratos.html>>. Acesso em: 30 Jun. 2016.

CARICOS. **Volkswagen Passat Alltrack.** Disponível em: <<http://www.caricos.com/cars/v/vw/2012-volkswagen-passat-alltrack/images/63.html>>. Acesso em: 21 dez. 2016.

CASTLEMAN, K. R. **Digital Image Processing**. 1st.ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 1996.

CHEN, T.-C.; CHUNG, K.-L. An Efficient Randomized Algorithm for Detecting Circles. **Computer Vision and Image Understanding**, [S.l.], v.83, n.2, p.172–191, Ago. 2001.

CORRMAP. **The Homography transformation.** Disponível em: <http://www.corrmapp.com/features/homography_transformation.php>. Acesso em: 21 nov. 2016.

COULOMBEAU, P.; LAURGEAU, C. Vehicle yaw, pitch, roll and 3D lane shape recovery by vision. In: IEEE INTELLIGENT VEHICLE SYMPOSIUM, 1., 2002, Versailles. **Anais...** [S.l.]: IEEE, 2002. v.2, n.2, p.619 – 625.

DENG, J. et al. Fast Lane Detection Based on the B-Spline Fitting. **International Journal of Research in Engineering and Technology (IJRET)**, Bangalore, v.2, n.4, p.1–4, Maio 2013.

DEREK BRADLEY, G. R. Adaptive Thresholding using the Integral Image. **Journal of Graphics, GPU, and Game Tools**, [S.l.], v.12, n.2, p.13–21, Jan. 2007.

DERPANIS, K. G. **Overview of the RANSAC Algorithm.** Disponível em: <http://www.cse.yorku.ca/kosta/CompVis_Notes/ransac.pdf>. Acesso em: 24 jan. 2016.

DEUSCH, H. et al. A random finite set approach to multiple lane detection. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 15., 2012, Anchorage. **Anais...** [S.l.]: IEEE, 2012. p.270–275.

DILMEN, N. **Beyoglu 4671 tricolor**. Disponível em: <https://upload.wikimedia.org/wikipedia/commons/thumb/3/33/Beyoglu_4671_tricolor.png/800px-Beyoglu_4671_tricolor.png>. Acesso em: 23 dez. 2016.

DUAN, D. et al. An improved Hough transform for line detection. In: INTERNATIONAL CONFERENCE ON COMPUTER APPLICATION AND SYSTEM MODELING (ICCASM 2010), 1., 2010, Taiyuan. **Anais...** [S.l.]: IEEE, 2010. v.2, p.354–357.

DUONG, T. T. et al. Near real-time ego-lane detection in highway and urban streets. In: IEEE INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS-ASIA (ICCE-ASIA), 1., 2016, Seoul. **Anais...** [S.l.]: IEEE, 2016. p.1–4.

FAN, R.; PROKHOROV, V.; DAHNOUN, N. Faster-than-real-time linear lane detection implementation using SoC DSP TMS320C6678. In: IEEE INTERNATIONAL CONFERENCE ON IMAGING SYSTEMS AND TECHNIQUES (IST), 1., 2016, Chania. **Anais...** [S.l.]: IEEE, 2016. p.306–311.

FISCHLER, M. A.; BOLLES, R. C. A Paradigm for Model Fitting with Applications To Image Analysis and Automated Cartography. **Graphics and Image Processing**, [S.l.], v.24, n.6, p.1–15, Jun. 1981.

FORD. **Available Lane-Keeping System**. Disponível em: <<https://www.ford.com/suvs/escape/gallery/videos-and-demos/Videos/available-lane-keeping-system/FMFT1339000H/>>. Acesso em: 20 dez. 2016.

GERNOT, H. **Luminance Models For The Grayscale Conversion**. 1st.ed. São Paulo, SP, Brasil: Mcgraw Hill do Brasil, 2001.

GM. **VEHICLE SAFETY TECHNOLOGY**. Disponível em: <<http://www.gmc.com/vehicle-safety-technology.html>>. Acesso em: 11 jul. 2016.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 3rd.ed. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

GU, J.; ZHANG, Q.; KAMATA, S. i. Robust road lane detection using extremal-region enhancement. In: PATTERN RECOGNITION (ACPR), 3., 2015, Kuala Lumpur. **Anais...** [S.l.]: IEEE, 2015. p.519–523.

GUIZZO, E. **How Google's Self-Driving Car Works**. Disponível em: <<http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>>. Acesso em: 21 set. 2016.

GUO, J.; WEI, Z.; MIAO, D. Lane Detection Method Based on Improved RANSAC Algorithm. In: IEEE TWELFTH INTERNATIONAL SYMPOSIUM ON AUTONOMOUS DECENTRALIZED SYSTEMS, 12., 2015, Taichung. **Proceedings...** Piscataway: IEEE, 2015. p.285–288.

GURGHIAN, A. et al. DeepLanes: end-to-end lane position estimation using deep neural networks. In: CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION WORKSHOPS (CVPRW), 29., 2016, Las Vegas. **Anais...** [S.l.]: IEEE, 2016. p.38–45.

HARTLEY, R. I.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. 1st.ed. London, UK: Cambridge University Press, 2000.

HEIJ, R. et al. **Snakes for Lane Detection**. Disponível em: <<http://www.rmsmelik.nl/PDF/lanedetection.pdf>>. Acesso em: 25 jan. 2016.

HOANG, T. M. et al. Road lane detection by discriminating dashed and solid road lanes using a visible light camera sensor. **Sensors (Switzerland)**, [S.l.], v.16, n.8, p.1–13, Ago. 2016.

HONDA. **The Audi Q7 4.2 TDI**. Disponível em: <<http://www.audiworld.com/articles/the-audi-q7-4-2-tdi/>>. Acesso em: 11 jul. 2016.

HOUGH, P. **Method and means for recognizing complex patterns**. Disponível em: <<https://www.google.com/patents/US3069654>>. Acesso em: 24 jan. 2016.

HOWARD, B. **What is lane departure warning, and how does it work?** Disponível em: <<http://www.extremetech.com/extreme/165320-what-is-lane-departure-warning-and-how-does-it-work>>. Acesso em: 10 jul. 2016.

HUANG, X. et al. Extended-Search, Bézier Curve-Based Lane Detection and Reconstruction System for an Intelligent Vehicle. **International Journal of Advanced Robotic Systems**, [S.l.], v.12, n.9, p.132, Set. 2015.

HUARONG XU XIAODONG WANG, H. H. K. W. Q. F. A fast and stable lane detection method based on B-spline curve. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER-AIDED INDUSTRIAL DESIGN CONCEPTUAL DESIGN, 10., 2009, Wenzhou. **Anais...** [S.l.]: IEEE, 2009. p.1036–1040.

HUBER, P. J.; RONCHETTI, E. M. **Robust statistics**. 2nd.ed. Hoboken, NJ: Wiley, 2009. (Wiley Series in Probability and Statistics).

HWANG, S.; LEE, Y. FPGA-based real-time lane detection for advanced driver assistance systems. In: IEEE ASIA PACIFIC CONFERENCE ON CIRCUITS AND SYSTEMS (APCCAS), 13., 2016, Jeju. **Anais...** [S.l.]: IEEE, 2016. p.218–219.

INAVI. **Lane Departure Warning System (LDWS) in THINKWARE Dash Cam**. Disponível em: <<https://www.youtube.com/watch?v=CJqVCVzGgS0>>. Acesso em: 11 out. 2016.

INTEL. **The Internet of Things is on the Road to Autonomous Driving**. Disponível em: <<http://www.intel.com/content/www/us/en/internet-of-things/infographics/iot-autonomous-driving-infographic.html>>. Acesso em: 21 dez. 2016.

ISTOÉDINHEIRO. **20 tecnologias que ajudam a evitar acidentes de trânsito**. Disponível em: <<http://www.istoedinheiro.com.br/noticias/mercado-digital/20130520/tecnologias-que-ajudam-evitar-acidentes-transito/111138.shtml>>. Acesso em: 1 jul. 2016.

JEN, E. **How Does Autopilot React to Disappearing Lane Markings**. Disponível em: <<http://www.teslarati.com/autopilot-experience-disappearing-lane-markings/>>. Acesso em: 21 set. 2016.

- JUNG, C. R.; KELBER, C. R. A lane departure warning system based on a linear-parabolic lane model. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 1., 2004, Parma. **Anais...** [S.l.]: IEEE, 2004. p.891–895.
- JUNG, S.; YOUN, J.; SULL, S. Efficient lane detection based on spatiotemporal images. **IEEE Transactions on Intelligent Transportation Systems**, Rio de Janeiro, v.17, n.1, p.289–295, Jan. 2016.
- KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the ASME–Journal of Basic Engineering**, [S.l.], v.82, n.D, p.35–45, Mar. 1960.
- KIM, S.; LEE, J.; KIM, Y. Speed-adaptive ratio-based lane detection algorithm for self-driving vehicles. In: INTERNATIONAL SOC DESIGN CONFERENCE (ISODC), 1., 2016, Seoul. **Anais...** [S.l.]: IEEE, 2016. p.269–270.
- KIM, Z. Robust lane detection and tracking in challenging scenarios. **IEEE Transactions on Intelligent Transportation Systems**, Los Angeles, v.9, n.1, p.16–26, Mar. 2008.
- KLUGE, K.; LAKSHMANAN, S. A deformable-template approach to lane detection. In: PROCEEDINGS OF THE INTELLIGENT VEHICLES '95. SYMPOSIUM, 1., 1995, Detroit. **Anais...** [S.l.]: IEEE, 1995. v.2110, p.54–59.
- KOMPETENZENTRUM. **Safe Small Electric Vehicles through Advanced Simulation Methodologies**. Disponível em: <http://www.project-safeev.eu/pdf/2013_06_30_safeev_d1-2_under_approval.pdf>. Acesso em: 11 ju. 2016.
- KOWALENKO, K. **Keeping Cars From Crashing**. Disponível em: <<http://theinstitute.ieee.org/technology-focus/technology-topic/keeping-cars-from-crashing725>>. Acesso em: 30 Jun. 2016.
- LEE, Y.; KIM, H. Real-time lane detection and departure warning system on embedded platform. In: INTERNATIONAL CONFERENCE ON CONSUMER ELECTRONICS, 6., 2016, Berlin. **Anais...** [S.l.]: IEEE, 2016. p.1–4.
- LEI, Y.; WONG, K. C. Ellipse detection based on symmetry. **Pattern Recognition Letters**, [S.l.], v.20, n.1, p.41–47, Jan. 1999.
- LEXUS. **Japanese market Lexus LS460 boasts a host of world's firsts**. Disponível em: <<http://www.autoblog.com/2006/09/19/japanese-market-lexus-ls460-boasts-a-host-of-worlds-firsts/>>. Acesso em: 11 jul. 2016.
- LI, H.; FENG, M.; WANG, X. Inverse perspective mapping based Urban road markings detection. In: IEEE INTERNATIONAL CONFERENCE ON CLOUD COMPUTING AND INTELLIGENCE SYSTEMS, 2., 2012, Hangzho. **Anais...** [S.l.]: IEEE, 2012. v.3, p.1178–1182.
- LI, Y. et al. Nighttime lane markings recognition based on Canny detection and Hough transform. In: IEEE INTERNATIONAL CONFERENCE ON REAL-TIME COMPUTING AND ROBOTICS (RCAR), 1., 2016, Combodia. **Anais...** [S.l.]: IEEE, 2016. p.411–415.

- LI, Z. Q.; MA, H. M.; LIU, Z. Y. Road Lane Detection with Gabor Filters. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEM AND ARTIFICIAL INTELLIGENCE (ISAI), 1., 2016, Hong Kong. **Anais...** [S.l.]: IEEE, 2016. p.436–440.
- LIM, K. H. et al. River Flow Lane Detection and Kalman Filtering-Based B-Spline Lane Tracking. **International Journal of Vehicular Technology**, [S.l.], v.12, p.1–10, Nov. 2012.
- LIN, C. C.; WANG, M. S. A vision based top-view transformation model for a vehicle parking assistant. **Sensors**, [S.l.], v.12, n.4, p.4431–4446, Dez. 2012.
- LIU, G.; WÖRGÖTTER, F.; MARKELIĆ, I. Combining Statistical Hough Transform and Particle Filter for robust lane detection and tracking. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 1., 2010, San Diego. **Anais...** [S.l.]: IEEE, 2010. p.993–997.
- LOPEZ, A. et al. Detection of lane markings based on ridgeness and RANSAC. In: IEEE INTELLIGENT TRANSPORTATION SYSTEMS, 1., 2005, Vienna. **Proceedings...** [S.l.]: IEEE, 2005. p.254–259.
- MACHADO, A. **Perspectiva**. 1st.ed. São Paulo, SP, Brasil: Mcgraw Hill, 1983.
- MASSY, K. **Good vibrations for new BMW drivers**. Disponível em: <<http://www.cnet.com/news/good-vibrations-for-new-bmw-drivers/>>. Acesso em: 11 jul. 2016.
- MENDEL, J. M.; SHYNK, J. **Lessons in Estimation Theory for Signal Processing, Communications, and Control**. 1st.ed. London, UK: Pearson Education, 1995.
- MERCEDES. **Lactive Lane Keeping Assist**. Disponível em: <<https://www.mbusa.com/mercedes/technology/videos/detail/title-safety/videoId-e84b9423c67a7410VgnVCM100000ccec1e35RCRD>>. Acesso em: 10 jul. 2016.
- MIGLIACCI, P. **Tesla e Google seguem rumos diferentes com carros autônomos?** Disponível em: <<http://www1.folha.uol.com.br/tec/2016/07/1788653-tesla-e-google-seguem-rumos-diferentes-com-carros-autonomos.shtml>>. Acesso em: 21 set. 2016.
- MOTAVALLI, J. **3D Laser Wars - Slashing Price and Size for Self-Driving Cars**. Disponível em: <<http://www.automotiveitnews.org/articles/719476/3d>>. Acesso em: 21 dez. 2016.
- NISSAN. **Safety Activities**. Disponível em: <http://www.nissan-global.com/EN/DOCUMENT/PDF/TECHNOLOGY/safety_activities/safety_activities_E.pdf>. Acesso em: 10 jul. 2016.
- OMS. **Década de Ação pela Segurança no Trânsito 2011-2020 é lançada oficialmente hoje em todo o mundo**. Disponível em: <<https://nacoesunidas.org/decada-de-acao-pela-seguranca-no-transito-2011-2020-e-lancada-oficialmente-hoje-11-em-todo-o-mundo/>>. Acesso em: 30 jun. 2016.
- ONSV. **90% dos acidentes são causados por falhas humanas, alerta ONSV**. Disponível em: <<http://www.onsv.org.br/noticias/90-dos-acidentes-sao-causados-por-falhas-humanas-alerta-observatorio/>>. Acesso em: 1 jul. 2016.

- OPENCV. **Adaptive Thresholding**. Disponível em: <<http://docs.opencv.org/trunk/d7/d4d/tutorial-py-thresholding.html>>. Acesso em: 29 nov. 2016.
- PAETZOLD, F.; FRANKE, U. Road recognition in urban environment. **Image and Vision Computing**, [S.l.], v.18, n.5, p.377–387, Set. 2000.
- PAULA, M. B. d.; JUNG, C. R. Real-Time Detection and Classification of Road Lane Markings. In: CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES, 15., 2013, Arequipa. **Anais...** [S.l.]: IEEE, 2013. p.83–90.
- PAULA, M. B. de. **Index of / mbpaula/roadLaneMarkings/datasetVideos**. Disponível em: <<http://inf.ufrgs.br/mbpaula/roadLaneMarkings/datasetVideos/>>. Acesso em: 18 jan. 2017.
- PEDRINI, H.; SCHWARTZ, W. R. **Análise de Imagens Digitais: princípios, algoritmos e aplicações**. 1st.ed. São Paulo: Cengage Learning, 2007.
- PINHONI, M. **Comportamento do motorista causa maioria dos acidentes**. Disponível em: <<http://veja.abril.com.br/noticia/brasil/comportamento-do-motorista-causa-maioria-dos-acidentes>>. Acesso em: 1 jul. 2016.
- PRINCE, S. J. **Computer vision: models, learning and inference**. Disponível em: <<http://slideplayer.com/slide/2310891/>>. Acesso em: 21 nov. 2016.
- ROUSSEEUW, P. J.; LEROY, A. M. **Robust Regression and Outlier Detection**. 1st.ed. New York, NY, USA: John Wiley & Sons, Inc., 1987.
- SANCHES, C. H. et al. **Técnicas de Suavização de Imagens e Eliminação de Ruídos**. Disponível em: <<http://eati.info/eati/2015/assets/anais/Longos/L2.pdf>>. Acesso em: 25 jan. 2016.
- SANTOS, M. C. dos. **Revisão de Conceitos em Projeção, Homografia, Calibração de Câmera, Geometria Epipolar, Mapas de Profundidade e Varredura de Planos**. Disponível em: <<http://www.ic.unicamp.br/~rocha/teaching/2012s1/mc949/aulas/additional-material-revision-of-concepts-homography-and-related-topics.pdf>>. Acesso em: 20 jan. 2017.
- SAUDI, A. et al. Fast lane detection with Randomized Hough Transform. In: INTERNATIONAL SYMPOSIUM ON INFORMATION TECHNOLOGY, 1., 2008, Kuala Lumpur. **Anais...** IEEE, 2008. v.4, p.1–5.
- SHOPSEG. **Sensor CCD, CMOS ou EXMOR? Qual é o sensor ideal? A Shopseg responde**. Disponível em: <<http://blog.shopsegribeirao.com.br/2016/02/sensor-ccd-cmos-ou-exmor-qual-e-o.html>>. Acesso em: 20 jan. 2017.
- SONG, M.; LIU, X. Research in video detection of lane curve and its application in speed alert system. **Research Journal of Applied Sciences, Engineering and Technology**, [S.l.], v.7, n.5, p.957–962, Fev. 2014.
- STASZEWSKI, R.; ESTL, H. **Making cars safer through technology innovation**. Disponível em: <<http://www.ti.com/lit/wp/sszy009/sszy009.pdf>>. Acesso em: 21 jan. 2016.

SULEMAN, K. **Intel paves the road for BMW's iNEXT autonomous cars in 2021.** Disponível em: <<http://www.intel.co.uk/content/www/uk/en/it-managers/autonomous-cars.html>>. Acesso em: 21 dez. 2016.

SYSTEM, C. A. **Collision avoidance system.** Disponível em: <<http://109.235.148.119/collisionavoidancesystem/2016/01/24/lane-departure-warning-system/f>>. Acesso em: 11 Jul. 2016.

TESCHL, G. **Ordinary differential equations and dynamical systems.** 1st.ed. Providence, Rhode Island, EUA: American Mathematical Society, 2012.

THEGUARDIAN. **Google self-driving car caught on video colliding with bu.** Disponível em: <<https://www.theguardian.com/technology/2016/mar/09/google-self-driving-car-crash-video-accident-bus>>. Acesso em: 12 jul. 2017.

THOME, A. G. **Aquisição e Representação da Imagem Digital.** Disponível em: <http://equipe.nce.ufrj.br/thome/p_grad/nn_img/transp/c2_aquis_v2.pdf>. Acesso em: 11 out. 2016.

TJ. **Lane Departure Warning System.** Disponível em: <<http://usedsemitrailers.com/lane-departure-warning-system/>>. Acesso em: 10 jul 2016.

TRW. **Sensor Fact Sheet.** Disponível em: <http://www.trw.com.br/sites/default/files/Sensor%20fact%20Sheet_0.pdf>. Acesso em: 11 jul. 2016.

TSUGAWA, S. Issues and Recent Trends in Vehicle Safety Communication Systems. **IATSS Research**, [S.l.], v.29, n.1, p.7–15, Fev. 2005.

VOLKSWAGEN. **Lane Assist.** Disponível em: <<http://www.volkswagen.co.uk/technology/proximity-sensing/lane-assist>>, Acesso em: 21 dez. 2016.

VOLKSWAGEN. **Lane Assist.** Disponível em: <<http://en.volkswagen.com/en/innovation-and-technology/technical-glossary/spurhalteassistentlaneassist.html>>. Acesso em: 10 jul. 2017.

VOUSDEN, M. **What is lane assist?.** Disponível em: <<https://www.carwow.co.uk/blog/lane-assist-explaine>>. Acesso em: 10 jul. 2016.

WADHWA, J.; KALRA, G.; KRANTHI, B. Real Time Lane Detection in Autonomous Vehicles Using Image Processing. **Research Journal of Applied Sciences, Engineering and Technology**, Kuala Lumpur, v.11, n.4, p.429–433, Maio 2015.

WANG, J. et al. An Approach of Lane Detection Based on Inverse Perspective Mapping. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 17., 2014, Qingdao. **Anais...** [S.l.]: IEEE, 2014. p.35–38.

WHATWHENHOW. **What-when-how:** in depth tutorials and information. Disponível em: <<http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/>>. Acesso em: 7 dez. 2016.

WOOLLASTON, V. **What the brain of Google's self-driving car sees: the 'terminator's-eye-view' that shows just what it takes to navigate a city.** Disponível em: <<http://www.dailymail.co.uk/sciencetech/article-2317594/New-images-Google-self-driving-cars-world.html>>. Acesso em: 11 out. 2016.

WORLD HEALTH ORGANIZATION. **Global status on road report 2015: summary.** Geneva: , 2015. 14 p. Disponível em: <http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/>. Acesso em: 27 Jan. 2017.

YING, Z.; LI, G. Robust lane marking detection using boundary-based inverse perspective mapping. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING (ICASSP), 1., 2016, Shangai. **Anais...** [S.l.]: IEEE, 2016. p.1921–1925.

YU, H. et al. Vision-Based Lane Marking Detection and Moving Vehicle Detection. In: INTERNATIONAL CONFERENCE ON INTELLIGENT HUMAN-MACHINE SYSTEMS AND CYBERNETICS (IHMSC), 8., 2016, Hangzhou. **Anais...** [S.l.]: IEEE, 2016. v.2, p.574–577.

ZHANG, F. et al. A lane marking extraction approach based on Random Finite Set Statistics. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 4., 2013, Gold Coast. **Anais...** [S.l.]: IEEE, 2013. p.10–11.

ZHOU, S. et al. A novel lane detection based on geometrical model and Gabor filter. In: IEEE INTELLIGENT VEHICLES SYMPOSIUM, 4., 2010, San Diego. **Anais...** IEEE, 2010. p.59–64.

ZULIANI, M. **Ransac for Dummies.** Disponível em: <<http://www.ic.unicamp.br/rocha/teaching/2012s1/mc949/aulas/ransac-4-dummies.pdf>>. Acesso em: 24 jan. 2016.