

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA**

Thiago Rateke

**DETECÇÃO DE CAMINHO EM TEMPO REAL PARA VEÍCULO
AUTÔNOMO UTILIZANDO VISÃO PASSIVA**

Florianópolis

2015

Thiago Rateke

**DETECÇÃO DE CAMINHO EM TEMPO REAL PARA VEÍCULO
AUTÔNOMO UTILIZANDO VISÃO PASSIVA**

Dissertação submetida ao Programa de Pós-
Graduação em Ciência da Computação para
a obtenção do Grau de Mestre.

Orientador: Prof. Aldo von Wangenheim,
Dr. rer. nat.

Florianópolis

2015

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Rateke, Thiago
Detecção de Caminho em Tempo Real para Veículo Autônomo
Utilizando Visão Passiva / Thiago Rateke ; orientador,
Aldo von Wangenheim - Florianópolis, SC, 2015.
121 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico. Programa de Pós-Graduação em
Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Detecção de Caminho. 3.
Segmentação de Imagem. 4. Visão Computacional. 5. Navegação
Autônoma. I. von Wangenheim, Aldo. II. Universidade Federal
de Santa Catarina. Programa de Pós-Graduação em Ciência da
Computação. III. Título.

Thiago Rateke

**Detecção de Caminho em Tempo Real para Veículo Autônomo
Utilizando Visão Passiva**

Esta Dissertação foi julgada adequada para obtenção do Título de “Mestre”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação

Florianópolis, 5 de março de 2015.

Prof. Ronaldo dos Santos Mello, Dr.
Coordenador do Programa

Prof. Aldo von Wangenheim, Dr. rer. nat.
Orientador

Banca Examinadora:

Prof. Eros Comunello, Dr. rer. nat.
Universidade do Vale do Itajaí

Prof. Mario Antonio Ribeiro Dantas, Dr.
Universidade Federal de Santa Catarina

Prof. Roberto Willrich, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho aos meus pais, Gilberto Rateke e Maria Helena Lemos Rateke. E a minha namorada, Karla Aparecida Justen.

AGRADECIMENTOS

À Karla Aparecida Justen. Não é fácil abdicar de momentos de lazer e descanso, perder noites de sono e se dedicar à um trabalho onde muitas vezes o resultado aparentava não ser possível de se alcançar dentro do prazo. Pelo apoio, muito além de incentivo, mas participação ativa no desenvolvimento do trabalho que resultou nesta dissertação, agradeço a minha namorada, amiga, consciência, anjo da guarda e, principalmente neste caso, minha parceira neste trabalho. Devo ao seu empenho e dedicação: muitas ideias e soluções criativas; diversas revisões; melhorias na parte de hardware (veículo, câmera, controle e transmissão do vídeo); desvendar o Arduino; documentação em relatórios do trabalho; idealização, desenho e, literalmente, construção do circuito de testes. A cada dia que passa confirmando e aumentando a admiração que sinto.

Aos meus pais, Gilberto Rateke e Maria Helena Lemos Rateke, pela criação e apoio que continuam dando a cada etapa de minha vida, se hoje tenho alguma característica de pensamento livre, curiosidade e questionamentos, desejo de encarar desafios e também não desistir perante as dificuldades, devo a educação que investiram e me deram. Além claro, de compreenderem minha ausência em muitos momentos e facilitar sempre que possível e no que poderiam para que este trabalho pudesse ser desenvolvido. Um dos maiores prêmios em cada objetivo que concluo é saber que estou os deixando mais felizes e orgulhosos.

À minha família, Simone Cristina Rateke, Andrus da Silva e Natália Rateke da Silva; Gilberto Rateke Jr., Samira Moratti e Gilberto Rateke Neto; Fabiano Rateke, Simone Hansen Rateke, Leonardo Hansen Rateke e Lucas Hansen Rateke. Pelo apoio, e por saber que posso contar com vocês quando preciso.

À família da minha namorada pelo apoio e por compreenderem sua ausência e dedicação. À Karina Aparecida Justen e Karen Aparecida Justen pela ajuda nos momentos de testes, a Maria Aparecida Florinda Sales por todo apoio e também por ajudar e facilitar sempre que necessário, e ao Edson Nazareno Justen pelo grande apoio na parte de hardware (construção da plataforma de controle, adaptações no veículo) e também pelo circuito de testes.

Ao meu orientador, Aldo von Wangenheim, pelas oportunidades e orientação durante o período desta dissertação.

Ao Eros Comunello e ao Antonio C. Sobieranski, pela ajuda com o artigo e sugestões e incentivos relacionados ao projeto.

Aos colegas do Lapix pelas discussões, sugestões, revisões e apoio.

Aos meus amigos, por compreenderem minha ausência e me apoiarem.

Reunir-se é um começo, permanecer juntos é
um progresso e trabalhar juntos é um sucesso.
(Henry Ford)

RESUMO

Os veículos que trafegam de forma autônoma necessitam identificar ao longo do seu caminho vários elementos tais como: a estrada a seguir, se há necessidade de desviar de obstáculos fixos ou móveis e até mesmo parar. Esta dissertação tem como objetivo desenvolver uma metodologia para controlar a trajetória de um pequeno veículo através do seu sistema de rádio controle e equipado com câmera, utilizando técnicas da Visão Computacional e Processamento Digital de Imagens. Pode-se observar que diversos fatores impactam o controle do veículo pelo fato de estar em constante movimento. As mudanças constantes na iluminação natural, no tipo e na qualidade do terreno e a cor do cenário a ser trafegado, levaram a utilizar uma técnica adaptativa em tempo real nos algoritmos de segmentação para identificar a estrada e manter o veículo no caminho a ser seguido. Um pequeno circuito de testes em campo foi construído, para avaliar a metodologia desenvolvida simulando algumas situações reais de funcionamento. Os resultados dos testes indicaram que a metodologia desenvolvida foi capaz de manter autonomamente o veículo no caminho simulado, com diferentes tipos de terreno e em condições de variabilidade na iluminação natural.

Palavras-chave: Detecção de Caminho. Segmentação de Imagem. Visão Computacional.

ABSTRACT

The autonomous vehicles need to identify several elements such as the road to travel, deviate from fixed or mobile obstacles and even stop. This thesis aims to develop a methodology to control the trajectory of a small vehicle by radio control and equipped with camera, using techniques of Computer Vision and Digital Image Processing. It can be observed that several factors impact the control of the vehicle by the fact of being in constant motion. The constant changes in natural lighting, the type and quality of the land and the pixels color of the scene being trafficked, led to use an adaptive technique in real time on segmentation algorithms to identify the road and keep the vehicle on the way forward. A small testing circuit was built to evaluate the developed methodology simulating some real situations of operation. The test results indicated that this methodology was able to autonomously maintain the vehicle in the road, with different types of terrain and variability in natural daylight conditions.

Keywords: Path Detection. Image Segmentation. Computer Vision.

LISTA DE FIGURAS

Figura 1	Detecção de Linhas (adaptado de (NEVES; PELAES, 2001)).	34
Figura 2	Resultados de (GUO; MITA, 2009b).	39
Figura 3	Resultados de (GUO; MITA; MCALLESTER, 2012).	40
Figura 4	Resultados de (BROGGI; CATTANI, 2006).	40
Figura 5	Resultados de (DANESCU; NEDEVSCI, 2009) Esquerda resultado do artigo, direita comparação com <i>Kalman Filter</i> .	41
Figura 6	Resultados de (DORNAIKA et al., 2011).	42
Figura 7	Resultados de (CHEN; TAI, 2010). Na primeira coluna as imagens originais. Segunda coluna imagem segmentada. Na terceira coluna, ajuste de iluminação. Na quarta coluna, segmentação com base na imagem original e na imagem com ajuste de iluminação.	43
Figura 8	Resultados de (ÁLVAREZ; GEVERS; LÓPEZ, 2009).	44
Figura 9	Resultados de (ÁLVAREZ; LÓPEZ, 2011).	44
Figura 10	Resultados de (KONG; AUDIBERT; PONCE, 2010).	45
Figura 11	Resultados de (ZHOU et al., 2010). Em (a), as imagens originais e os pontos de amostra. Em (b), classificação dos resultados, vermelho estrada e amarelo não-estrada. Em (c), resultados de estrada rotulados em tom de verde.	45
Figura 12	Resultados de (WEDEL et al., 2009).	46
Figura 13	Resultados de (SON; MITA; TAKEUCHI, 2008).	46
Figura 14	Resultados de (LEE; LEE; LEE, 2009).	47
Figura 15	Resultados de (KIM; SON; SOHN, 2011).	47
Figura 16	Resultados de (BAI; WANG, 2011).	48
Figura 17	Resultados de (BAI; WANG; FAIRHURST, 2008).	49
Figura 18	Resultados de (MARZOTTO et al., 2010).	50
Figura 19	Resultados de (MATSUSHITA; MIURA, 2011).	51
Figura 20	Resultados de (OBRADOVIC et al., 2013).	52
Figura 21	Resultados de (WANG; DAHNOUN; ACHIM, 2012). Onde na primeira linha estão os resultados do mapa de características proposto e na segunda linha estão os resultados dos mapas de gradientes tradicionais.	53
Figura 22	Resultados de (ZHAO; WANG; YAN, 2012).	53
Figura 23	Resultados de (AMDITIS et al., 2010).	54

Figura 24 Resultados de (GOPALAN et al., 2012).	55
Figura 25 Resultados de (GUO; MITA, 2009a).	56
Figura 26 Resultados de (KODAGODA; WIJESOMA; BALASURIYA, 2006).	56
Figura 27 Resultados de (LUNDQUIST; ORGUNER; GUSTAFSSON, 2011).	57
Figura 28 Resultados de (ONIGA; NEDEVSCHI, 2010). (a) Saída do classificador baseado na superfície de estrada. (b) Saída do classificador baseado em densidade. (c) Resultado após fusão dos dois classificadores e da filtragem de erro.	58
Figura 29 Resultados de (WANG; BAI; FAIRHURST, 2008).	59
Figura 30 Resultados de (WU; LIN; CHEN, 2009).	60
Figura 31 Etapas de desenvolvimento da Proposta.	72
Figura 32 (a) - imagem original. (b) - imagem suavizada. (c) - imagem em ton de cinza. (d) imagem de bordas.	75
Figura 33 (a) - imagem original. (b) - imagem com a detecção de caminho.	77
Figura 34 Regiões de interesse para o controle.	78
Figura 35 Imagem com as informações de controle.	78
Figura 36 Imagem com as informações de controle.	80
Figura 37 Comparação entre a estrutura inicial e a posteriormente desenvolvida para a comunicação entre o servo e o potenciômetro.	81
Figura 38 Marcação para teste de Aceleração.	82
Figura 39 Gráfico de comparação entre teste: voltas/10 segundos de cada ângulo.	83
Figura 40 Barra de valores representando as reações causadas por cada ângulo.	84
Figura 41 Intervalo de ângulos zeros nos testes feitos.	84
Figura 42 Estrutura de filmagem para o teste de direcionamento.	85
Figura 43 Algumas das imagens resultantes de um teste realizado e apresentação do tempo de acordo com o ângulo do servo.	86
Figura 44 Detecção dos ângulos de direção - primeira etapa.	87
Figura 45 Detecção dos ângulos de direção - segunda etapa.	87
Figura 46 Detecção dos ângulos de direção - terceira etapa.	88
Figura 47 Comparação entre o primeiro e o último teste feito, relacionando a direção (em graus) com cada um dos 180 ângulos do servo motor.	89

Figura 48 Controle remoto Spektrum DX6i.	90
Figura 49 DSM2 DualLink AR6200.	90
Figura 50 Apresentação das alterações feitas no circuito do Controle Re- moto DX6i.	91
Figura 51 Detalhamento do item 4 do Ciclo do Projeto.	92
Figura 52 Passos da transmissão de vídeo.	93
Figura 53 Veículo para experimentos e avaliação do trabalho.	95
Figura 54 Protótipo do Circuito.	96
Figura 55 (a) Asfalto. (b) Terra. (c) Lajota. (d) Intersecção asfalto e terra. (e) Intersecção terra e lajota. (f) Intersecção lajota e asfalto.	97
Figura 56 Detecção do caminho no trecho com lajota.	107
Figura 57 Detecção do caminho com luz solar forte.	107

LISTA DE TABELAS

Tabela 1	Tabela Comparativa - Detecção de Caminho	61
Tabela 2	Requisitos Não-Funcionais	70
Tabela 3	Requisitos Funcionais	71
Tabela 4	Resultados - Trecho 1	101
Tabela 5	Resultados - Trecho 2	102
Tabela 6	Resultados - Trecho 3	103
Tabela 7	Resultados - Trecho 4	104
Tabela 8	Resultados - Trecho 5	105
Tabela 9	Resultados - Trecho 6	106

LISTA DE ABREVIATURAS E SIGLAS

PDI	Processamento Digital de Imagens	25
VC	Visão Computacional	25
LIDAR	<i>Light Detection and Ranging</i>	26
GPS	<i>global positioning system</i>	31
DARPA	<i>Defense Advanced Research Projects Agency</i>	31
MRF	<i>Markov Random Field</i>	38
RANSAC	<i>Catmull-Rom splines based on RANdom SAmple Consensus</i>	38
FPGA	<i>Field Programmable Gate Array</i>	49
LDW	<i>Lane Departure Warning</i>	49
GSPF	<i>Gaussian Sum Particle Filter</i>	52
SIR	<i>Sequential Importance Resampling</i>	52
GPF	<i>Gaussian Particle Filter</i>	52
CuTE	<i>Curb Tracking and Estimation</i>	56
DEM	<i>Digital Elevation Map</i>	57
VP	<i>Vanishing Point</i>	58
SVM	<i>Support Vector Machines</i>	62
ROI	<i>Region of Interest</i>	77

SUMÁRIO

1	INTRODUÇÃO	25
1.1	CONTEXTUALIZAÇÃO DO PROBLEMA	26
1.2	OBJETIVOS	27
1.2.1	Objetivo Geral	27
1.2.2	Objetivos Específicos	27
1.3	JUSTIFICATIVA	27
1.3.1	Cenários de Uso	28
1.4	MATERIAIS E MÉTODOS	28
1.5	ESTRUTURA DA DISSERTAÇÃO	29
2	FUNDAMENTAÇÃO TEÓRICA	31
2.1	DETECÇÃO DE CAMINHO	32
2.2	SEGMENTAÇÃO DE IMAGENS	33
2.2.1	Deteção de Descontinuidades	34
2.2.2	Deteção de Similaridade	35
2.3	REVISÃO BIBLIOGRÁFICA	36
2.3.1	Revisão Sistemática - Deteção de Caminho	37
2.3.1.1	Trabalhos Correlatos - Deteção de Caminho	38
2.3.1.2	Comparativo Entre Trabalhos Correlatos	60
2.3.1.3	Considerações Sobre os Trabalhos Correlatos e Proposta	67
3	PROPOSTA	69
3.1	REQUISITOS	69
3.1.1	Requisitos Não-Funcionais	70
3.1.2	Requisitos Funcionais	71
3.2	ETAPAS DE DESENVOLVIMENTO DA PROPOSTA	72
3.3	CÓDIGO - PROCESSAMENTO DOS DADOS	73
3.3.1	Deteção do Caminho - Segmentação da Imagem	73
3.3.2	Análise da imagem - Controle do Veículo	77
3.4	TRANSMISSÃO DOS RESULTADOS PARA O RÁDIO CONTROLADO	79
3.4.1	Transmissão de Dados do Notebook ao Arduino	79
3.4.2	Estrutura de Transmissão do Arduino ao RC	80
3.5	CONTROLE REMOTO	89
3.5.1	Rádio Controle	90
3.6	VEÍCULO	91
3.7	TRANSMISSÃO DE VÍDEO	92
4	ESTRATÉGIA DE AVALIAÇÃO	95
5	RESULTADOS	99

6	CONCLUSÃO	109
6.1	TRABALHOS FUTUROS	110
	REFERÊNCIAS	113

1 INTRODUÇÃO

Processamento Digital de Imagens (PDI) e Visão Computacional (VC) são técnicas que têm cada vez mais aplicação em diversas atividades e segmentos da sociedade, contribuindo de forma considerável na coleta, interpretação e análise dos dados em imagens. Apesar de serem trabalhadas muitas vezes juntas, é necessário discernir seus conceitos. PDI refere-se a um processo, baseado em imagens, operações com intuito de gerar um resultado para interpretação do usuário. Já Visão Computacional, é a análise de imagens por um equipamento eletrônico com o objetivo de realizar uma interpretação, ou seja, simulando a visão humana. Aplicações que utilizam técnicas de PDI e VC são desenvolvidas em áreas como: sensoriamento remoto, astronomia, área militar, esportes, segurança, biomedicina, robótica (PEDRINI; SCHWARTZ, 2007).

No âmbito esportivo é executado em tarefas como detecção e rastreamento de atletas, permitindo calcular a distância percorrida, velocidade média dos atletas, por exemplo. Já no ramo da segurança, pode-se realizar o reconhecimento de padrões, como: um ambiente onde as pessoas estejam abaixadas ou com as mãos levantadas pode sinalizar um assalto. Ainda na segurança, há também a biometria, onde as aplicações ocorrem no contexto de análise de identidade ao aplicar o reconhecimento facial, a identificação de impressões digitais, a verificação da veracidade das assinaturas (PEDRINI; SCHWARTZ, 2007). Já na medicina, a análise e interpretação de imagens (raio-x, ressonância magnética, ultrassonografia, tomografia computadorizada, microscópios ópticos ou eletrônicos) auxiliam e aumentam a precisão do diagnóstico na identificação de lesões; regiões afetadas por câncer; má formações; proporção de células sanguíneas; etc (FILHO; NETO, 1999). A aplicação de PDI e VC no campo da robótica é cada vez mais frequente, como no controle de qualidade, na montagem e inspeção de produtos, compreensão da superfície terrestre, entre outros (PEDRINI; SCHWARTZ, 2007).

Análises podem ser feitas com base nas imagens do trânsito das estradas para a identificação de placas de automóveis, verificação da velocidade dos veículos em uma determinada faixa. Além de também ser aplicado para automatizar veículos. Usando PDI e VC é possível detectar o caminho que deve ser percorrido e os obstáculos que apareçam; sendo que o ideal é que a detecção de caminho ocorra mesmo com alterações nas características do caminho, ou seja, não importando se é asfalto, terra ou paralelepípedos (PEDRINI; SCHWARTZ, 2007).

1.1 CONTEXTUALIZAÇÃO DO PROBLEMA

A navegação autônoma de veículos já tem destaque há alguns anos e diversos estudos e pesquisas são feitos com este enfoque. Mas ainda é uma área desafiadora, pois, como o veículo autônomo pode saber se um terreno é mesmo seguro para ir adiante? Se há uma mudança nos padrões do terreno, como o veículo pode ter certeza de seguir adiante sem que precise de uma interferência manual? Outra etapa importante na navegação autônoma é a detecção de obstáculos, evitando assim acidentes ou colisões.

O objetivo de um sistema de detecção de estrada baseado em VC, é possibilitar a detecção da estrada (caminho) com uma fonte de captura que esteja na frente do veículo. Um sistema de detecção baseado em VC é bastante desafiador, pois lida com cenários abertos e com o veículo em constante movimento, ocasionando assim problemas de iluminação, mudança constante do fundo da cena (background), presença de diferentes objetos (veículos, pessoas, etc.), diferentes cenários e tipos de terrenos (urbanos, rodovias, estradas de terra) (ÁLVAREZ; GEVERS; LÓPEZ, 2009).

Guo, Mita e McAllester (2012) vem embasar a dificuldade na detecção de caminho pela VC, justamente pelo grande número de fatores (tipos de terreno, iluminação, condições de tempo e clima) que não são facilmente medidos e que influenciam consideravelmente e tem um grande impacto na detecção de estrada.

Atualmente os casos de grande destaque na navegação autônoma utilizam diversos sensores, não somente câmera (URMSON et al., 2008), (URMSON, 2014b), (URMSON, 2014a) e (FERNANDES et al., 2014). Sendo, que um desses sensores, e talvez o principal, é o LIDAR (*Light Detection and Ranging*), que é um laser óptico utilizado para detecção remota de propriedades da luz refletida, de modo a medir distâncias entre o sensor e o objeto alvo (LIDAR, 2015). Nos projetos citados o LIDAR é considerado Class1, que é a categoria de menor impacto e com comprimentos de onda de 905nm. Em (GROUP SAFETY PUBLICATION. INTERNATIONAL STANDARD, 2001) e (STANDARD, 2005) é dito que lasers Class1 quando não tem contato direto com o olho humano durante um período mais extenso não representam perigo à retina. Entretanto, em (GROUP SAFETY PUBLICATION. INTERNATIONAL STANDARD, 2001) é apresentado uma tabela com a categorização dos lasers e os possíveis riscos com o excesso de exposição à cada um dos níveis. Os lasers com comprimento de onda entre 780nm e 1400nm, que é a faixa de abrangência do LIDAR, podem causar catarata e queimar a retina. Assim, caso os veículos autônomos sejam uma realidade no dia a dia civil, LIDAR pode não ser uma solução segura para uma grande quantidade de veículos, devido ao excesso de exposição sem proteção.

A visão passiva pode ser uma alternativa nesse cenário futuro e pesquisas realizadas com este enfoque, utilizando somente a informação da câmera (imagem capturada) são feitas à bastante tempo, porém são normalmente testadas somente em vídeos gravados, onde um algoritmo de detecção de caminho é forçado a se adaptar, pois o veículo continua seu trajeto independente da detecção falhar em momentos de alteração do terreno ou problemas de iluminação.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Controlar o veículo de forma autônoma com base nos resultados da detecção de caminho, com métodos de visão computacional e processamento digital de imagens.

1.2.2 Objetivos Específicos

- Detectar caminho em tempo real e de forma adaptativa, ou seja, continuar detectando o caminho independente das mudanças nas características do terreno (asfalto, terra e lajotas). Onde o cenário de captura seja em imagens frontais e horizontais aos veículos, ou imagens que simulem estar numa posição frontal no veículo.
- Controlar o veículo em tempo real, sendo que as decisões devem ser tomadas com base nos resultados da detecção de caminho;
- Avaliar a detecção de caminho e o controle do veículo em um circuito preparado com alterações nas características do caminho.

1.3 JUSTIFICATIVA

Levando em consideração que a navegação autônoma está cada vez mais próxima de tornar-se uma realidade, e que os trabalhos atuais de maior sucesso utilizam sensores a laser, que em um futuro onde a navegação autônoma seja utilizada em grande escala, pode ser prejudicial a saúde humana. Desta forma, esta dissertação pretende verificar que a visão passiva (somente câmeras) seja o suficiente para a realização da navegação autônoma.

Para isso foram utilizadas técnicas de Visão Computacional para a

detecção de caminho de forma adaptativa e em tempo real. Com base nos resultados obtidos dessa detecção de caminho, é feito o controle de direção e aceleração do veículo, sem interferências manuais.

Para realizar a detecção de caminho, métodos como segmentação de imagem e detecção de bordas são utilizados, definindo desta forma, a parte da imagem que é considerada um caminho que pode ser seguido, e separando do restante da imagem. E isso deve ocorrer em tempo real e de forma adaptativa, ou seja, não importando as mudanças no caminho (cor, tipo de terreno, etc.) a detecção deve continuar acontecendo.

O controle do protótipo é feito com base nos resultados obtidos das imagens de detecção do caminho, e então os valores corretos de direção e aceleração são enviados ao veículo, possibilitando dessa forma sua navegação autônoma e também a avaliação do trabalho.

1.3.1 Cenários de Uso

- **Ambiente Urbano:** Esse veículo deve ser capaz de cumprir uma rota em ambiente urbano. Dentro dessa rota, o veículo deve conseguir identificar um caminho; e isso é possível através de técnicas de processamento de imagens onde após a captura por uma câmera acoplada na parte frontal do veículo torna-se possível identificar o que é um caminho, por exemplo, se estiver andando em uma rodovia, o carro deve identificar a pista e as faixas da pista para se manter em sua pista. E deve manter-se em seu caminho, mesmo que o padrão da rodovia ou do trajeto por onde o veículo esteja passando mude consideravelmente (são considerados as informações de cor no momento da captura da imagem).
- **Ambiente Rural:** O veículo também deve ser capaz de cumprir uma rota com alterações mais bruscas no padrão do terreno, como por exemplo de asfalto para terra, e mesmo assim continuar a detecção (adaptativa em tempo real). E mesmo dentro do cenário rural, o padrão de terreno varia muito, gramado, terra com diferentes cores. Um trajeto deve ser definido e o veículo deve cumprir esse trajeto, independente dessas mudanças de cor do caminho.

1.4 MATERIAIS E MÉTODOS

Para a detecção do caminho, algoritmos de detecção de bordas e segmentação de imagem serão utilizados. Define-se uma borda como a fronteira

entre duas regiões, onde nessas regiões há uma descontinuidade nos níveis de cinza. Bordas podem ser tanto de textura como de luminosidade (PEDRINI; SCHWARTZ, 2007). A Segmentação de Imagem é comumente fundamentada em homogeneidade (características em comum que os pixels compartilham) do sinal e continuidade (combinação dos padrões de homogeneidade ao longo do domínio espacial da imagem) (SOBIERANSKI, 2010). A segmentação de imagem pode seguir duas propriedades, similaridade e descontinuidade (NEVES; PELAES, 2001). A detecção de bordas é uma técnica de segmentação, mais especificamente segmentação por descontinuidade.

Será utilizado, mais especificamente, algoritmos de segmentação por Crescimento de Regiões (*Region Growing*). A segmentação por Crescimento de Regiões agrupa *pixels* ou sub-regiões em regiões maiores. O crescimento de regiões inicia em relação às coordenadas de entrada sobre uma imagem $I(x,y)$ ou em relação a um ou mais valores de cor ou intensidade. Essa inicialização é conhecida como *seed points* (pontos sementes), e a partir desses pontos crescem as regiões pelo agrupamento de regiões próximas e similares (SOBIERANSKI, 2010). Os algoritmos de crescimento de regiões dependem muito do contexto da aplicação e também da métrica de similaridade usada no agrupamento de regiões. O critério de parada para o método de crescimento por regiões pode variar de acordo com a área da região ou até que não tenha mais *pixels* vizinhos que possam ser incluídos na sub-região (SOBIERANSKI, 2010).

Ao final da etapa de Detecção de Caminho, o trabalho é avaliado através de experimentos em campo, com dados reais, utilizando para isso um veículo que deve cumprir um trajeto enquanto roda o processamento digital da cena capturada em tempo real. Nesta avaliação o trajeto deve conter mudanças no padrão de terreno, buscando garantir o funcionamento da detecção adaptativa do caminho. Como Método para a avaliação é realizado uma comparação com o *Ground Truth* gravado manualmente (filmagens do caminho com o veículo em condução manual) e comparar com o mesmo caminho de forma autônoma.

1.5 ESTRUTURA DA DISSERTAÇÃO

Este trabalho está estruturado em seis capítulos, buscando com isso atender a todas etapas necessárias para o cumprimento da Dissertação, desde a introdução até as conclusões, passando por: fundamentação teórica, desenvolvimento, avaliação e resultados.

No Capítulo 1 tem-se a Introdução, onde foi apresentada uma visão geral do trabalho, falando sobre a detecção de caminho e segmentação de

imagem. Também na introdução, são apresentados a problematização, solução proposta, objetivo geral, objetivos específicos e a metodologia.

Já no Capítulo 2, a Fundamentação Teórica, é apresentada a revisão bibliográfica considerando os seguintes temas e áreas: Veículos autônomos e Detecção de Caminho, Segmentação de Imagem e por fim, uma série de trabalhos correlatos ao projeto.

No Capítulo 3 é apresentado o desenvolvimento do trabalho, os métodos utilizados, o que são e como funcionam. Desde os métodos de segmentação de imagem para a detecção do caminho, passando pela análise das imagens para a realização do controle e finalmente, a comunicação com o veículo através do Arduino.

A estratégia de avaliação e experimentos são apresentados de forma detalhada no Capítulo 4. Ainda neste capítulo, é apresentado o veículo utilizado para os testes, o ambiente onde serão realizados os testes, assim como o planejamento de testes feitos previamente.

Por fim, no Capítulo 5, os resultados obtidos são apresentados. E no Capítulo 6, apresentam-se as conclusões obtidas com esse trabalho. Falando sobre os resultados alcançados, problemas e soluções durante o desenvolvimento da pesquisa e mostrando as oportunidades para trabalhos futuros, possíveis de serem realizados a partir desta dissertação.

2 FUNDAMENTAÇÃO TEÓRICA

O veículo autônomo, ou seja um veículo que dirige sozinho, mostra-se útil em diversas situações, por exemplo, onde é necessária a análise de um ambiente porém há dificuldade de acesso de seres humanos (área terrestre muito acidentada, destroços de um acidente ou um ambiente hostil para a respiração) pode-se utilizar um veículo autônomo, evitando assim, colocar em risco a vida de um ser humano.

Um veículo autônomo é um sistema mecânico e elétrico complexo, exigindo o conhecimento de diversas áreas como engenharia elétrica, mecânica e computação. Os dispositivos e componentes eletrônicos, sensores, câmeras e controladores devem ser capazes de atuar em conjunto, e o software embarcado nesse sistema deve ser suficientemente complexo e inteligente para que a aquisição e processamento dos dados capturados e controle do veículo sejam realizados sem problemas. Isso permite definir um veículo autônomo como um robô móvel que pode navegar e interagir com um ambiente sem que haja intervenção humana (SCHWORER, 2005).

Existem diversas aplicações para veículos autônomos, em diferentes segmentos da sociedade, como: as indústrias, civis e militares. Uma das principais características é justamente que a navegação seja feita de forma autônoma. Isso vem sendo estudado desde a década de 80. Um veículo autônomo inteligente deve ser capaz de percorrer ambientes compostos por diferentes conjuntos de ruas e/ou tipos de estradas (asfalto, terra, etc.) (SALES et al., 2014).

Há diversas pesquisas e projetos relacionados aos veículos autônomos, e como justificativa por essa busca tem-se a segurança humana. Por exemplo no caso de veículos militares, com a utilização de um veículo autônomo para acesso a áreas hostis exclui-se o perigo de um soldado que precisaria estar dirigindo o veículo; na indústria, máquinas que realizam tarefas consideradas de risco para o controlador ou demais trabalhadores ao redor não estariam sujeitas as falhas humanas de concentração e atenção; assim como veículos civis, onde grande parte dos acidentes são causadas por falhas humanas de atenção, imprudência e desrespeito às leis de trânsito.

Os casos de maior sucesso atualmente, utilizam-se de uma grande variedade de sensores, não somente câmeras, mas informação de sensores de posicionamento global (GPS - *global positioning system*), radares e principalmente lasers. Como por exemplo o vencedor da terceira edição da competição de veículos autônomos organizada pela DARPA (*Defense Advanced Research Projects Agency*), o veículo nomeado de Boss (URMSON et al., 2008). A Google contrata muitos dos profissionais envolvidos na competição

criada pela DARPA, e desenvolve atualmente o seu próprio veículo autônomo que já vem sendo testado em ambientes urbanos e movimentados (URMSON, 2014b), (URMSON, 2014a), sendo o diretor desse projeto, o mesmo responsável pelo prêmio na terceira edição da DARPA com o veículo Boss. No Brasil também há um exemplo de projeto que já tem bastante destaque, o veículo conhecido como CaRINA (FERNANDES et al., 2014). Todos esses projetos utilizam diferentes sensores incluindo o LIDAR.

Porém, a navegação ainda é um enorme desafio que os pesquisadores e engenheiros devem superar ao projetar um veículo autônomo, sendo o principal fator para o sucesso e sua utilidade, afinal é a navegação que descreve como o veículo projetado se move e interage com o ambiente. Há dois focos principais no desenvolvimento de um carro autônomo: detecção e planejamento do caminho e desvio de obstáculos (SCHWORER, 2005).

2.1 DETECÇÃO DE CAMINHO

Pode-se definir caminho como: rota, trajeto, rua, estrada, ou seja, uma superfície onde os veículos podem transitar. Uma estrada pode ser definida como uma estrutura atravessável, linear e com uma cor que é distinta dos arredores que a envolvem, também deve-se estender por uma certa distância em frente ao veículo e ser suficientemente larga. Ou seja, pode-se utilizar as seguintes propriedades para a detecção de caminho: superfície, borda e direção. Entretanto, a dificuldade na detecção de caminho reside justamente nas variações que essas propriedades podem ter, como por exemplo iluminação (sol, sombra) ou alterações na superfície do caminho (OSOSINSKI; LABROSSE, 2013).

A detecção de caminho é uma das questões-chave no desenvolvimento de um veículo inteligente, por isso muita pesquisa vem sendo feita neste enfoque, inclusive verificando a viabilidade da utilização de métodos baseados em visão computacional (GUO; YAMABE; MITA, 2013). Na extração de características do caminho com base nas imagens, pode-se utilizar técnicas de visão computacional e processamento digital de imagens, mais especificamente, métodos de segmentação, permitindo assim separar toda a região considerada como caminho do restante da imagem.

Segundo um levantamento feito por (OSOSINSKI; LABROSSE, 2013), devido a grande variabilidade que as superfícies, bordas e direções que um caminho pode ter, alguns trabalhos propõem a utilização de redes neurais treinadas para reconhecer características específicas do caminho diretamente das imagens, como por exemplo os trabalhos de: (JOCHEM; POMERLEAU; THORPE, 1995) e (POMERLEAU, 1993). Entretanto, conforme (OSOSINSKI;

LABROSSE, 2013) esses trabalhos que tentaram utilizar redes neurais apresentaram dificuldades em construir um conjunto representativo de formação de imagens ou treinar as redes neurais para aprenderem as variabilidades das propriedades na detecção de caminho. Buscando superar essas dificuldades, alguns métodos dependem de processos externos e que utilizam diferentes parâmetros de rede neural com base no tipo de caminho a ser conduzido, por exemplo: (ROSENBLUM, 2000). Outro problema é com as imagens em resolução completa ou mesmo a resolução completa da representação de cores (BAO; CHEN; YU, 2012).

Ainda segundo (OSOSINSKI; LABROSSE, 2013), alguns autores buscando maior segurança e confiabilidade na detecção de caminho optaram por métodos que realizam a extração de características do caminho, marcações geralmente pintadas nas estradas estruturadas e recentemente também com relação as bordas de estradas não estruturadas. Esses métodos, no entanto, necessitam de caminhos que tenham marcações ou bordas bem definidas, e isso, em muitos casos não é possível.

O trabalho de (ZHOU et al., 2010) categoriza os algoritmos de detecção de estrada em três tipos, cujas diferenças entre eles são no que se baseiam para obter o resultado: técnicas baseadas em características; técnicas baseadas em modelo e técnicas baseadas em regiões. Onde, as baseadas em características são mais eficazes, mas requerem que as estradas tenham marcações bem definidas e de fácil identificação, e ruídos podem interromper todo o processo de detecção. Os baseados em modelo são mais robustos, entretanto, tem uma grande restrição justamente pela geometria dos modelos. Os mais eficazes das técnicas baseadas em regiões podem ser vistos como problemas de aprendizado de máquina. Utilizando o aprendizado de máquina, pode-se lidar com problemas de ruídos e com as constantes alterações no ambiente. Nesse caso a maior dificuldade é realizar o treinamento em tempo real da máquina para que ela consiga realizar o aprendizado do novo ambiente.

As técnicas baseadas em regiões tratam, mais especificamente, da extração de informações de cor e textura dos *pixels* na imagem capturada, onde deve-se com base na cor dos pixels capturados, segmentar a imagem capturada em regiões consideradas ou não do caminho.

2.2 SEGMENTAÇÃO DE IMAGENS

Normalmente o primeiro passo na análise de uma imagem é a sua segmentação. Na área de processamento digital de imagem (PDI), segmentar significa dividir a imagem de entrada I em partes, sabendo-se que a quantidade de divisões efetuadas depende do contexto do problema. Geralmente a

segmentação é uma das tarefas mais difíceis em PDI, mas recompensa por tornar mais simples a análise computacional da imagem (NEVES; PELAES, 2001).

A segmentação de imagem é basicamente separar a imagem por regiões (e suas fronteiras) onde tenham características de sinais em comum que os *pixels* compartilham, tais como: níveis de intensidade, textura ou cor (SOBIERANSKI, 2010). Com o processo de segmentação, a análise da imagem torna-se mais simples, uma vez que existe uma redução no campo de busca devido aos agrupamentos dos *pixels* em segmentos.

Como visto anteriormente, na introdução, a segmentação de imagem é comumente fundamentada em homogeneidade do sinal (características em comum que os pixels compartilham, tais como: níveis de intensidade, textura ou cor) e continuidade (combinação de padrões de homogeneidade ao longo do domínio espacial da imagem). Esses critérios permitem a localização de regiões homogêneas e das fronteiras que são compartilhadas entre as regiões adjacentes (SOBIERANSKI, 2010).

A segmentação pode seguir duas propriedades: descontinuidade e similaridade. No caso da descontinuidade a partição da imagem é feita com base nas alterações bruscas de intensidade, como por exemplo, a detecção de contornos (bordas), detecção de pontos isolados e detecção de linhas. Para a similaridade a partição da imagem é feita com base na similaridade entre os *pixels*, seguindo um critério definido como, binarização, crescimento de regiões (*Region Growing*), divisão e junção de regiões (*Split & Merge*), aglomeração (*Clustering*) e limiares (*Thresholding*) (NEVES; PELAES, 2001).

2.2.1 Detecção de Descontinuidades

Detecção de descontinuidades divide-se em três tipos: detecção de pontos isolados, detecção de linhas e detecção de bordas. Conforme o exemplo da Figura 1, onde tem-se um exemplo de máscara utilizada para a detecção das linhas, colunas e/ou diagonais.

Figura 1 – Detecção de Linhas (adaptado de (NEVES; PELAES, 2001)).

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		

Na detecção de pontos isolados é feita uma detecção de forma direta e basicamente mede a diferença entre o ponto central e seus vizinhos. A detecção de linhas, assim como a detecção de pontos isolados, também pode segmentar para detectar linhas, colunas e diagonais. A detecção de bordas é provavelmente a abordagem mais utilizada para a detecção de descontinuidades (NEVES; PELAES, 2001).

2.2.2 Detecção de Similaridade

Segundo Sobieranski (2010), similaridade é a notação usada para indicar a existência de semelhança entre dois ou mais objetos. Para a área de PDI e Visão Computacional (VC), uma função de similaridade influencia diretamente na qualidade dos resultados. Dois *pixels* somente podem ser considerados similares, quando uma função identificar que existe semelhança entre esses *pixels*. Uma função de similaridade é analogamente uma métrica de distância.

Algumas das técnicas baseadas em similaridade são:

- **Limiarização (*Thresholding*):** A operação de limiarização pode ser descrita como uma técnica de processamento de imagens na qual uma de entrada $f(x, y)$ de N níveis de cinza produz uma imagem de saída $g(x, y)$:

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) \geq T \\ 0 & \text{se } f(x, y) < T \end{cases} \quad (2.1)$$

onde, os *pixels* rotulados com 1 correspondem aos objetos e os *pixels* rotulados com 0 correspondem ao fundo, e T é um valor predefinido denominado de limiar (NEVES; PELAES, 2001).

- **Aglomeración (*Clustering*):** Refere-se ao agrupamento de um dado conjunto de objetos em subconjuntos de acordo com as propriedades de cada objeto. Atua sobre um espaço chamado de espaço de atributos, que representa um espaço de dimensão elevada, onde cada ponto da imagem é representado por um vetor de atributos, esses atributos são medidas feitas sobre a imagem, como tamanho dos objetos na imagem. A segmentação por aglomeração é dividida em duas etapas, sendo a primeira uma etapa onde são computados os vetores de atributos e na segunda é determinada um número ótimo de aglomerados (NEVES; PELAES, 2001).

- **Crescimento de Regiões (*Region Growing*):** Agrupa *pixels* ou sub-regiões em regiões maiores. O crescimento de regiões inicia em relação às coordenadas de entrada sobre uma imagem $I(x,y)$ ou em relação a um ou mais valores de cor ou intensidade. Essa inicialização é conhecida como *seed points* (pontos sementes), e a partir desses pontos crescem as regiões pelo agrupamento de regiões próximas e similares (SOBIERANSKI, 2010). Os algoritmos de crescimento de regiões dependem muito do contexto da aplicação e também da métrica de similaridade usada no agrupamento de regiões. O critério de parada para o método de crescimento por regiões pode variar de acordo com a área da região ou até que não tenha mais *pixels* vizinhos que possam ser incluídos na sub-região (SOBIERANSKI, 2010).
- ***Splitting and Merging*:** Subdivide uma imagem de entrada I em um conjunto de regiões disjuntas, podendo então, mesclar ou separar essas regiões. Esse método é baseado em duas etapas. *Splitting*, que subdivide uma imagem de entrada em quadrantes menores. E a etapa *Merging*, que ocorre depois da realização da etapa *Splitting*. Esse processo é repetido enquanto nenhum *merging* ou *splitting* for possível (SOBIERANSKI, 2010).

2.3 REVISÃO BIBLIOGRÁFICA

Uma série de trabalhos foram levantados relacionados à detecção de caminho. Importante ressaltar que, a seleção destes trabalhos levou em consideração somente os trabalhos que a câmera estivesse posicionada à frente do veículo, ou seja uma visão frontal e horizontal para o cenário de captura, excluindo assim diversos trabalhos, onde o foco era por imagens aéreas com vista para o solo.

Foram utilizados como critérios de busca: *journals* com data de 2006 até atualmente com foco em trabalhos de Visão Computacional. Foram definidos dois locais de pesquisa, que são: IEEE e Science Direct.

Para a detecção de caminho foram utilizados os seguintes termos de pesquisa: *road detection*, *road tracking* e *road segmentation*. Chegando a um total de 301 artigos na busca, onde após a exclusão restaram 20 artigos, na IEEE e 114 artigos na busca com 9 mantidos na Science Direct. Como critérios para exclusão, foram eliminadas imagens aéreas (mantendo somente imagens frontais), foi levado em consideração a relevância do trabalho, os métodos utilizados (visão computacional) e se realmente se encaixavam no tema (veículos autônomos).

2.3.1 Revisão Sistemática - Detecção de Caminho

Para os trabalhos na etapa de detecção de caminho, foram levantados: os métodos utilizados, cenário de uso e se era um trabalho adaptativo em relação à troca de tipos de terreno.

Diferentes métodos foram encontrados para a detecção de caminho, dentre estes a Detecção de Bordas é o método mais utilizado, totalizando 11 dos 29 artigos utilizados na revisão bibliográfica. Seguido pelas técnicas relacionadas à Visão Estereoscópica (Correspondência Estéreo). Técnicas de Segmentação de Imagens também foram utilizadas, sendo este um processo que se encaixa perfeitamente no contexto de detecção de caminho. Uma série de outras técnicas e métodos foram encontrados, entretanto em menor quantidade de utilização. Houve inclusive um trabalho que utilizava modelos de estrada previamente definidos.

O Critério de Cenário foi definido para verificar se o trabalho era voltado para detecções de caminho em diferentes tipos de terreno (Ex.: Asfalto, terra e variações), trabalhos que realizam a detecção de caminho independente da alteração do terreno são classificados como adaptativos (não necessariamente em tempo-real).

Crítérios de Busca

- Artigos;
- 2006 até atualmente;
- Foco em trabalhos de Visão Computacional;

Locais de Pesquisa

- IEEE;
- *Science Direct*;

Termos de Pesquisa

- *road detection*;
- *road tracking*;
- *road segmentation*;

IEEE Xplore

- **Busca:** *road detection + road tracking + road segmentation*; (Em: título, resumo ou palavras-chave);

- **Total de Artigos Encontrados:** 301 artigos;
- **Total de Artigos Mantidos:** 20 artigos;

Science Direct

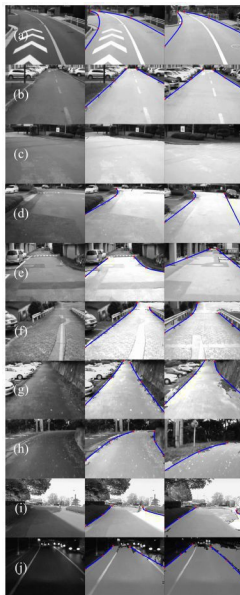
- **Busca:** *road detection OR road tracking OR road segmentation* (Em: título, resumo ou palavras-chave);
- **Total de Artigos Encontrados:** 114 artigos;
- **Total de Artigos Mantidos:** 9 artigos;

CrITÉrio para Exclusão: Trabalhos com imagens frontais (eliminar trabalhos com imagens aéreas), Relevância do trabalho, métodos utilizados e se realmente encaixa no tema (veículos autônomos);

2.3.1.1 Trabalhos Correlatos - Detecção de Caminho

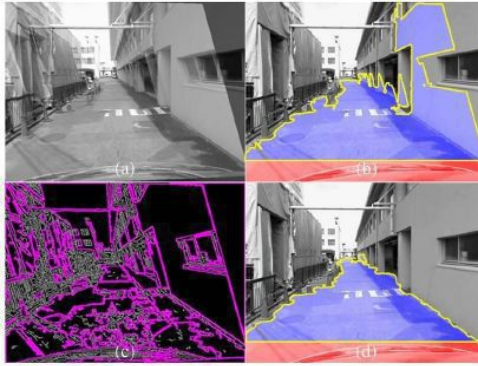
[1] Em (GUO; MITA, 2009b) é utilizado Visão Estereoscópica para a detecção das bordas laterais de uma estrada. Para isso, primeiramente é feita uma rotulação para verificar se cada *pixel* na imagem é de estrada ou não, neste caso, *pixel* de estrada recebe valor 1 e *pixel* de não-estrada recebe valor 0. Isso ocorre através de uma função de minimização de energia que representa a região plana de estrada. Esta função de energia é definida pela utilização das transformações de projeção 2D vindas como informações da estereoscopia e de um algoritmo de inferência em pedaços binários de *Markov Random Field* (MRF). Por fim, os limites da estrada são gerados separadamente usando o algoritmo *Catmull-Rom splines based on RANdom SAmple Consensus* (RANSAC) com diferentes modelos de estrutura da estrada para ajudar um veículo autônomo a entender a estrutura bem como a faixa de segurança da estrada atual. Os resultados mostrados são em estradas de asfalto mas com superfícies não homogêneas.

Figura 2 – Resultados de (GUO; MITA, 2009b).



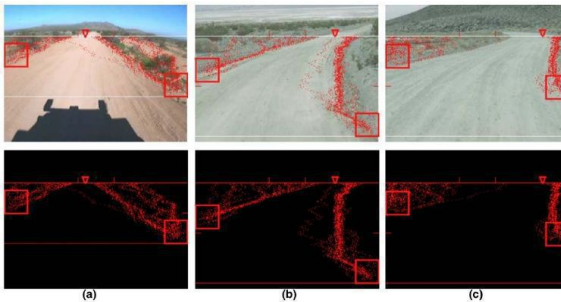
[2] O trabalho de (GUO; MITA; MCALLESTER, 2012) é basicamente um avanço no trabalho de (GUO; MITA, 2009b) e também utiliza Visão Estereoscópica, com isso tem informações de geometria (razoavelmente plana) da estrada como critérios para a detecção do caminho. Assim como em (GUO; MITA, 2009b) realiza a rotulação dos pixels (estrada ou não) pelo classificador de estrada *Markov Random Field-Based*. Porém neste trabalho o cenário é flexível para qualquer tipo de estrada, tornando assim num trabalho adaptativo, através de um aprendizado não supervisionado. Outra melhora que também foi realizada neste trabalho, é a eliminação de regiões sem textura e que dificultam a detecção da estrada com áreas que não deveriam estar na detecção, por exemplo: paredes.

Figura 3 – Resultados de (GUO; MITA; MCALLESTER, 2012).



[3] Em (BROGGI; CATTANI, 2006), apresentam um trabalho feito para a grande competição de carros autônomos, DARPA em 2004. O sistema de visão artificial desenvolvido é parte do veículo TerraMax, veículo este que foi utilizado na competição. Esse trabalho basicamente detecta bordas laterais da estrada, para isso utiliza o algoritmo da Otimização da Colônia de Formigas (*Ant Colony Optimization*) para a detecção das bordas da estrada. Os resultados deste trabalho foram apresentados utilizando imagens de estrada de terra não homogênea. Porém é dito no artigo que a competição Darpa inclui em seu trajeto estradas não pavimentadas, trilhas arenosas e rochosas, arbustos, lagos secos e, em pequena porcentagem, estradas pavimentadas.

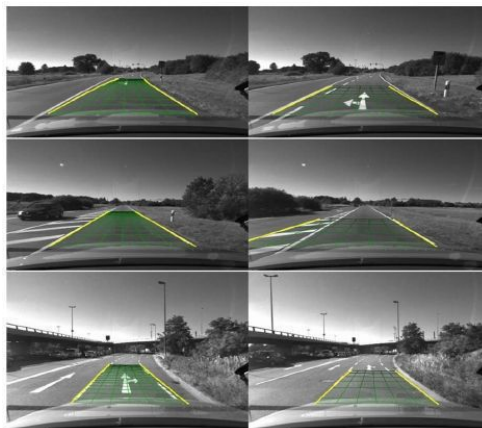
Figura 4 – Resultados de (BROGGI; CATTANI, 2006).



[4] O trabalho de (DANESCU; NEDEVSCHI, 2009) utiliza uma técnica para a detecção de tons com base na fusão de duas imagens da estrada obtidas através de Visão Estereoscópica. Utiliza também um método para

a medição de partículas e de pesagem utilizando múltiplas delimitações da pista, extraídos pelos tons de cinza e processamento estéreo. Por fim, um método para decidir sobre a validade dos resultados de estimação de estrada. Amostras de inicialização são utilizadas para tratamento uniforme das descontinuidades da estrada, eliminando assim, a necessidade de inicialização explícita do caminho. Os resultados deste trabalho são mostrados em estradas de asfalto.

Figura 5 – Resultados de (DANESCU; NEDEVSKI, 2009) Esquerda resultado do artigo, direita comparação com *Kalman Filter*.



[5] Em (DORNAIKA et al., 2011) mais um trabalho que utiliza Visão Estereoscópica é apresentado. Neste artigo é proposto um *framework* para estimação em tempo real, o qual funciona em estradas urbanas, rodovias e em estradas de terra (nas imagens do artigo são apresentados exemplos em estradas de asfalto). A abordagem é dividida em duas etapas que ocorrem a cada par de frames capturadas pelo sistema estereoscópico. A primeira etapa é a segmentação da região de estrada em uma captura monocular. A segunda etapa estima a pose da câmera usando um registro inexpressivo entre a região da estrada segmentada pela visão monocular e a outra visão do par estéreo.

Figura 6 – Resultados de (DORNAIKA et al., 2011).



[6] No trabalho de (CHEN; TAI, 2010) é apresentado um novo espaço de cor, chamado *RGB color ratio space*, que nada mais é do que um espaço de cor que herda características dos componentes de cor do espaço de cor RGB e características de intensidade do espaço HSI. Com base nesse novo espaço de cor é feita a segmentação do caminho, e para essa segmentação é utilizada Lógica Fuzzy para o agrupamento dos pixels. A Lógica Fuzzy tem como entrada os componentes de cor RGB e a intensidade, as regras de saída são definidas como similares quando os componentes RGB ficam dentro de limites das funções de pertinência. Esse agrupamento é adaptativo, pois sempre altera-se o *pixel* semente utilizado para comparar com o restante da imagem, e os limites de pertinência fuzzy são baseados nesses *pixels* sementes. Por fim utiliza-se uma rede neural *backpropagation* para o ganho de intensidade dos *pixels*. Pelo fato de conter estas características este trabalho é bastante flexível quanto aos tipos de terrenos segmentados como estrada.

Figura 7 – Resultados de (CHEN; TAI, 2010). Na primeira coluna as imagens originais. Segunda coluna imagem segmentada. Na terceira coluna, ajuste de iluminação. Na quarta coluna, segmentação com base na imagem original e na imagem com ajuste de iluminação.



[7] O trabalho de (ÁLVAREZ; GEVERS; LÓPEZ, 2009) utiliza uma base de casos, com algumas geometrias (formas de estrada), como por exemplo: estrada reta sem obstáculo, estrada com obstáculo na direita, estrada com obstáculo na esquerda, estrada com curva leve para direita ou esquerda e estradas com curvas acentuadas para esquerda ou direita. Além dessa base, também é realizada a detecção de estrada, tendo como base os *pixels* da imagem. Por fim, a abordagem proposta combina a probabilidade do caminho (em relação a base de casos) com o algoritmo de detecção de estrada com base em *pixel*. O cenário, entretanto, é limitado à rodovias não urbanas evitando assim a necessidade de mais formas na base caso fosse em um ambiente urbano.

Figura 8 – Resultados de (ÁLVAREZ; GEVERS; LÓPEZ, 2009).



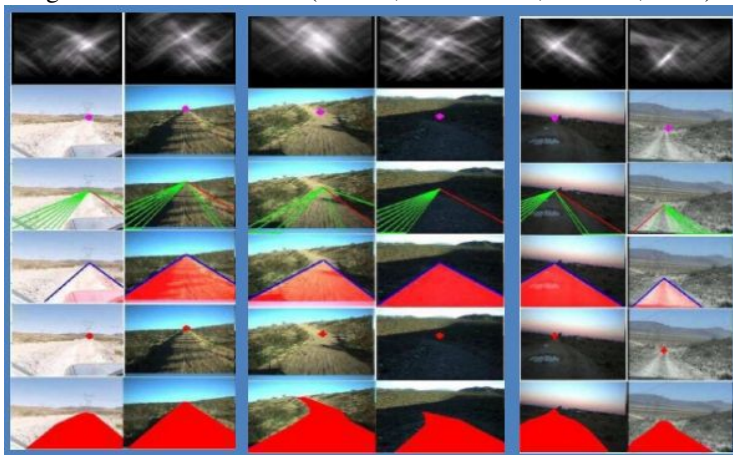
[8] Buscando tratar problemas de sombras e variação de iluminação (ÁLVAREZ; LÓPEZ, 2011) propõe uma abordagem que se baseia na utilização de um espaço de características invariantes à sombra combinado com um classificador baseado em modelo. Modelo este que é construído online, melhorando a adaptabilidade do algoritmo para a iluminação atual e a presença de outros veículos na cena. O algoritmo proposto neste trabalho funciona em imagens estáticas e diferente do (ÁLVAREZ; GEVERS; LÓPEZ, 2009) não depende de formas de estrada e não tem restrições temporais.

Figura 9 – Resultados de (ÁLVAREZ; LÓPEZ, 2011).



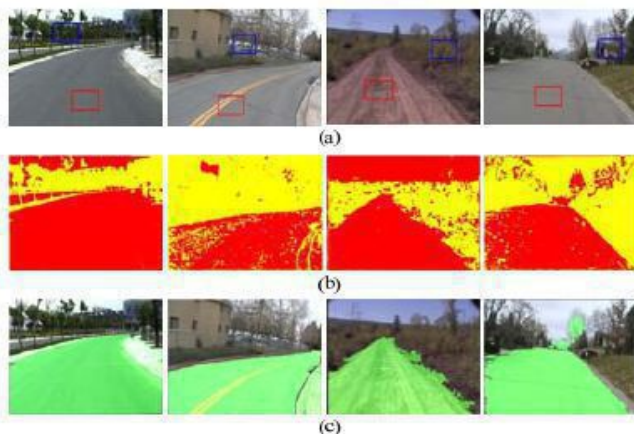
[9] Mais um trabalho com foco adaptativo é (KONG; AUDIBERT; PONCE, 2010), provavelmente é o trabalho encontrado que mostra exemplos mais flexíveis de detecção de caminho, como por exemplo: estradas com terra em diversas texturas, asfalto e estradas com neve. A primeira etapa neste trabalho é a detecção do ponto de desaparecimento da estrada, ou seja o ponto de horizonte. Detecta as bordas laterais da estrada através das características de cor combinadas com a razão de consistência da orientação. Por fim, segmenta a estrada com base nas bordas encontradas.

Figura 10 – Resultados de (KONG; AUDIBERT; PONCE, 2010).



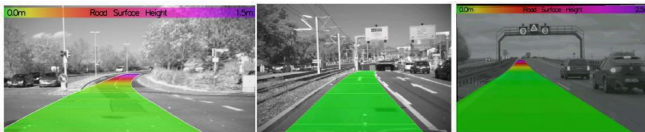
[10] (ZHOU et al., 2010) faz a detecção de estrada através do classificador *Support Vector Machines* (SVM). Este também é um trabalho adaptativo com treinamento e aprendizado em tempo real. E o cenário é flexível e funciona para asfalto, terra e neve.

Figura 11 – Resultados de (ZHOU et al., 2010). Em (a), as imagens originais e os pontos de amostra. Em (b), classificação dos resultados, vermelho estrada e amarelo não-estrada. Em (c), resultados de estrada rotulados em tom de verde.



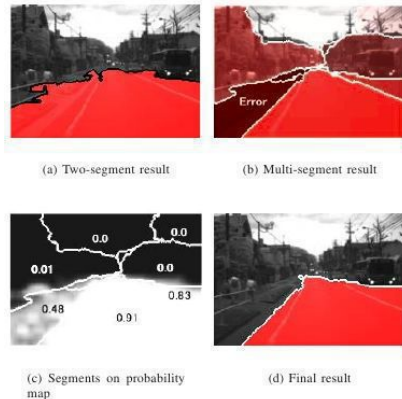
[11] O trabalho de (WEDEL et al., 2009) propõe uma técnica genérica para a modelagem da superfície visível em frente ao veículo. A abordagem do trabalho para o problema da dificuldade de trabalhar com estradas não planares é a de representar a superfície como uma curva B-Spline paramétrica. Os parâmetros para a superfície são estimados utilizando visão estéreo para calcular o espaço livre em frente ao veículo. Para alcançar o objetivo, os autores adotaram um algoritmo de segmentação de obstáculos em estradas recentemente proposto junto com sua abordagem de representação da superfície via B-Spline. Os autores ainda citam que o trabalho apresentado é parte de um projeto maior buscando melhorar sistemas de manobra automática de veículos.

Figura 12 – Resultados de (WEDEL et al., 2009).



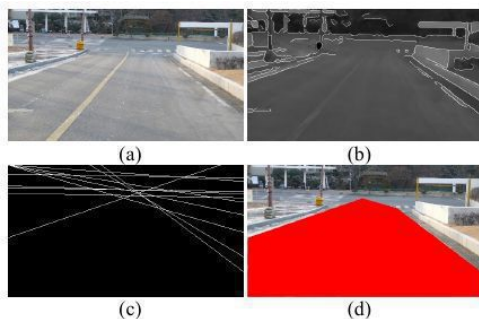
[12] No trabalho de (SON; MITA; TAKEUCHI, 2008) é apresentado um método de detecção de estrada baseada em algoritmo de segmentação de imagem. O método combina probabilidade a posteriori e informações visuais para a segmentação de imagem. Primeiramente, o identificador de estrada é treinado com um algoritmo de ensino supervisionado. Em seguida, as regiões da estrada são detectadas combinando cálculos de probabilidade e informação visual usando algoritmos de segmentação de imagem.

Figura 13 – Resultados de (SON; MITA; TAKEUCHI, 2008).



[13] Em (LEE; LEE; LEE, 2009) é proposto um sistema de geração e atualização de estradas desconhecidas utilizando GPS e reconhecimento de estradas. Ao adentrar uma estrada que não se encontra no banco de dados, ou se a estrada foi alterada, o dispositivo envia imagens da estrada para ser analisada pelo sistema, onde então o mapa é gerado. O sistema define os limites da estrada a partir de um algoritmo de detecção de limite da estrada. As fases para detectar a estrada segue da seguinte forma: é gerada uma imagem da estrada, onde a seguir os limites são detectados utilizando um filtro; após são detectados os segmentos de linha reta para então identificar a área da estrada.

Figura 14 – Resultados de (LEE; LEE; LEE, 2009).



[14] (KIM; SON; SOHN, 2011) propõe um método de detecção de estrada com iluminação invariante. É adotado um método de aprendizado para a direção da iluminação invariante sobre a superfície da estrada. Assim que a direção é especificada, os pixels são classificados como de estrada ou não. Ainda, é incorporado um método de layout de cena para reduzir os falso-positivos.

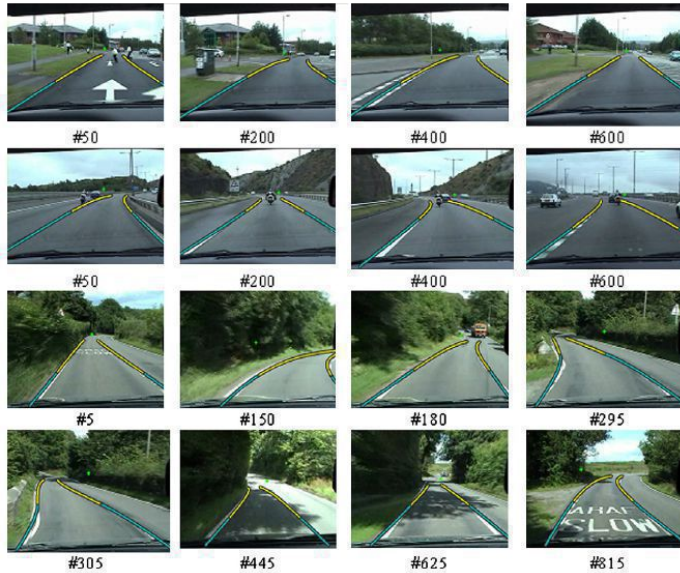
Figura 15 – Resultados de (KIM; SON; SOHN, 2011).



[15] (BAI; WANG, 2011) apresenta uma extensão do algoritmo de amostragem utilizado em filtros de partículas, o qual é integrado em um *framework* de múltiplos *trackers* para rastreamento de estradas. O trabalho utiliza a modelagem da estrada a partir de múltiplas hipérbolas (múltiplas, pois apenas

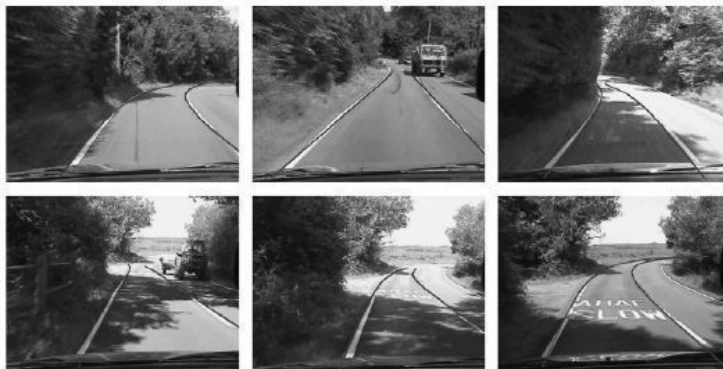
utilizando duas hipérbolas não se consegue modelar mudanças na estrada). Para o correto rastreamento da estrada o artigo explica que define estados para a hipérbole, e que a partir destes estados os autores utilizam o *framework* de amostragem de partículas.

Figura 16 – Resultados de (BAI; WANG, 2011).



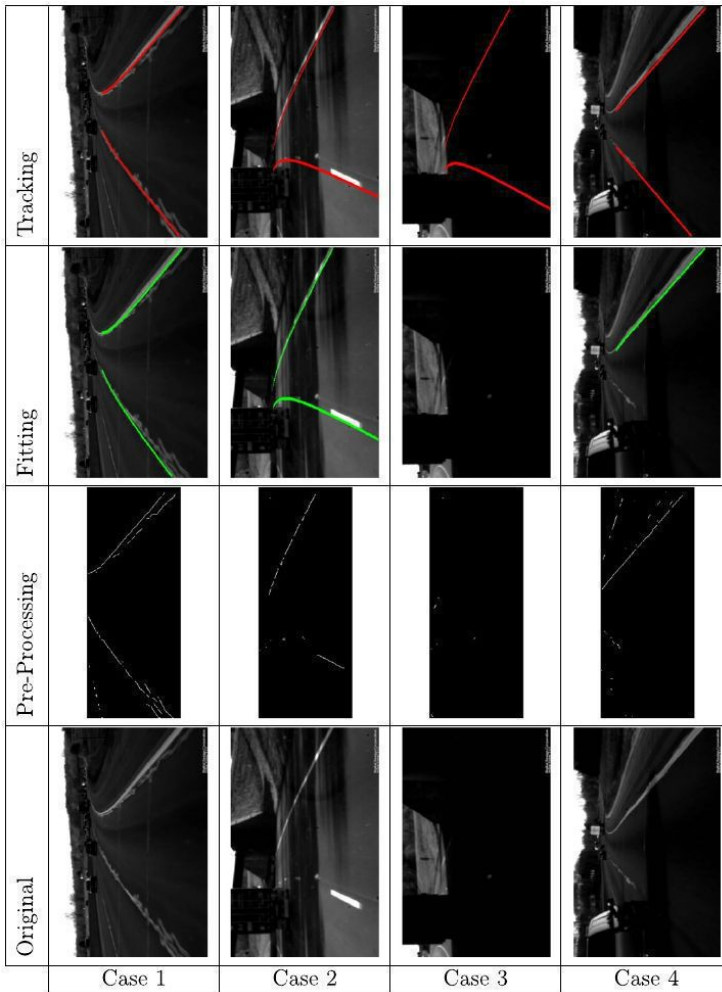
[16] A proposta de (BAI; WANG; FAIRHURST, 2008) apresenta um método robusto para detecção e rastreamento de estrada usando vários pontos de horizonte (pontos de fuga) e do filtro de condensação. A estrada é representada usando um modelo de hipérbole prolongado com adicional de um termo não-linear para lidar com transições entre segmentos de estradas retas e curvas. Os parâmetros do modelo de estrada são estimados usando múltiplos pontos de fuga localizados nos segmentos de estrada. A linha de fuga é determinada usando uma técnica apropriada de curva iterativa robusta para recuperar os parâmetros do modelo de estrada. Estes são então alimentados a um rastreador de condensação robusto (ISARD; BLAKE, 1998) para seguir a estrada. O rastreador é capaz de lidar com difíceis condições de estradas. Experimentos utilizando vídeos reais em estradas, demonstram a adequação da abordagem proposta em aplicações em tempo-real. Também é feita uma comparação com a técnica de filtragem de Kalman (KALMAN, 1960), mostrando assim a solidez da abordagem proposta.

Figura 17 – Resultados de (BAI; WANG; FAIRHURST, 2008).



[17] O trabalho de (MARZOTTO et al., 2010) apresenta um algoritmo de extração e rastreamento de estradas e caminhos, e sua implementação é em um dispositivo *Field Programmable Gate Array* (FPGA). A aplicação é particularmente adequada para utilização como um componente central de um sistema *Lane Departure Warning* - Aviso de Desvio na Pista (LDW), exigindo assim, alta performance de processamento digital de imagem, bem como dispositivos semicondutores de baixo custo, apropriados para o alto volume de produção do mercado automotivo. A tecnologia FPGA satisfaz estas duas exigências e o algoritmo apresentado foi todo projetado para ser incorporado neste tipo de hardware. As principais contribuições deste trabalho estão em: seleção adequada, customização e integração das principais funções para extração e monitoramento de estradas, e a implementação em um hardware FPGA. O algoritmo apresentado neste trabalho possui três módulos principais que são: (1) etapa de pré-processamento, onde ocorre a remoção de ruído, aumentando a qualidade da imagem e extraindo os candidatos a faixas de marcação na estrada (Métodos utilizados na primeira etapa: Corte da região de interesse, redução de ruído com uma Gaussiana, alongamento de histograma, extração de gradientes, limiarização para detecção de bordas, marcação de linhas com reconhecimento de padrões, operações morfológicas), (2) um módulo de ajustamento do modelo, que estima os parâmetros do modelo de estrada parabólico, utilizando um algoritmo *RANdom Sample and Consensus* (RANSAC), (3) uma fase para controlar os parâmetros do modelo para uma operação mais estável e confiável. Não são necessárias calibrações na câmera, o que, segundo os autores, torna o método mais adequado para ser implementado mais facilmente em aplicações reais.

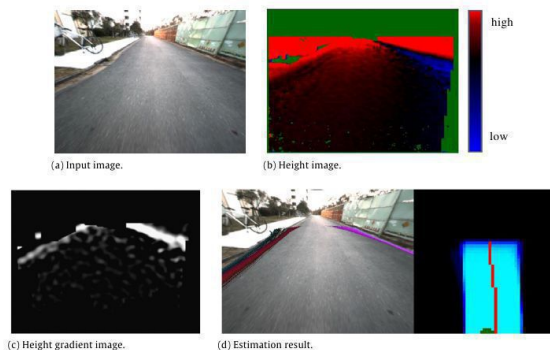
Figura 18 – Resultados de (MARZOTTO et al., 2010).



[18] (MATSUSHITA; MIURA, 2011) descreve um método de modelagem robusta de limites da estrada em tempo real para navegação autônoma. Sabendo que uma evidência sensorial para a detecção dos limites da estrada pode alterar de um lugar para outro, não se pode depender de uma única sugestão, usando assim várias características. Esse método também é necessário para lidar com as várias formas diferentes de estrada e com as alterações do tipo de estrada. As etapas contidas neste trabalho são: (1) A etapa de tran-

sição de estado gera um novo conjunto de partículas. Existem basicamente duas operações nesta etapa. Uma é a transição devido ao movimento do robô (veículo), que é prevista a partir da odometria (método para estimar posição de um robô) e os dados de imagem. A outra é da forma da estrada e alteração de tipo que ocorre quando uma nova parte da estrada se torna visível. (2) A etapa de predição de observação prevê a próxima observação da posição de robô e os parâmetros de estrada. (3) O passo de cálculo do peso determina primeiro as funções de probabilidade a partir do intervalo extraído e as características da imagem e em seguida, calcula o peso de importância de cada partícula. (4) O passo de reamostragem seletiva executa a reamostragem apenas quando necessário. Se o número eficaz de partículas é menor que a metade do número de partículas, a reamostragem é realizada. O método proposto foi aplicado com sucesso em várias cenas de estrada com câmeras e um telêmetro a laser. E para mostrar que o método proposto é aplicável a outros sensores, alguns resultados preliminares utilizando visão estéreo são descritos.

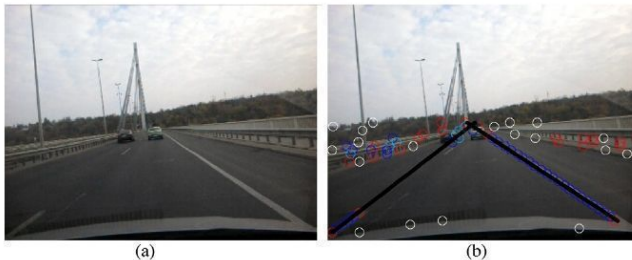
Figura 19 – Resultados de (MATSUSHITA; MIURA, 2011).



[19] (OBRADOVIC et al., 2013) propõe um novo modelo para detecção de pistas de estrada com base em um espaço de matemática fuzzy linear, juntamente com um robusto método de detecção de pistas de estrada usando agrupamento por meios fuzzy. A linha difusa baseada no modelo de pistas de estrada apresentada neste trabalho, descreve uma pista como um conjunto de pontos Fuzzy colineares. O algoritmo proposto para a detecção de linhas da estrada é capaz de lidar com dados imprecisos e permite reduzir a complexidade computacional (proporcional ao número de pontos difusos, multiplicado pelo número de linhas difusas) contra uma transformação de Hough padrão. Neste trabalho é utilizado um Smart Phone para a captura das imagens, assim como sensores integrados no dispositivo móvel (Android),

como: GPS, acelerômetro, orientação, rotação, sensor de luz. Então com essas entradas é realizada a extração de características da estrada (para isso é utilizado um detector de bordas, possibilitando encontrar as linhas de marcação na estrada, Ex.: Canny ou Sobel). Esses pontos entram no conjunto de pontos Fuzzy. Posteriormente é feita a detecção da pista da estrada (onde é descrito o novo algoritmo para detecção de linhas difusas a partir de imagens digitais, chamado FLDetector. Este algoritmo é implementado como um algoritmo Fuzzy c-means modificado, para agrupar conjuntos de pontos nebulosos. Cada ponto difuso pertence a uma linha difusa de acordo com uma relação colinear Fuzzy). E por fim a conclusão do módulo de controle Fuzzy. Os resultados experimentais mostram que o método proposto é bastante rápido, e robusto para utilização em aplicações em tempo real. O método proposto foi implementado como um aplicativo de telefone baseado no Android móvel.

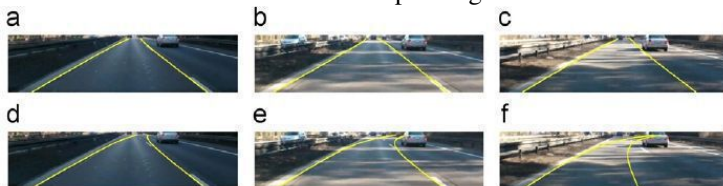
Figura 20 – Resultados de (OBRADOVIC et al., 2013).



[20] Em (WANG; DAHNOUN; ACHIM, 2012) é apresentado uma detecção e rastreamento de pista baseado em uma nova abordagem de extração de características de pistas e *Gaussian Sum Particle Filter* - filtro de soma de partículas Gaussianas (GSPF). A abordagem de extração de características proposta é baseada no fato de que com zoom nos pontos de horizonte das pistas, as marcações e limites da pista irão passar as mesmas linhas retas. Este algoritmo é capaz de refinar iterativamente vários mapas de características tradicionais e operar com as estradas curvas. Três tipos de algoritmos de rastreamento são comparados neste estudo: o filtro de partículas original *Sequential Importance Resampling* - Reamostragem de Importância Sequencial (SIR), o filtro de partículas Gaussianas - *Gaussian Particle Filter* (GPF), e a soma Gaussianas de partículas - *Gaussian Sum Particles Filter* (GSPF). O GSPF obtém o melhor desempenho, integrando uma nova função de probabilidade e um processo de seleção de parâmetros intuitiva. Tanto o GSPF como o GPF proporcionam melhor desempenho de rastreamento e exigem

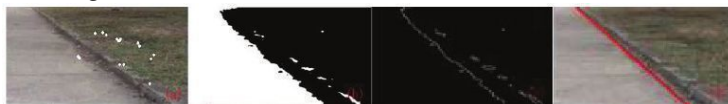
menos poder computacional do que o SIR. Também pode-se dizer que o desempenho de detecção e rastreamento é melhorado significativamente através da incorporação de um refinado mapa de características.

Figura 21 – Resultados de (WANG; DAHNOUN; ACHIM, 2012). Onde na primeira linha estão os resultados do mapa de características proposto e na segunda linha estão os resultados dos mapas de gradientes tradicionais.



[21] (ZHAO; WANG; YAN, 2012) propõe dois métodos complementares de detecção de bordas de estrada para estradas desestruturadas. Os métodos podem ser utilizados para navegação visual e para controle inteligente de robôs móveis. Um dos métodos é baseado em cor, aplicado em situações como uma mudança de estrada de asfalto e uma estrada de terra, onde a cor é obviamente diferente. O outro é baseado em uma segmentação por limiarização e pode ser aplicado sempre que o limite da estrada não é claro. Mesmo que a sombra seja um problema enorme na limiarização, segundo os autores, não tem grande impacto no método baseado em cor. Isso torna os dois métodos complementares. Além disso, com base na detecção de borda, a transformação de projeção perspectiva inversa é carregada para calcular o estado do robô. O experimentos em uma plataforma móvel, verifica que os dois métodos propostos complementam-se em diferentes ambientes de estrada.

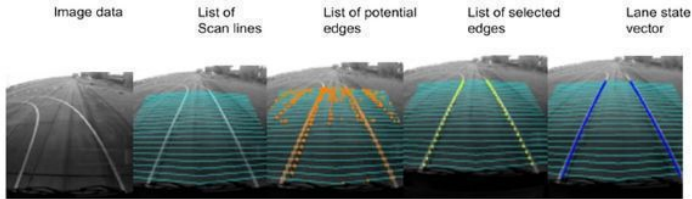
Figura 22 – Resultados de (ZHAO; WANG; YAN, 2012).



[22] Como os pesquisadores apresentam em (AMDITIS et al., 2010) aborda o desenvolvimento de um sistema que é capaz de lidar com um grande conjunto de situações de trânsito diferentes. É fundamentado em um ambiente de detecção onde um sistema de decisão é contruído. O módulo de decisão pode avisar o condutor ou corrigi-lo por meio de atuadores adequados que são acoplados para controle estratégico. A entrada para o sistema vem de câmeras, que são complementadas por sensores ativos (como radares e scan-

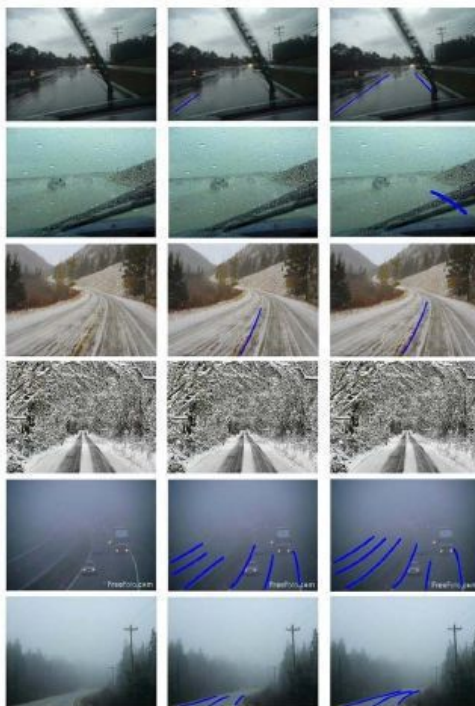
ners a laser) e os dados de veículos dinâmicos, mapas de estradas digitais e precisos dados de posicionamento do veículo. Neste trabalho, a concepção do sistema é apresentada dividido em três camadas: a camada de percepção, o qual é responsável pela percepção do ambiente, e as camadas de decisão e de ação, as quais são responsáveis por avaliar e executar ações, respectivamente.

Figura 23 – Resultados de (AMDITIS et al., 2010).



[23] No trabalho apresentado por (GOPALAN et al., 2012) existe uma abordagem baseada em aprendizagem, utilizando para isso, informações visuais, capturadas através de uma câmera acoplada na frente do veículo. O trabalho pode ser dividido em três partes, onde primeiramente há um descritor de características baseado em pixel, para modelar a informação compartilhada por marcações na pista. Depois aplica-se um algoritmo de impulsioneamento, robusto para selecionar as características contextuais relevantes para a detecção de marcação na pista. Por fim, utiliza-se o filtro de partículas para rastrear as marcações na pista, sem o conhecimento da velocidade do veículo, assumindo que as marcações na pista serão estáticas através da sequência de vídeo, que possibilita aprender as variações na cena da estrada a partir das estatísticas dos parâmetros do modelo rastreado.

Figura 24 – Resultados de (GOPALAN et al., 2012).



[24] Em (GUO; MITA, 2009a) é apresentado um método de detecção de região de estrada, baseado em estimativa de homografia, com aparência de estrada e modelos de estado de condução. No método, a região de estrada planar é detectada e os objetos dentro da região são localizados por uma transformação projectiva 2D entre o par de imagens estéreo, calculando a homografia induzida pelo plano de estrada dinamicamente. O método é composto por três módulos, que são: Módulo de classificação preliminar, que seleciona o classificador mais apropriado a partir do modelo de aparência de estrada para detectar, primeiramente, a região de estrada. O segundo módulo, é o módulo de detecção baseado em característica, o qual encontra as correspondências de pontos característicos no plano de estrada para estimar a homografia para o primeiro par de imagem, e em seguida, extrai a região de estrada dirigível. O terceiro módulo, que é o módulo de detecção com base na área, um processo de otimização não linear utiliza os resultados obtidos no módulo 2, tal como os valores iniciais para a estimativa de homografia, bem como a detecção

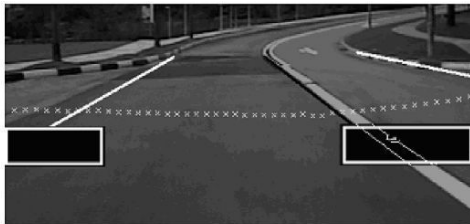
ção da região de estrada dirigível para os pares de imagem subsequentes com o modelo de estado condução, com base na informação sequencial. A combinação destes três módulos usa tanto evidência da imagem como informação temporal, enquanto isso, um mecanismo de correção de erro é aplicado.

Figura 25 – Resultados de (GUO; MITA, 2009a).



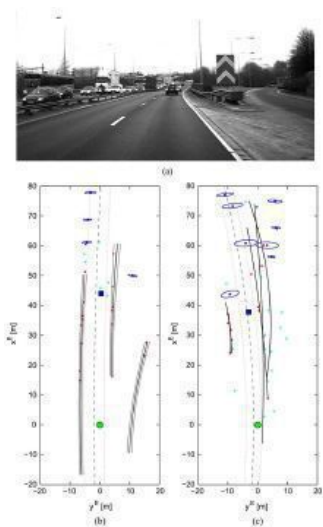
[25] O trabalho de (KODAGODA; WIJESOMA; BALASURIYA, 2006) atua em cenários de ambientes urbanos com asfalto muito bem definidos, depende fortemente das boas condições de marcação na estrada, e considera inclusive a existência de calçadas em ambos os lados da estrada, como informação de limite de estrada. Este trabalho apresenta uma nova metodologia de rastreamento de meio-fio. O rastreamento é feito por sensores no próprio veículo, ou seja uma plataforma móvel. Um segmento de meio-fio se presume como alvo e é modelado como um processo não-linear de comutação de Markov. A orientação do alvo e suas medições e localização são simultaneamente obtidas utilizando uma varredura de radar laser (LADAR) bidimensional, 2D, e uma câmera monocular, dispositivo de carga acoplada (CCD), e são então modeladas como observações do estado base. Um algoritmo de rastreamento de meio-fio, conhecido como *Curb Tracking and Estimation* (CuTE), é aplicado usando informações de sensor modal múltiplo, assim, sintetizado em uma estrutura de filtragem de modelo múltipla interativa e melhorada de imagem.

Figura 26 – Resultados de (KODAGODA; WIJESOMA; BALASURIYA, 2006).



[26] Em (LUNDQUIST; ORGUNER; GUSTAFSSON, 2011) é apresentado um *framework* de rastreamento de alvo que usa polinômios para modelar objetos na cena de interesse a partir de dados de um sensor de imagens. Modelos de espaço de estado são propostos para objetos estendidos, que permite o uso do filtro de Kalman no rastreamento. Lundquist, Orguner e Gustafsson (2011) apresentam também um algoritmo de rastreamento de alvos geral que utiliza um método de associação de dados específicos para os alvos apresentados. O algoritmo depende de uma informação prévia para que se possa detectar e inicializar as faixas a partir de pontos na cena.

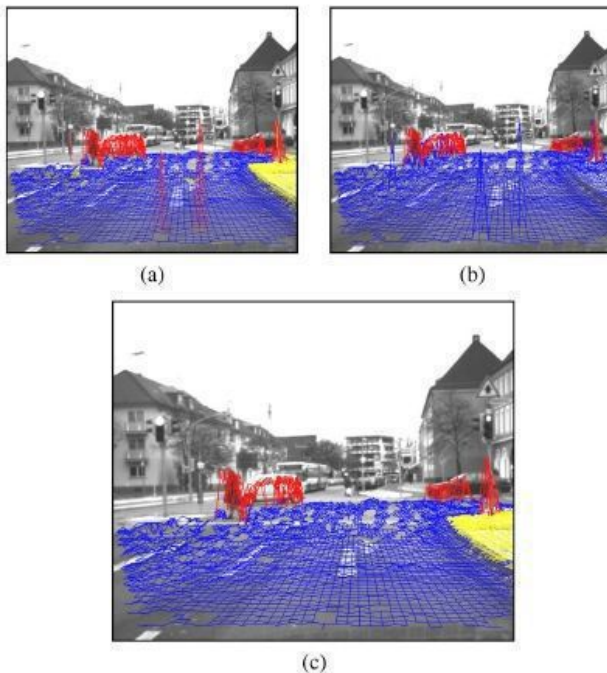
Figura 27 – Resultados de (LUNDQUIST; ORGUNER; GUSTAFSSON, 2011).



[27] Em (ONIGA; NEDEVSCHI, 2010), é apresentado uma nova abordagem para a detecção de estrada e de obstáculos em tempo real. Esse algoritmo foi avaliado em cenário urbano, e pode ser utilizado em complexas aplicações de desvio de obstáculos, para evitar colisões no planejamento de caminhos. Primeiramente é feita uma reconstrução 3D, através de densidade estéreo. Os dados 3D obtidos da densidade estéreo são transformados em Mapa de Elevação Digital - *Digital Elevation Map* (DEM) retangular. Dois classificadores são propostos, um baseado em densidade e outro baseado em superfície da estrada. O classificador de obstáculos baseado em densidade, marca no DEM as células como estrada ou como obstáculo, utilizando os pontos de densidade 3D como critério. Um modelo quadrático de superfície

de estrada é inicialmente ajustado por um método de consenso da amostra aleatória Random Sample Consensus (RANSAC) para a região em frente ao veículo. A superfície de estrada é utilizada como discriminação entre os pontos de estrada, ilha de tráfego, e obstáculo. Por fim uma fusão e filtragem de erro são realizadas sobre os resultados dos classificadores.

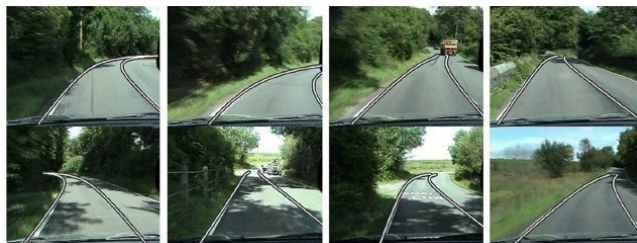
Figura 28 – Resultados de (ONIGA; NEDEVSKI, 2010). (a) Saída do classificador baseado na superfície de estrada. (b) Saída do classificador baseado em densidade. (c) Resultado após fusão dos dois classificadores e da filtragem de erro.



[28] Em (WANG; BAI; FAIRHURST, 2008), é apresentado um método para detecção e rastreamento, baseado em um filtro de partículas de condensação, para aplicações de vídeo em tempo real para navegação autônoma. A imagem é dividida em tiras horizontais, e a detecção do ponto de fuga (VP - *Vanishing Point*) é realizada em cada uma das tiras da imagem. O método proposto aqui, busca estimar a densidade de segmentos de linha de contorno (limite) da estrada na imagem, de modo que a detecção do ponto de fuga em uma tira horizontal da imagem toma em consideração os resultados de detec-

ção das tiras vizinhas. Segundo os autores, o uso desta informação contextual para a detecção dos pontos de fuga, proporciona resultados de detecção mais precisos. Os parâmetros de estrada estimados são utilizados para inicializar o rastreador de condensação.

Figura 29 – Resultados de (WANG; BAI; FAIRHURST, 2008).



[29] O Artigo apresentado por (WU; LIN; CHEN, 2009) mostra uma abordagem para calcular rapidamente a projeção de largura de vias, permitindo assim, prever as posições e largura das marcações de pista. Uma Máquina de estado finito de extração de marcações na pista é projetada para extrair pontos na imagem que tenham características de marcações da pista, e um B-spline cúbico é utilizado para a realização de ajuste de curva na reconstrução da geometria da estrada. Também é proposto neste trabalho um algoritmo de busca estatística para corretamente e adaptativamente determinar limiares sob vários tipos de condições de iluminação. Além disso, os parâmetros da câmera em um carro em movimento podem ser alterados com a vibração, então um algoritmo de calibração dinâmico é aplicado para calibrar os parâmetros da câmera e as larguras de faixa com a informação da pista de projeção. Além disso, a lógica fuzzy é aplicada para determinar a situação de oclusão. Finalmente, uma estratégia de determinação de região de interesse é desenvolvida para reduzir a área de busca e para tornar a detecção mais robusta com relação à oclusão nas marcações de pista ou de alterações complexas de curvas e limites de estrada.

Figura 30 – Resultados de (WU; LIN; CHEN, 2009).



2.3.1.2 Comparativo Entre Trabalhos Correlatos

A tabela 1 apresenta os trabalhos correlatos de forma mais simples e direta, destacando pontos de interesse, como: trabalhos que realizam a detecção do caminho de forma adaptativa (com a detecção se adaptando às mudanças no padrão do terreno), os cenários onde foram realizados os testes em cada trabalho (asfalto, terra, etc.) e quais métodos foram utilizados em cada trabalho. Com base na tabela é possível perceber, por exemplo, que os trabalhos que realizam a detecção adaptativa são minoria (9 de 29) perto de todos trabalhos selecionados sobre a detecção de caminho. Também com o auxílio da tabela facilita perceber, como dito no início deste capítulo, que o método mais utilizado é a detecção de bordas.

Tabela 1 – Tabela Comparativa - Detecção de Caminho

ID	Ano de Publicação	Adaptativo	Cenário	Métodos
[1]	2009		Asfalto (áreas não homogêneas)	Visão Estéreo; Detecção de Bordas
[2]	2012	x	Qualquer tipo de terreno	Visão Estéreo; Rotulação de <i>Pixels</i> ; Elimina Regiões sem Textura
[3]	2006		Estrada de terra	Detecção de Bordas; Otimização da Colônia de Formigas
[4]	2009		Asfalto	Visão Estéreo; Filtragem de Partículas
[5]	2011	x	Asfalto e terra	Visão Estéreo; Segmentação de Imagem (Na imagem de captura direita)
[6]	2010	x	Qualquer tipo de terreno	Segmentação de Imagem; Espaço de cor com características de RGB e HSI; Lógica Fuzzy (para agrupamento de pixels); Rede Neural Backpropagation (para ganho de intensidades)
[7]	2009		Asfalto (cenário não urbano)	Base de casos com modelos de estrada; Detecção de estrada <i>pixel-based</i> ; Combina a probabilidade de caminho (em relação a base de geometria de estradas) e o algoritmo de detecção de <i>pixels</i>

[8]	2011		Asfalto	Características de espaço de iluminação invariante (depende da calibração da câmera); Classificador de probabilidade de estrada (classe de estrada); Etapa de segmentação de imagem para destacar obstáculos somente na área de estrada, sem precisar verificar o restante da cena
[9]	2010	x	Asfalto, terra e variações	Detecta os “pontos de fuga”, “pontos de horizonte” (<i>vanishing points</i>); Detecta as bordas da estrada; Segmenta a estrada com base nas bordas encontradas
[10]	2010	x	Qualquer tipo de terreno	Treinamento e aprendizagem em tempo real; SVM (<i>Support Vector Machines</i>) para a detecção de estrada
[11]	2009	x	Asfalto (estradas urbanas)	Segmentação de Imagem; Kalman Filter; Visão Estéreo

[12]	2008	x	Asfalto	Segmentação de Imagem; Primeiro o identificador da estrada é treinado com um algoritmo de ensino supervisionado; Em seguida a região de estrada é detectada, combinando cálculos de probabilidade e informação visual, utilizando algoritmos de segmentação de imagem
[13]	2009	x	Asfalto	Detecção de Bordas
[14]	2011		Asfalto	Detecção de estrada com base na iluminação invariante na cena; Rotulação dos pixels como estrada ou não-estrada
[15]	2011		Asfalto	Extensão do algoritmo de amostragem utilizado em Filtro de partículas; Modelagem de estrada a partir de múltiplas hipérboles
[16]	2008		Asfalto	Detecta os “pontos de fuga”, “pontos de horizonte” (<i>vanishing points</i>); Detecta as bordas da estrada

[17]	2010		Asfalto	Pré-processamento: remoção de ruído na imagem, extração dos candidatos a faixas de marcação de estrada; Métodos utilizados na primeira etapa: corte da região de interesse, redução do ruído com uma Gaussiana, alongamento de histogramas, extração de gradientes, limiarização para detecção de bordas, marcação de linhas com reconhecimento de padrões, operações morfológicas
[18]	2011		Asfalto	Visão Estéreo; Filtro de Partículas; Detecção de Bordas
[19]	2013		Asfalto	Lógica Fuzzy (detecta <i>pixels</i> de estrada por meio de agrupamento Fuzzy); Detector de Bordas (Canny ou Sobel); Fuzzy c-means modificado
[20]	2012		Asfalto	Extração de características de pista; <i>Gaussian Sum Particle Filter</i> ; Pontos de fuga

[21]	2012		Asfalto	Detecção de Bordas por cor (onde ocorre alteração da cor de estrada para não-estrada); Detecção de Bordas por Limiarização
[22]	2010		Asfalto	Detecção de Bordas (para encontrar as marcações na pista); Operador <i>Laplacian of Gaussian</i> ; Bordas são detectadas quando a magnitude do gradiente é máxima
[23]	2012	x	Flexível (tipo de estrada, condição climática e iluminação)	Descritor de características baseado em <i>pixel</i> ; Detecção de marcações na pista; Filtro de partículas para rastrear as marcações na pista
[24]	2009		Asfalto (cenário urbano e não urbano)	Visão Estéreo; Detecção baseada em características;
[25]	2006		Asfalto	Detecção e rastreamento de bordas (meio-fio)
[26]	2011		Asfalto (cenário urbano e rural)	Filtro de Kalman; <i>Multi-Target Tracking Algorithm</i>

[27]	2010		Asfalto (cenário urbano)	Detecção de caminho e Detecção de Obstáculos; Visão Estéreo; Reconstrução 3D; Os dados 3D obtidos são transformados em um Mapa de Elevação Digital; Dois classificadores: 1) baseado em intensidade, 2) baseado em superfície de estrada; Classifica os pontos como sendo de estrada ou obstáculos
[28]	2008		Asfalto	Detecção e rastreamento baseado em um filtro de partículas de condensação; Detecção de pontos de fuga; Imagem é dividida em tiras horizontais que são analisados separadamente, mas levando em consideração as tiras vizinhas; Detecta os contornos da estrada (bordas)
[29]	2009		Asfalto (diferentes condições de iluminação. dia e noite)	Extraí (detecta) pontos na imagem com características de marcação de pista (linhas, faixas); Lógica Fuzzy; Calibração automática da câmera utilizada para corrigir parâmetros em momentos de vibração do carro

2.3.1.3 Considerações Sobre os Trabalhos Correlatos e Proposta

A revisão dos trabalhos correlatos para a detecção de caminho na navegação autônoma focou trabalhos que utilizam visão passiva (câmeras), excluindo assim trabalhos que utilizavam outros sensores que não uma câmera, por exemplo: lasers. Pode-se notar que a grande maioria dos trabalhos focou em um padrão de estrutura no caminho, ou seja, não avaliando situações onde ocorreram alterações bruscas de cor no caminho. Entretanto alguns trabalhos aplicaram uma abordagem de detecção de caminho adaptativa, que é o mesmo foco da proposta apresentada nesta dissertação. A diferença entre esses trabalhos e a proposta aqui apresentada está na forma como foram avaliados os trabalhos, já que os trabalhos correlatos encontrados não fizeram uma avaliação em tempo real da detecção do caminho, ou seja, em muitos casos a detecção pode ser forçada, pois o veículo estará seguindo a rota independente da detecção.

3 PROPOSTA

Para realizar a detecção de caminho, métodos como segmentação de imagem e detecção de bordas são utilizados, definindo desta forma, a parte da imagem que é considerada um caminho que pode ser seguido, e separando do restante da imagem. E isso deve ocorrer em tempo real e de forma adaptativa, ou seja, não importando as mudanças no caminho (cor, tipo de terreno, etc.) a detecção deve continuar acontecendo.

A detecção do caminho é testada em um protótipo com uma câmera de captura fixada na parte frontal, o controle do protótipo será feito com base nos resultados obtidos das imagens de detecção do caminho, e então os valores corretos de direção e aceleração devem ser enviados ao veículo, possibilitando dessa forma sua navegação autônoma e também a validação do trabalho.

3.1 REQUISITOS

É importante, antes de qualquer explicação sobre o projeto como um todo, levantar alguns requisitos importantes para o bom andamento do trabalho. Os requisitos são apresentados em forma de tabela. Foram levantados requisitos funcionais e não-funcionais para o sistema. O objetivo da análise e gerenciamento de requisitos é identificar organizar e documentar as condições ou capacidades que o sistema a ser desenvolvido deve estar de acordo. Os requisitos encontrados e levantados para esse projeto são divididos em duas categorias: Funcionais e Não-Funcionais e estão listados em duas tabelas.

3.1.1 Requisitos Não-Funcionais

Tabela 2 – Requisitos Não-Funcionais

Código	Título	Descrição
RNF01	Sistema Operacional	Desenvolvido no SO Linux, distribuição Ubuntu 14.04.
RNF02	Compilador	Compilador GCC Linux.
RNF03	Linguagem de Desenvolvimento	Utilizado a linguagem de programação C++. Por sua facilidade e bom desempenho, além de ter boa compatibilidade com a biblioteca OpenCV.
RNF04	Biblioteca de Visão Computacional	Utilizado a biblioteca OpenCV (biblioteca com diversos métodos de VC) para auxílio no desenvolvimento.
RNF05	IDE Utilizada	Utilizado o NetBeans como ambiente de desenvolvimento. Por ajudar na organização e facilitar a ligação entre as bibliotecas utilizadas.
RNF06	Interface Gráfica	Utilizado o QtWidgets para o desenvolvimento da Interface Gráfica. Por sua compatibilidade com o C++ além de diversos métodos que facilitam o desenvolvimento.
RNF07	IDE Gráfica	Utilizado o Qt Design como ambiente de desenvolvimento da interface. Devido a facilidade de construção dos layouts e de compatibilidade com o NetBeans.
RNF08	Veículo para Teste e Avaliação	O sistema será testado e avaliado em um veículo.
RNF09	Comunicação entre Código e Veículo	A comunicação e comandos entre código e veículo será feita com a utilização de Arduino e servo motores.

RNF10	Biblioteca para Comunicação C++ e Arduino	A biblioteca utilizada para o C++ se comunicar com o Arduino é a LibSerial.
RNF11	Câmera	Será utilizada uma câmera (GoPro Hero 3) na parte dianteira do veículo, capturando imagens frontais para a detecção de caminho.
RNF12	Entrada USB 2.0	A máquina que executará o sistema precisa ter entrada USB 2.0 para receber a imagem da câmera pelo Dispositivo de Captura de Vídeo.

3.1.2 Requisitos Funcionais

Tabela 3 – Requisitos Funcionais

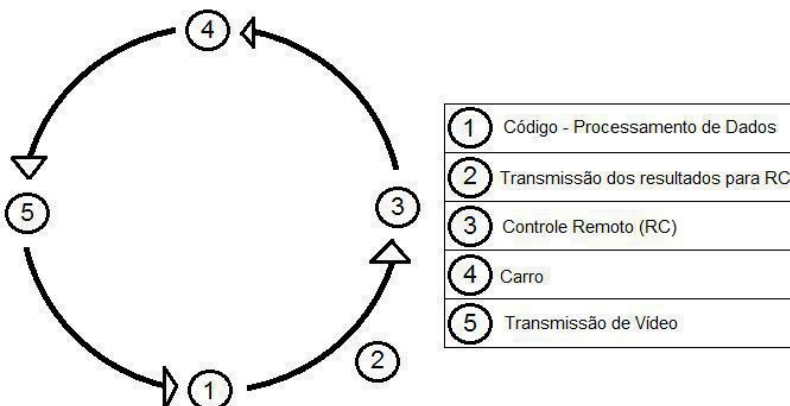
Código	Título	Descrição
RF01	Abrir Vídeo	O sistema deve possibilitar a abertura de vídeo gravados.
RF02	Abrir Câmera	O sistema deve possibilitar a abertura de imagem capturada por uma câmera em tempo real.
RF03	Dados de Saída	Deve apresentar na interface os dados de saída, que são: Direção e Aceleração passados ao veículo.
RF04	Média dos Pontos de Entrada	É feita uma coleta de pontos de entrada onde há estrada (com informações de cor). E então uma média para posterior comparação com o restante da imagem.
RF05	Comparação e Segmentação da Imagem	Com uma métrica de similaridade e um limiar, a média dos pontos de entrada (estrada) é comparada com cada pixel da imagem para verificar o que também faz parte da estrada.

RF06	Pintar a Estrada	Os pontos considerados de caminho são pintados de outra cor para destaque na imagem e avaliação humana.
RF07	Envio de Dados para o Veículo	Após o Processamento dos métodos para a detecção de caminho, o sistema deve passar valores de direção e aceleração para o veículo se movimentar.
RF08	Cálculo	Os dados que serão enviados ao veículo devem ser calculados previamente conforme o caminho disponível a sua frente.
RF09	Fechar Interface	O sistema deve poder ser fechado através da interface gráfica.

3.2 ETAPAS DE DESENVOLVIMENTO DA PROPOSTA

Inicialmente o desenvolvimento da proposta foi dividido em duas partes principais, uma sendo o hardware e todos os equipamentos utilizados e suas funcionalidades, e outra sendo o software e os métodos de segmentação e de controle do veículo de testes. Com a intenção de melhor explicar a proposta dessa dissertação, cinco etapas foram separadas (Figura 31) e explicadas detalhadamente.

Figura 31 – Etapas de desenvolvimento da Proposta.



O protótipo automobilístico (4) possui uma entrada de dados, onde recebe os comandos a serem executados; e uma saída de informações: a transmissão de vídeo (5), que será usada como base para a determinação dos comandos a serem transmitidos. Em (1) é processado o algoritmo que, a partir dos *frames* (capturados pela câmera fixada em 4), determina quais ações deverão ser executadas por (4). O envio de comandos para (4) é feita através de (3), o controle remoto. Esse teve algumas de suas conexões alteradas para permitir que (1) o controlasse. Além de complementos (como Arduino), que são representados como (2). Assim é completado o ciclo de informações que formam o projeto: o código (1) envia os comandos para o carrinho (4), através de (2) e (3), fechando com (4) transmitindo para (1) as filmagens feitas do que há a frente, através de (5).

3.3 CÓDIGO - PROCESSAMENTO DOS DADOS

O programa desenvolvido foi executado em um notebook. Onde são recebidos os *frames* obtidos através da câmera do carrinho e de onde são enviados os dados que coordenam a reação do carrinho. É essencial que tudo ocorra em tempo real, pois a filmagem recebida é utilizada como base para extrair informações, que permitem a detecção de caminho e de obstáculos, auxiliando nas tomadas de decisões executadas pelo carrinho.

3.3.1 Detecção do Caminho - Segmentação da Imagem

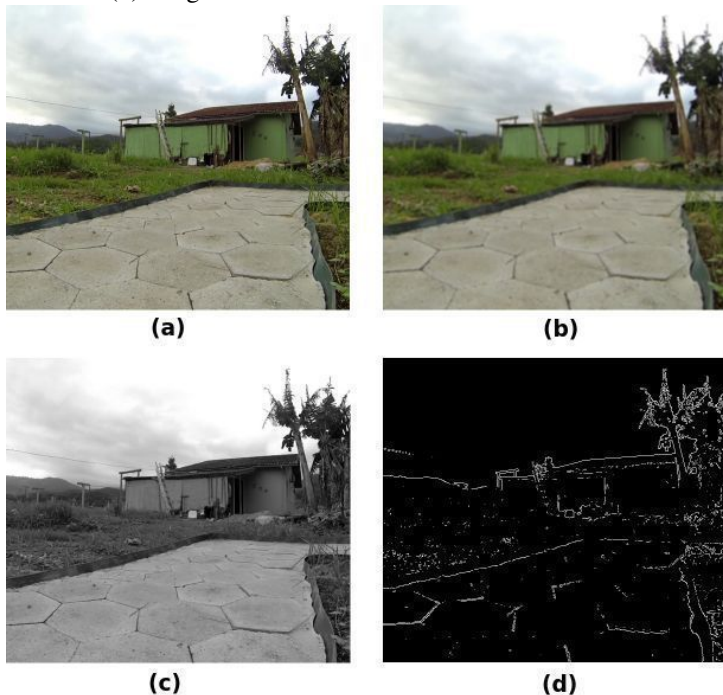
Para realizar a detecção do caminho é aplicado a segmentação de imagens a cada *frame* enviado pela câmera. A ideia consiste, basicamente, em dividir a imagem em várias regiões, onde cada uma dessas regiões contenha um índice, separando dessa forma a imagem em diversas áreas indexadas. Para que a segmentação realmente defina uma parte da imagem como o caminho, foi aplicado primeiramente uma detecção de bordas (*canny edge*), permitindo então a utilização de um algoritmo de preenchimento de regiões (*floodfill*). Pode-se separar o processo em três etapas, pré-processamento, segmentação da imagem e localização do caminho.

Ao receber o *frame* é feito o pré-processamento da imagem, buscando reduzir os ruídos sem que haja perdas das descontinuidades que representam as fronteiras das regiões, o que permitirá que a área detectada como caminho, não ultrapasse seus limites e segmente outras regiões próximas a área como fazendo parte do caminho de forma errada. Dessa forma, é utilizado o algoritmo de detecção de bordas, Canny (CANNY, 1986).

A detecção de bordas deve pegar todas fronteiras possíveis na imagem, ou seja, quando ocorre mudanças bruscas no valor de intensidade entre um *pixel* e outro (A imagem deve estar em tom de cinza para que o algoritmo Canny funcione trabalhando em seus níveis de intensidade), isso tentando ao máximo evitar que ruídos na imagem sejam considerados bordas.

A detecção de bordas Canny funciona de forma que os ruídos não atrapalhem o resultado final e pode ser dividida em 5 etapas. Primeiro é feita a suavização da imagem para remover os ruídos, utilizando para isso o filtro Gaussian. Então são procurados os gradientes da imagem, as bordas são definidas onde a intensidade em tons de cinza da imagem tem alterações mais bruscas. Após encontrar os *pixels* onde há grande mudanças de intensidade, é feita uma verificação nos gradientes, onde são considerados cada *pixel* e seus vizinhos, mantendo somente valores que sejam realmente de borda. Ainda, para garantir que somente *pixels* de bordas sejam marcadas, são utilizados dois limiares, um de valor alto e outro de valor baixo, quando o valor do gradiente de borda do *pixel* for maior que o limiar alto, ele é considerado como uma borda forte, se ficar entre o limiar baixo e alto, é uma borda fraca, e se ficar abaixo do limiar baixo, é excluído. Por fim, os *pixels* que são considerados fortemente como borda ficam na imagem, e os que forem considerados como bordas fracas, somente ficarão na imagem caso façam fronteira com *pixels* considerados fortes (P.K. Kalra, 2009).

Figura 32 – (a) - imagem original. (b) - imagem suavizada. (c) - imagem em ton de cinza. (d) imagem de bordas.



Após o pré-processamento da imagem, inicia-se a segmentação, que visa dividir a imagem em áreas homogêneas distintas. Para isso, é criado um mapa de indexação que mantém um determinado valor correspondente para o qual a área de algum dado *pixel* pertence. O método de segmentação é baseado em uma medida de similaridade para cada *pixel* adjacente, levando em consideração uma rede de gradientes responsável por determinar restrições durante o processo de fusão dos *pixels*, garantindo desta forma que nenhuma região será unida fora dos seus limites, após isso é aplicado o método de crescimento de regiões.

O algoritmo inicia com o mapa da rede de gradientes, onde o ponto semente (*seed points*) inicial é também definido como o ponto de partida do algoritmo. Uma rede de similaridades é preenchida sobre o mapa de gradientes, onde os níveis de intensidades dos *pixels* são medidos de acordo com uma função de similaridade definida por (WANGENHEIM et al., 2007), permitindo extrair informação bidimensional do caminho navegável. Após ex-

traída, informação temporal também é analisada, onde a rede de gradientes é expandida para os *frames* anteriores, proporcionando um esquema de segmentação robusto. Este é dado por:

$$D^2 = \sum_{k=1}^n \Delta C_n \quad (3.1)$$

onde n é a dimensionalidade do espaço de cor utilizado, e

$$\Delta C = \|C_p - C_s\| \quad (3.2)$$

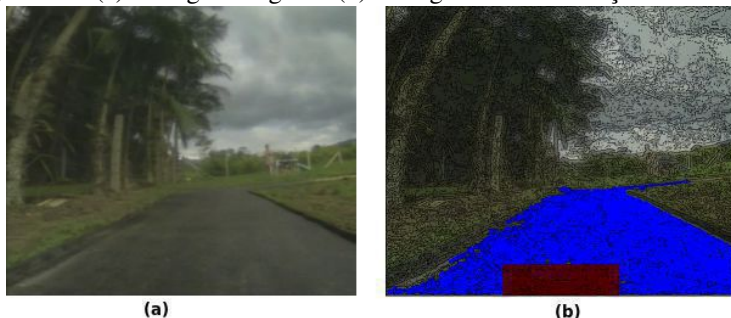
é a intensidade da cor invariante para um espaço de cores específico, calculado sobre os *frames* anteriores. ΔC é a diferença entre o *pixel* analisado e o ponto semente para um dado canal k , onde $\Delta D < th$ é determinado por um valor limiar onde os *pixels* são considerados como sendo da mesma área. Quando essa condição é satisfeita, o mesmo procedimento é aplicado ao *pixel* atual.

A segmentação por crescimento de regiões utilizadas nesta dissertação tem os *seed points* adaptativos, o que significa que os valores dos pontos de entrada são sempre alterados para novos valores de *pixels* de interesse. Isso permite continuar facilmente a segmentação e indexação das áreas caso ocorra alguma alteração na intensidade dos *pixels* no caminho, afinal a imagem de bordas deve limitar o caminho do restante da imagem, evitando que o caminho seja reconhecido de forma incorreta em circunstâncias normais. Se os pontos de entrada (*seed points*) fossem estáticos, poderia haver dois problemas: o caminho não ser inteiramente reconhecido por causa de possíveis alterações de luz, sombras ou pequenas alterações ao longo do caminho, e a segmentação extrapolando os limites do caminho. Quando mais nenhum *pixel* pode ser adicionado a região de interesse, o algoritmo busca pelo próximo ponto semente. O mapa de áreas é temporariamente verificado buscando o primeiro ponto sem área definida, quando encontrado o procedimento anterior é aplicado a ele com a área correspondente. Esse processo é repetido até que todos os *pixels* da imagem pertençam a alguma área.

Por fim, após a imagem segmentada, ocorre a localização do caminho navegável. O objetivo é encontrar qual das áreas anteriormente obtidas é o caminho na imagem. Esta etapa é realizada considerando que o caminho navegável existirá dentro de uma região de interesse na imagem, apresentada por um retângulo vermelho (Figura 33 (b)) utilizado nos primeiros *frames*. Baseado nessa calibração inicial, o método de rede de gradientes previamente explicado é utilizado para encontrar o caminho navegável temporariamente. Por outro lado, sabe-se que em aplicações visando detecção de caminho e navegação autônoma a câmera fica localizada na parte frontal do veículo,

pode-se afirmar então que o caminho sempre vai existir dentro dessa área. O retângulo com a região de interesse pode ser adaptado e movido conforme o cenário utilizado.

Figura 33 – (a) - imagem original. (b) - imagem com a detecção de caminho.



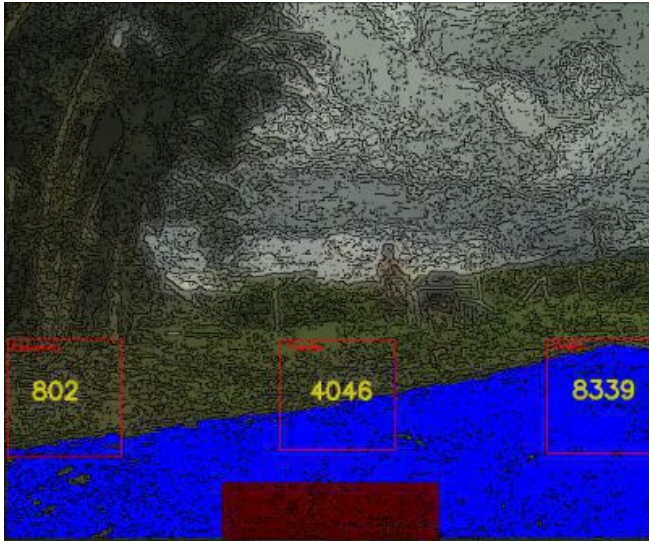
3.3.2 Análise da imagem - Controle do Veículo

Com a imagem segmentada e o caminho navegável detectado, é iniciado a análise dessa imagem para o controle do veículo. Para isso, são definidas três regiões de interesse (*region of interest* - ROI) na imagem, uma para verificar quando o caminho inicia uma curva para a esquerda, outra para verificar quando o caminho inicia uma curva para a direita e outra para verificar quando deve-se continuar reto. Essas regiões podem ser alteradas dependendo do cenário e da posição da fonte de captura.

Com as três regiões de interesse definidas, é feita uma varredura a cada *frame* recebido com o caminho detectado em cada uma das ROI's, a região que estiver com a maior quantidade de *pixels* pertencentes a área do caminho é considerada como a região que o veículo autônomo deve seguir. Por exemplo, se das três regiões de interesse, a região que representa a verificação de curva para direita for a que tiver mais *pixels* pertencentes ao caminho, então é enviado comando ao carrinho para que ele vire à direita, e isso funciona da mesma forma para as outras duas regiões de interesse.

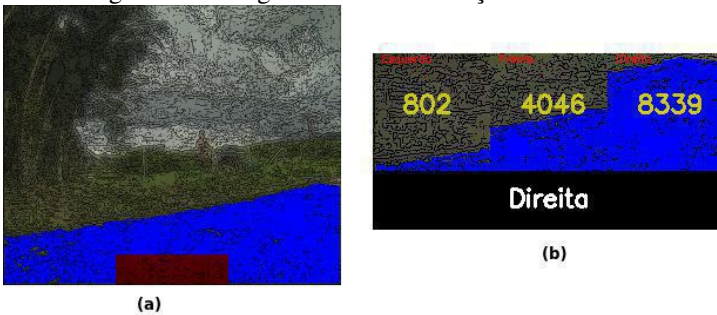
Após os primeiros experimentos, verificou-se a necessidade de adaptar o controle. Como a região de interesse relacionado a reta é que constantemente possui mais *pixels* do caminho, ocorria em situações em que o veículo estava finalizando uma curva, de a região de interesse de reta ter mais *pixels* de caminho do que a região pertencente a curva (direita ou esquerda), fazendo com que o veículo não completasse totalmente a curva. Então, após verifica-

Figura 34 – Regiões de interesse para o controle.



ção dos resultados e análise do cenário, foi colocada mais uma condição para as curvas, considerando essa situação que se repetia. Então, quando a região de interesse considerada como frente é a de maior quantidade de *pixels* de caminho, ainda é verificar a diferença entre sua quantidade de *pixels* com as regiões de curva, e estando dentro de um limiar definido, deve-se continuar a curva. Esses valores são então passados ao Arduino.

Figura 35 – Imagem com as informações de controle.



3.4 TRANSMISSÃO DOS RESULTADOS PARA O RÁDIO CONTROLE

A transmissão de comandos do código, processado no notebook, para o rádio controle Spektrum DX6i é estruturado com o auxílio dos seguintes itens:

- **Ângulos:** Os valores do código de Visão Computacional (VC) são enviados para o arquivo da porta serial (criado automaticamente, assim que o Arduino é conectado à máquina), chamado `ttyACM0`, criado para a comunicação com o Arduino; Esses valores determinam as posições, em graus, que os servos correspondentes deverão ficar.
- **Plataforma de prototipagem eletrônica:** O Arduino, que recebe os ângulos e disponibiliza essas variáveis em pinos, onde são transmitidos para os servos.
- **Protoboard e jumpers:** Responsáveis pela conexão entre os servos e o Arduino.
- **Servomotores 180°:** São dispositivos que abrangem 180 graus com alta precisão, são utilizados dois desses para posicionar os respectivos potenciômetros.
- **Potenciômetros:** Cada potenciômetro permitirá uma corrente com amplitude específica causando uma reação no carrinho. Sendo um responsável pelo direcionamento e o outro pela velocidade.

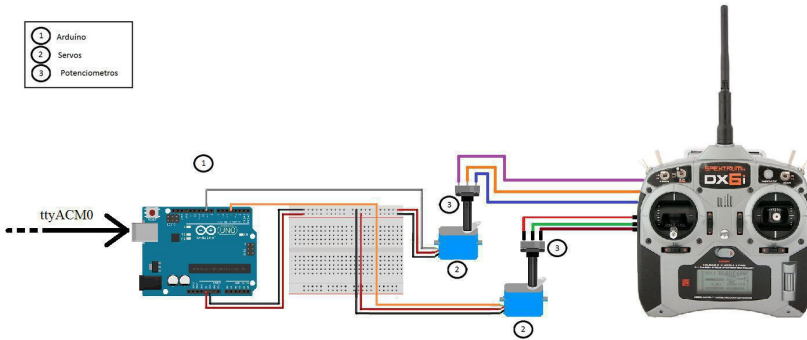
Com essa estrutura montada, o sistema registrará os valores em que os servos devem se posicionar. Assim que configurados, conseqüente e paralelamente os potenciômetros são ajustados na mesma posição, comandando o RC para enviar o estado em que o carrinho deverá ficar. Como representado pela Figura 36.

3.4.1 Transmissão de Dados do Notebook ao Arduino

Para o discernimento do carrinho é necessário que o código constantemente envie os parâmetros necessários. Como o Arduino não possui processador como um PC ou um Notebook, isso fica dependendo do notebook. O desafio então, foi realizar essa comunicação, mais especificamente entre o código em C++ e Arduino.

Para fazer a comunicação e enviar os dados para o Arduino, basicamente é feita escrita no arquivo criado pela USB do Arduino como porta

Figura 36 – Imagem com as informações de controle.



serial (arquivo: ttyACM0). Finalmente, utilizando a LibSerial para o C++, foi possível compilar o projeto utilizando a flag “-lserial” no momento de compilar, exemplo: “g++ ‘nomeArquivo’.cpp -lserial -o”. Deve-se lembrar de manter a taxa de transferência de *bits* por segundo igual a esperada pelo código no Arduino, neste projeto está sendo utilizado uma taxa de 9600.

Foi testado a comunicação passando valores do código C++ para o Arduino e nisso o servo responde aos comandos perfeitamente. Também foi realizado um teste para verificar a passagem dos valores para o Arduino, para garantir que é passado um valor de cada vez, ou seja a porta serial limpa a cada valor enviado. Pode-se contatar com os testes de comunicação, que é possível enviar valores separados para cada servo, direção e aceleração.

3.4.2 Estrutura de Transmissão do Arduino ao RC

Um carro autônomo precisa verificar a velocidade e a direção ideal para não colidir com um carro ou motocicleta a frente no trânsito; para fazer uma ultrapassagem ou curvas; para manter um perímetro de segurança de pedestres em movimento, ou qualquer outro obstáculo. Nessa dissertação, o controle do veículo é realizado com base em Processamento Digital de Imagem (PDI) e Visão Computacional (VC) nos *frames* recolhidos pela filmagem feita em tempo real, frontal ao carrinho, além das métricas para a escolha dos ângulos que comandam os servos.

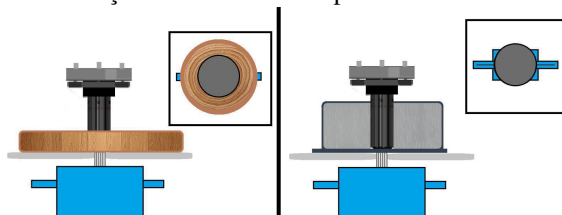
Visando verificar como cada valor (ângulo) influencia na movimentação dos servos, foram realizados testes e com os dados obtidos, a construção de uma base de dados. Os testes consistiram basicamente de filmagens das rodas do veículo e estudo das alterações para cada um dos 180 ângulos envi-

ados para cada servomotor, sendo um responsável pelo controle da direção e o outro pelo controle da aceleração.

O primeiro teste permitiu três observações. A primeira foi a verificação de ângulos irrelevantes, ou seja, que não causam alteração no estado da roda. A segunda foi que a tentativa de fazer a análise visual dos ângulos não foi bem sucedida, pois não estabelece parâmetro algum de precisão. Desta forma, a análise das alterações na roda ocorre através de PDI realizado nas imagens, como por exemplo subtração de *frames* buscando alterações na imagem. A terceira observação é que o encaixe entre os potenciômetros e os servos estavam frouxos, desta forma o valor passado computacionalmente ao servo não necessariamente gerava a resposta exata. Neste caso surgiu a necessidade de refazer os encaixes, buscando maior precisão nos dados.

A comunicação entre o servo e o potenciômetro é estabilizada mecanicamente, sendo inicialmente projetada com um disco de madeira, fixado na hélice do servo, envolto ao pino do potenciômetro. De acordo com o verificado no teste, esse encaixe permite um pequeno deslocamento entre as peças, o que é inaceitável. Então foi desenvolvido uma estrutura com lâmina de alumínio fixada nas hélices dos servos e encaixada na fenda do pino do potenciômetro. É importante salientar que houve uma preocupação em fazer a lâmina ocupar todo o espaço entre as hastes, evitando ainda mais imprecisões.

Figura 37 – Comparação entre a estrutura inicial e a posteriormente desenvolvida para a comunicação entre o servo e o potenciômetro.



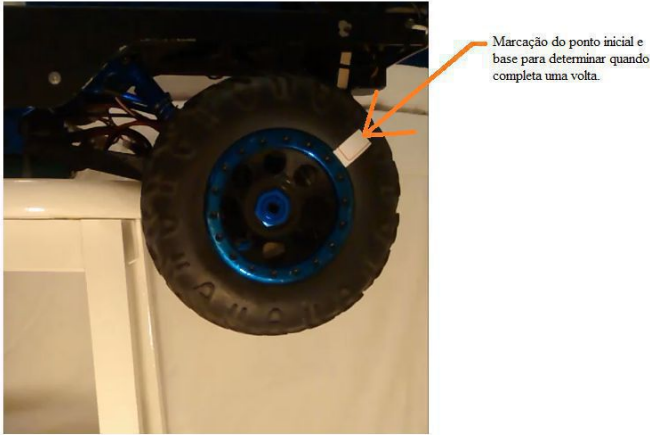
• Listagem dos Ângulos que Comandam a Aceleração:

Uma das funções de um motorista é verificar a velocidade em que o automóvel está e se é possível manter, aumentar, ou se deve diminuir a velocidade. Para isso ser aplicado a um carro autônomo, primeiramente é necessário que o sistema tenha um banco de dados que permita verificar quais dados deve enviar para que o carro alcance a velocidade calculada.

Baseado na estrutura montada, testes foram feitos para averiguar quais as velocidades alcançadas por cada ângulo do servo. A coleta desses dados foi feita através de filmagens, em que o servo fica, por 10 segundos, posicionado

num ângulo. A partir disso e da medida da circunferência do pneu, é possível calcular a velocidade que cada ângulo proporciona.

Figura 38 – Marcação para teste de Aceleração.



Com os vídeos dos testes para os valores de aceleração, foram contabilizadas as voltas realizadas por cada ângulo, gerando assim o arquivo com o armazenameto da velocidade em cada ângulo possível. Durante a contagem, devido ao aumento da velocidade, houve dificuldade de contabilizar as voltas dos ângulos superiores a 128, foi refeito o teste desse ângulo até 180 com um novo indicador no pneu. Na segunda avaliação feita, foi possível perceber grande distinção na velocidade dos ângulos analisados em relação ao primeiro teste. Isso ocorre devido a diferença da carga da bateria. Esse fator tem grande impacto e influência no controle do veículo. Com isso, foi sugerido gerar uma equação com as variáveis ângulo e carga de bateria para resultar na velocidade que será causada no carrinho ao aplicar o ângulo (deduz-se que a carga da bateria e a velocidade são diretamente proporcionais).

Observação: Uma vantagem ao aplicar num automóvel será de não ter essa questão de velocidade, pois não importa se tiver 100% ou 40% de combustível, as velocidades a serem atingidas serão as mesmas.

Assim foi realizado outro teste, a partir da bateria totalmente carregada, foram filmadas cinco vezes cada ângulo, por 10 segundos, dos 180. Para contabilizar a velocidade proporcionada por cada ângulo na mesma faixa de bateria, como para comparar a diferença causada pela carga da bateria.

Durante as filmagens foi idealizado uma máscara que auxiliasse na contabilização das voltas. Desenhada no programa *After Effects* (versão de demonstração), no sistema operacional Windows. Finalmente com os vídeos

com as máscaras, foram contabilizadas as voltas e calculado a velocidade, por exemplo:

No ângulo 01 é realizada -23,95 voltas em 10s. O sinal representa a direção, neste caso é para trás. Como a circunferência do pneu mede 57,85 cm, ao multiplicar:

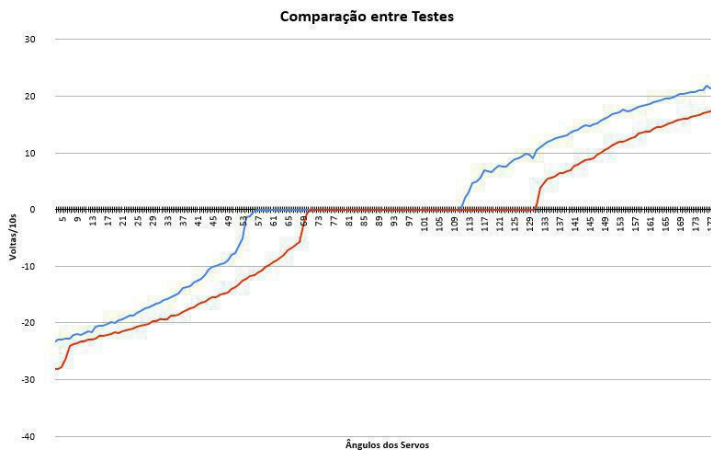
$$-23,95 * 57,85 = -138,5075 \frac{cm}{10s} \quad (3.3)$$

Dividindo por 10:

$$\frac{-138,5075}{10} = 13,85075 \frac{cm}{s} = 1,385075 \frac{m}{s} = 4,9878 \frac{km}{h} \quad (3.4)$$

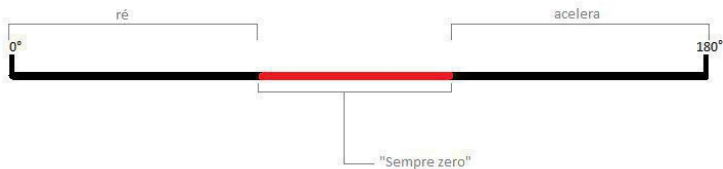
Juntando todos os valores em um gráfico, resultou no gráfico da Figura 39. Nele é possível perceber o deslocamento da velocidade para a direita de acordo com o uso da bateria; além da mudança de proporções entre os ângulos de comandos dos servos.

Figura 39 – Gráfico de comparação entre teste: voltas/10 segundos de cada ângulo.



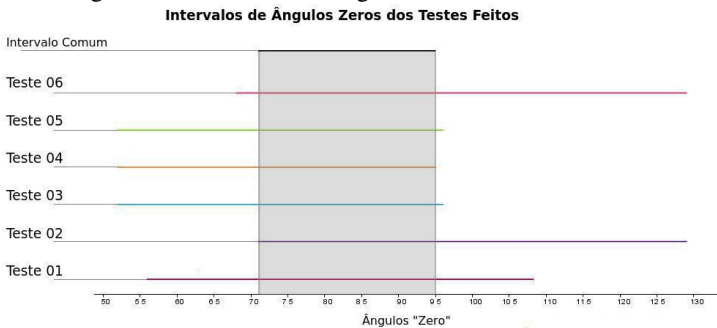
Com esses dados levantados e documentados facilita no envio de valores de aceleração ao veículo, onde pode-se evitar o uso de valores que ficaram na faixa de “ângulos zeros”, ou seja, valores que independente da carga da bateria, mantém o veículo parado. Após iniciado a aceleração do veículo os *frames* são verificados constantemente, possibilitando acelerar, frear e caso precise retroceder, irá passar para os valores menores ao intervalo dos “ângulos zeros” (Figura 40).

Figura 40 – Barra de valores representando as reações causadas por cada ângulo.



Assim foram comparados os intervalos de ângulos em que são feitas zero voltas por segundo (Figura 41) resultando em 71 a 95.

Figura 41 – Intervalo de ângulos zeros nos testes feitos.



- **Listagem de Ângulos que Comandam o Direcionamento:**

O discernimento em fazer o automóvel desviar de obstáculos, redirecionar na pista, movimentar-se em curva e manobrar é essencial e não pode ser ignorado por um carro autônomo. E assim como os valores de aceleração, para alcançar esse discernimento em relação aos valores de direção, foi preciso fazer testes para analisar quais as informações que o sistema deve enviar ao veículo para fazê-lo seguir em determinadas direções.

Os dados analisados são as posições em que as rodas ficam quando os servos recebem o valor de um ângulo, dos 180 que o servomotor possibilita. Então foram feitas filmagens da roda (com visão superior ao veículo), sendo verificado cada ângulo, a cada dois segundos, de maneira crescente.

As filmagens foram realizadas utilizando a seguinte estrutura: uma caixa de isopor, para evitar que luzes e sombras interferissem na filmagem; dentro dessa caixa foi colocado o veículo em cima de uma base alta; uma câmera em cima de uma fenda feita na caixa; uma aste foi colada no pneu filmado, possibilitando a evidencia da mudança de direção; e um cronômetro, assegurando em que momento (tempo) cada ângulo foi alterado (Figura 42).

Figura 42 – Estrutura de filmagem para o teste de direcionamento.

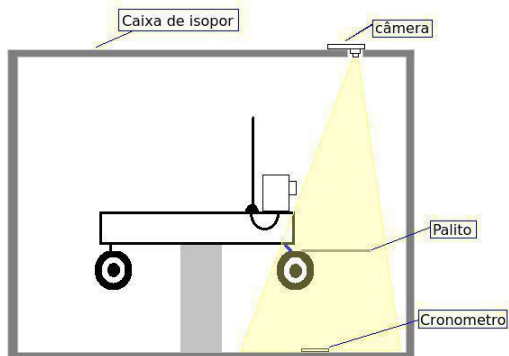
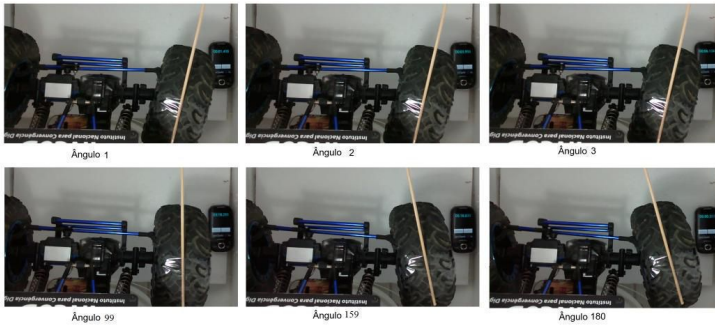


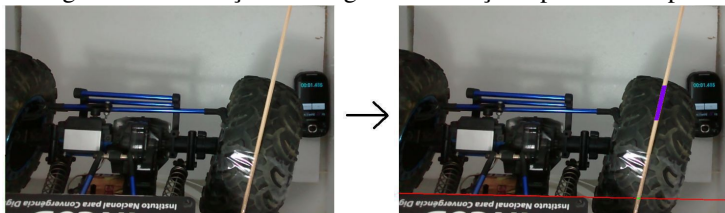
Figura 43 – Algumas das imagens resultantes de um teste realizado e apresentação do tempo de acordo com o ângulo do servo.



Foram realizadas 30 gravações, para verificar o impacto que a carga da bateria causa no desempenho do direcionamento. Já que nos testes de aceleração foi verificada uma dependência muito grande. De cada filmagem, foram capturados os *frames* em momentos respectivos a cada um dos 180 ângulos. Assim possibilita verificar exatamente a direção tomada pelo veículo, pois as medidas angulares resultantes, entre a linha frontal do veículo e a haste colocada na roda, que ficarem menor que 90 graus conduzem para a direita; as maiores, para a esquerda; e iguais a 90 pra frente. Como há mais de cinco mil imagens, foi desenvolvido um código para computar esses resultados. Ele foi dividido em 3 etapas:

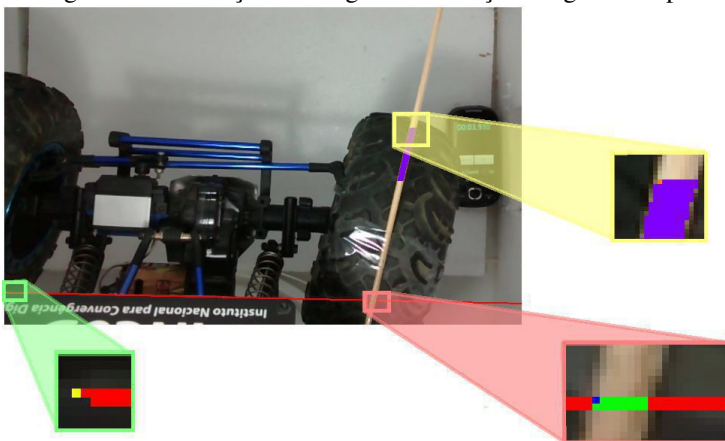
- Primeira etapa - Destacar regiões de interesse: Foi desenhado uma imagem como molde para a linha frontal do carro. Somando ela com cada uma das imagens permite demarcar essa área. Garantindo que todas tenham a mesma base e com mesma inclinação, assim qualquer mudança detectada seja, realmente, pela mudança de direção do pneu. Enquanto era impresso a barra frontal, foi adicionada uma condição para que pintasse de verde os pixels com cores correspondentes ao do palito, com isso era destacada a intersecção. Além da barra frontal e a intersecção, é necessário destacar uma região do palito, para isso foi demarcada uma região que destacasse onde houvesse o palito.

Figura 44 – Detecção dos ângulos de direção - primeira etapa.



- Segunda etapa - Marcar os pontos: Com as três áreas de interesse destacadas, resta destacar o primeiro ponto de cada uma delas para usar as coordenadas e calcular o ângulo entre eles.

Figura 45 – Detecção dos ângulos de direção - segunda etapa.



- Terceira etapa - Quantificar os graus: tendo as posições dos pixels demarcados como parâmetro, foi calculado o ângulo referente a intersecção.

Figura 46 – Detecção dos ângulos de direção - terceira etapa.



Com todos os valores tabelados, foi confirmado que a bateria é bastante influente nas direções tomadas (Figura 47). Para lidar com essa situação, foi realizada uma média aritmética com as medidas correspondentes a cada ângulo do servo. Apesar de não ser a melhor solução, pelo menos já facilita a tomada de decisão de qual ângulo usar para a direção desejada.

Analisando a tabela verifica-se que as médias apresentam os ângulos do servomotor entre 1 e 71 comandam o carrinho para a direita, e de 78 a 180 para a esquerda. Porém essa exatidão está longe de ser alcançada, pois o intervalo que mantém o carrinho reto (de 72 a 77) ficou muito deslocado de um teste para outro, inadequando. Sendo um desafio para a aplicação, já que o teste é iniciado com a carga da bateria do carrinho ao máximo.

Figura 47 – Comparação entre o primeiro e o último teste feito, relacionando a direção (em graus) com cada um dos 180 ângulos do servo motor.

Ângulos	Teste 01	Teste 30
1	101,859	101,69
2	10,859	101,69
3	101,859	101,456
4	101,62	101,456
5	101,62	100,989
6	101,141	100,989
7	101,141	100,755
8	101,141	100,755
9	101,141	100,755
10	101,141	100,285
11	100,661	100,285
12	100,661	100,285
13	100,42	100,285
14	100,42	99,814
15	100,42	99,814
16	100,179	99,814
17	99,9375	99,814
18	99,9375	99,814
19	99,9375	99,3411
20	99,6962	99,3411
21	99,6962	99,3411
22	99,4535	99,3411
23	99,4535	98,8669
24	99,4535	98,8669
25	99,2106	98,8669
26	98,7245	98,8669
27	98,7245	98,3913
28	98,7245	98,3913
29	98,7245	98,3913
30	98,237	98,3913
...		
152	79,3693	79,2113
153	79,129	78,9524
154	78,6485	78,6924
155	78,6485	78,6924
156	78,4084	78,4344
157	78,1696	78,4344
158	77,9303	77,4036
159	77,9303	77,4036
160	77,4538	77,4036
161	77,6923	76,8912
162	76,979	76,8912
163	76,979	76,8912
164	76,7427	76,8912
165	76,7427	76,3807
166	76,7427	76,3807
167	76,7427	76,1271
168	76,2705	75,6198
169	75,8002	75,6198
170	75,8002	75,8724
171	75,3317	75,3662
172	75,3317	75,1147
173	75,3317	74,8637
174	74,8652	74,6118
175	74,8652	74,6118
176	74,4006	74,1113
177	74,4006	74,1113
178	74,4006	74,1113
179	74,4006	73,3657
180	73,9681	73,3657

3.5 CONTROLE REMOTO

Para a realização do projeto, é utilizado o controle remoto Spektrum DX6i (Figura 48) com algumas alterações no circuito. Pois como a intenção é de comandar o RC eletricamente, ou seja, não manualmente; foi desconectada a fiação de uma das hastes (*stick*, em inglês) com a placa do RC (*Remote Control*) e soldadas ligações advindas dos potenciômetros.

Figura 48 – Controle remoto Spektrum DX6i.



3.5.1 Rádio Controle

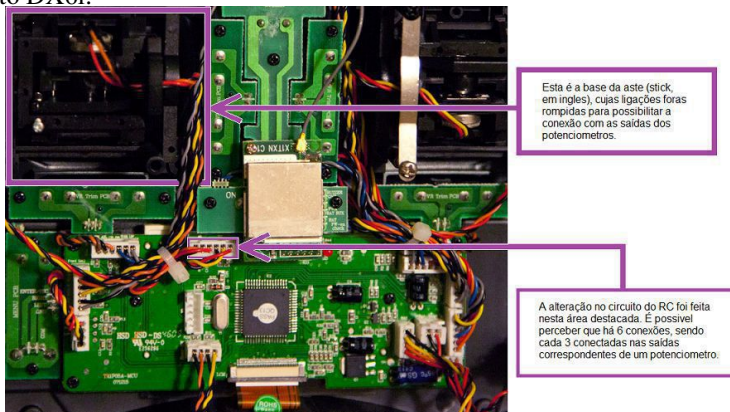
Por rádio controle, são recebidos os comandos que determinam as direções, velocidade e sentido de rotação dos pneus. Funcionando de maneira que as informações são capturadas, por ondas de 2,4GHz, através do DSM2 DualLink AR6200.

Figura 49 – DSM2 DualLink AR6200.



Esse receptor é conectado aos servos, que estão acoplados aos eixos das rodas. Conseqüentemente quando posicionados esses servos os pneus reagem. Para manter a estabilidade é preciso assegurar que os fios de recepção das ondas estejam eretos e sem contato com metais (lateria do carro, no caso), para haver, assim, o mínimo de interferência. Além da perda dos dados, a perturbação na recepção das ondas causa um descontrole nas informações transpassadas, conseqüentemente o carrinho faz movimentos não esperados. O que é inaceitável para um carro autônomo.

Figura 50 – Apresentação das alterações feitas no circuito do Controle Remoto DX6i.



Com essa estrutura, quando os servos rotacionam os potenciômetros ocorre mudança na corrente/tensão, fazendo com que o rádio controle comande o carrinho, respectivamente, a aceleração e o direcionamento a ser tomado.

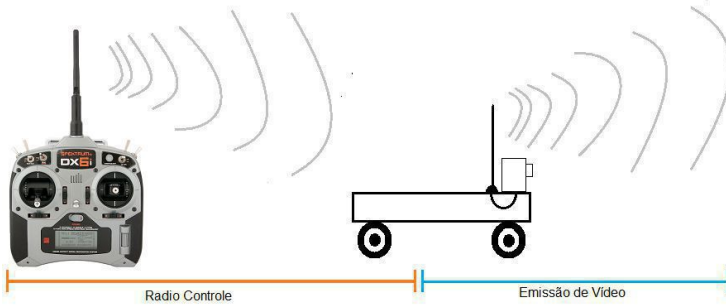
3.6 VEÍCULO

Considerado o ator principal do projeto, pois é com base no bom funcionamento do carrinho que este projeto é encaminhado; é ele quem irá comprovar a validade do código desenvolvido. Já que é o receptor dos comandos enviados pelo algoritmo e o emissor dos parâmetros para o processamento da ferramenta. Para realizar tais funções, como os equipamentos já foram entregues estruturados para o projeto, foi necessário primeiramente analisar a sua estrutura e, conseqüentemente, funcionamento.

Com base nas observações foi verificado que há instalado uma câmera; uma bateria reserva; os emissores do Rádio Controle; um emissor de vídeo; dois conectores para carregar bateria; interruptor; uma antena de 900Mhz, para emitir vídeo; além do motor e todas as suas conexões. Junto a ele veio dois carregadores, cujos os plugues (dois para cada carregador) ficam localizados na lateral do carrinho. Observado o circuito existente, foi possível certificar que um deles está objetivado a fornecer carga à bateria do carrinho, enquanto o outro esta conectado ao emissor de vídeo. Assim, para carregar a bateria reserva do carrinho, é preciso trocar as conexões com a bateria atual,

para a energia ser encaminhada para a bateria esperada.

Figura 51 – Detalhamento do item 4 do Ciclo do Projeto.

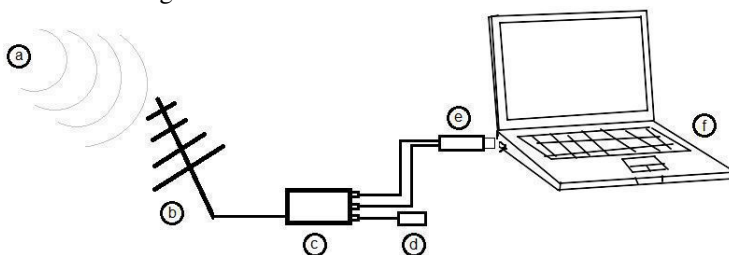


3.7 TRANSMISSÃO DE VÍDEO

Todas as informações, necessárias à ferramenta, são dependentes do vídeo emitido, em tempo real, através da câmera instalada na parte frontal do carrinho. Como meio de envio de dados ao notebook, a transmissão de vídeo ocorre do carrinho ao notebook, através de suas antenas. Utilizando a frequência de 900Mhz. No carrinho, há uma câmera (GoPro Hero 3), que captura os frames, conectada ao transmissor de vídeo e seguidamente a antena envia-os ao ambiente, como ondas eletromagnéticas irradiadas.

As ondas enviadas por esse circuito são capturadas pela antena, conectada ao receptor de vídeo (alimentado por uma bateria Ni-Mh 2200mAh 12 V). Finalmente, das saídas desse equipamento ao notebook, o circuito é completado com um dispositivo de captura de vídeo USB (Figura 52).

Figura 52 – Passos da transmissão de vídeo.

**Legenda:**

- (a) Ondas de 900Mhz, enviadas pelo emissor de vídeo instalado no carrinho;
- (b) Antena de captura;
- (c) Receptor de Vídeo;
- (d) Bateria que fornece energia ao receptor de vídeo;
- (e) Dispositivo de Captura de Vídeo USB;
- (f) Notebook, onde é processado a ferramenta desenvolvida;

4 ESTRATÉGIA DE AVALIAÇÃO

A avaliação deste trabalho se deu em campo, com dados reais e um veículo andando e reagindo/seguindo um caminho com base no resultado da captura de imagem e análise e interpretação da mesma (detecção do caminho). O veículo foi testado em um mini circuito que contém diferentes terrenos, como: asfalto, lajota e terra. O requisito de sucesso define-se da seguinte maneira: O veículo deve se manter dentro do caminho, ou seja, cumprir o trajeto desenhado realizando as curvas e fazendo correções de direção sempre que necessário. Lembrando que o veículo deve percorrer o trajeto definido sem nenhuma intervenção humana. O veículo utilizado pode ser visto na Figura 53. Para esse experimento foi criado primeiramente um *Ground Truth* (GT) com as imagens e detecção do caminho, com o veículo sendo controlado manualmente e posteriormente foi comparado com o veículo navegando de forma autônoma pelo mesmo caminho e em condições de iluminação similares. Vale lembrar que a proposta é avaliada ao ar livre, estando sujeita há diferentes condições de iluminação a cada novo experimento.

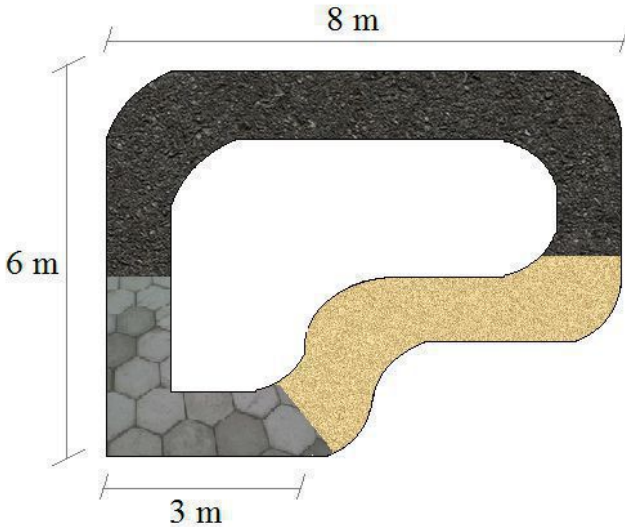
A maioria dos veículos autônomos avalia e valida o trabalho com o teste em campo e com verificação do sucesso no cenário e caminho proposto. Como por exemplo, os vencedores da DARPA *Urban Challenge* de 2007 (URMSON et al., 2008), onde a validação foi vencer a competição, ou seja, cumprir o objetivo proposto sem falhas e no menor tempo possível, não apresentando uma fórmula, métrica ou padrão de validação que possa ser repetido, a não ser o teste em campo do veículo.

Figura 53 – Veículo para experimentos e avaliação do trabalho.



Para a realização dos testes e avaliação dessa proposta de dissertação, foi construído um circuito particular, buscando assim um ambiente mais controlado e permitindo focar somente na detecção de caminho. Além de possibilitar uma quantidade de testes sem dependências de autorizações de uso ou agendamento de horários e movimentos constantes. Entretanto alguns fatores ainda influenciam, como por exemplo a iluminação e o clima no momento, pois o circuito encontra-se em ambiente externo.

Figura 54 – Protótipo do Circuito.



O circuito foi construído visando colocar o veículo em situações necessárias para a avaliação da detecção do caminho em tempo real. No circuito há três tipos de terrenos que provarão a eficácia do veículo em se adaptar e continuar seu trajeto independente da alteração do padrão do caminho. Os tipos são: asfalto, terra e lajotas, e os testes podem ser feitos em sentido horário e anti-horário, possibilitando todas combinações possíveis de alteração no terreno. Além disso, o veículo deverá ser capaz de realizar diferentes curvas e tamanhos de retas

Figura 55 – (a) Asfalto. (b) Terra. (c) Lajota. (d) Intersecção asfalto e terra. (e) Intersecção terra e lajota. (f) Intersecção lajota e asfalto.



5 RESULTADOS

Os resultados obtidos são apresentados considerando cada trecho do circuito e comparando o comportamento do veículo controlado “manualmente” com a imagem gravada em relação ao comportamento do veículo de forma autônoma com base na imagem transmitida em tempo real. Ou seja, na imagem gravada o veículo faz o trajeto de forma “manual” (foi controlado por Arduino, com código estático e valores de direção e aceleração enviados aos servos, previamente definidos) e, neste caso, fez os trechos do caminho independente da segmentação funcionar ou não. E na imagem transmitida, o veículo foi controlado de forma autônoma com base nessas imagens capturadas em tempo real. O vídeo com a navegação manual pode ser visto em: (JUSTEN; RATEKE, 2015) e o vídeo com a navegação autônoma pode ser visto em: (RATEKE; JUSTEN, 2015).

O primeiro trecho do circuito comparado, é a reta de início, onde o terreno é de asfalto. Esse trecho pode ser considerado o mais simples pelo fato do asfalto ter um padrão de cor homogênea (preto) e por ser uma reta longa o que facilita a detecção do caminho. A detecção do caminho obteve sucesso nos dois casos (manual e autônomo) e a navegação autônoma também teve êxito neste trecho (Tabela 4).

No segundo trecho, ocorre o primeiro desafio, neste trecho o veículo deve detectar o caminho em uma curva e no caso da navegação autônoma, realizar sozinho a curva. Neste trecho também houve êxito na detecção e navegação (Tabela 5).

A alteração entre os tipos de terreno (padrões de cor) é provavelmente a maior dificuldade na detecção de caminho na VC e PDI, o veículo deve continuar detectando o caminho e seguindo o trajeto. No caso da navegação manual (imagem gravada), o veículo seguiu em frente independente da detecção do caminho, e neste caso pode-se crer que a detecção foi forçada a ocorrer, já que os pontos de entrada seriam atualizados assim que capturados novos *pixels*. Entretanto, na navegação autônoma o veículo só continua caso a detecção de caminho funcione, e neste caso o veículo detectou o caminho e continuou seu trajeto com sucesso (Tabela 6).

A detecção de caminho deve continuar mesmo em diferentes tipos de terreno, e dentre os padrões de terreno utilizados nos experimentos para avaliação da proposta, a parte de terra é provavelmente a mais difícil de se reconhecer e segmentar, justamente pela falta de homogeneidade e variação na textura do terreno. A detecção continuou ocorrendo, assim como a navegação autônoma (Tabela 7).

No trecho 5, uma reta no terreno de terra, no controle manual (imagem

gravada), o veículo continuou o caminho de forma inclinada, mesmo que as informações de controle mostrando que deveria ir mais para a direita. Isso pois, como dito anteriormente, no controle manual, o veículo estava sendo controlado “manualmente” com um código direto no Arduino. Já na navegação autônoma (imagem transmitida), o veículo se alinhou corretamente e continuou reto no caminho (Tabela 8).

No trecho 6 percebe-se que a segmentação falhou no vídeo transmitido (controle autônomo) o que ocasionou de o veículo perder o caminho e andar reto durante um curto tempo até parar. Quando o veículo fica um certo tempo, uma certa quantidade de *frames*, sem detectar o caminho, ele recebe o comando de parar, e foi o que ocorreu com o veículo. Isso acontece pela variação de luminosidade e principalmente também pelo fato da transmissão do vídeo pelas antenas diminuir consideravelmente a qualidade da imagem. Entretanto, com o vídeo gravado (controle “manual”), é possível perceber com as informações de controle, que ao menos o controle do veículo estaria sendo feito de forma correta. Também nota-se que a lajota é detectada na intersecção com a areia (Tabela 9).

Os demais trechos, intersecção areia e lajota, curva na lajota, reta com lajota e intersecção com o asfalto não puderam ser analisados, pois além da falha na detecção do caminho na imagem transmitida. Devem ser feitos mais testes, tanto com imagens gravadas e controle manual, como com imagens transmitidas e controle autônomo.

Trecho 1 - Reta (Asfalto):

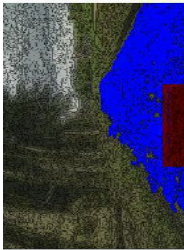


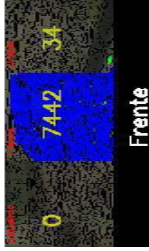
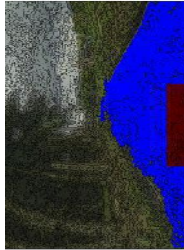



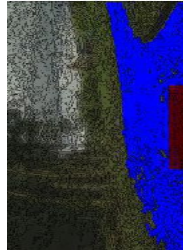

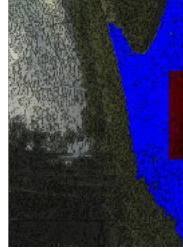

Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			
			
			

Tabela 4 – Resultados - Trecho 1

Trecho 2 - Curva (Asfalto):

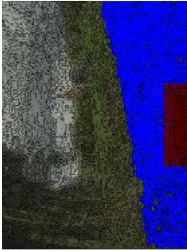

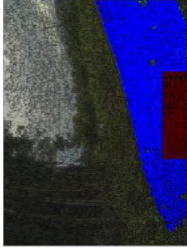
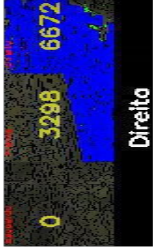
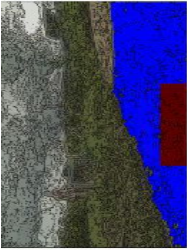
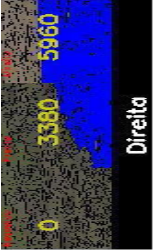
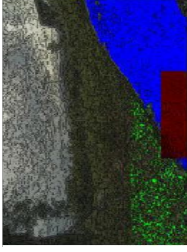
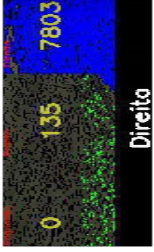
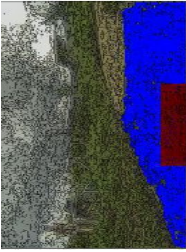

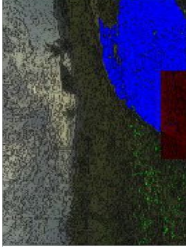
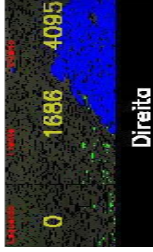
Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			
			
			

Tabela 5 – Resultados - Trecho 2

Trecho 3 - Reta (Interseção asfalto e terra):

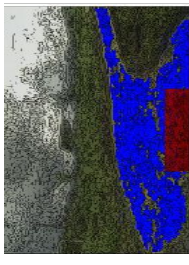
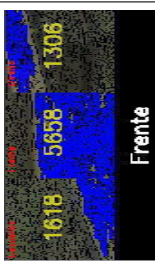
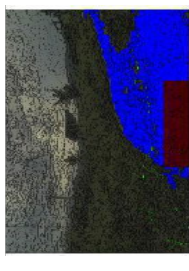
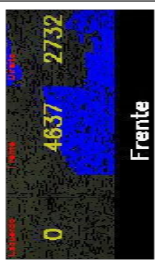
Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			

Tabela 6 – Resultados - Trecho 3

Trecho 4 - Curva (Terra):

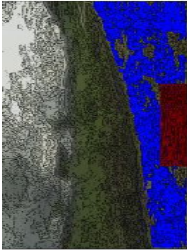
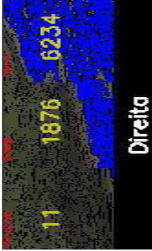
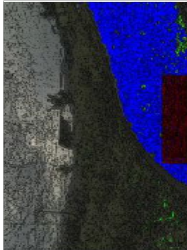

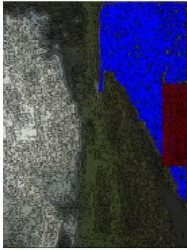
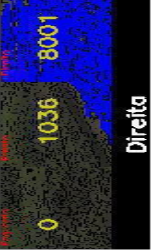
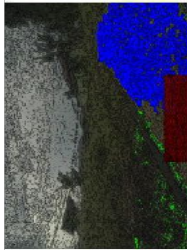
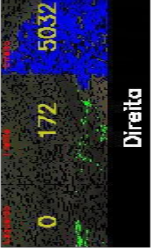
Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			
			

Tabela 7 – Resultados - Trecho 4

Trecho 5 - Retá (Terra):

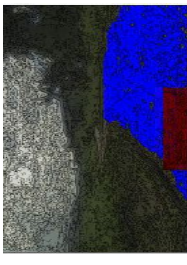
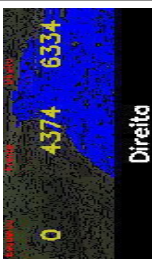


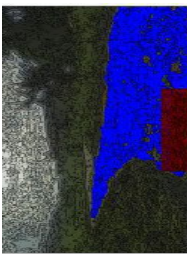
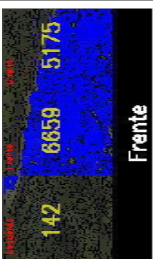
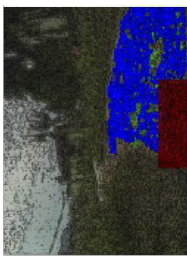
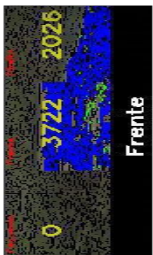
Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			
			

Tabela 8 – Resultados - Trecho 5

Trecho 6 - Curva (Terra):

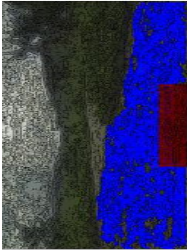
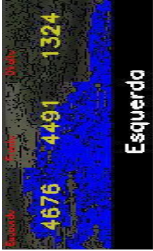
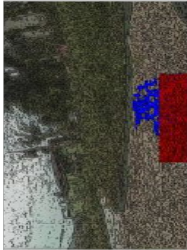
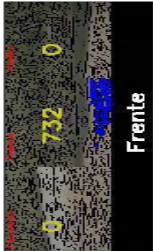
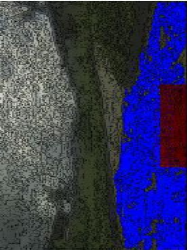

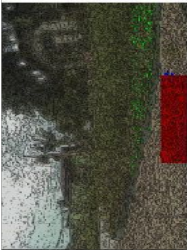
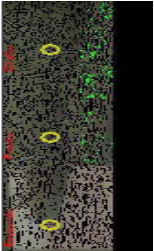
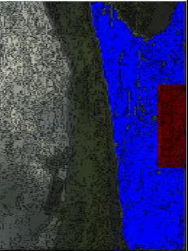

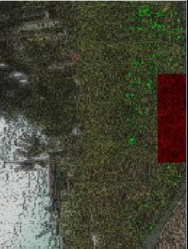
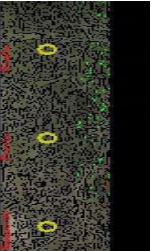
Controle Manual (imagem gravada)	Informações de Controle	Controle Autônomo (imagem transmitida)	Informações de Controle
			
			
			

Tabela 9 – Resultados - Trecho 6

Apesar do veículo não ter completado todos os trechos do trajeto de forma autônoma, a detecção de caminho pode ser verificada no trecho com lajota (Figura 56), que também apresenta os resultados em dia de sol forte, ou seja, mudanças de iluminação e padrões de cor do trajeto, como também pode ser conferido na Figura 57.

Figura 56 – Detecção do caminho no trecho com lajota.

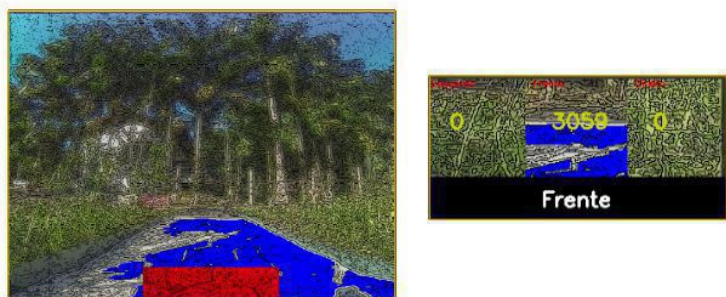
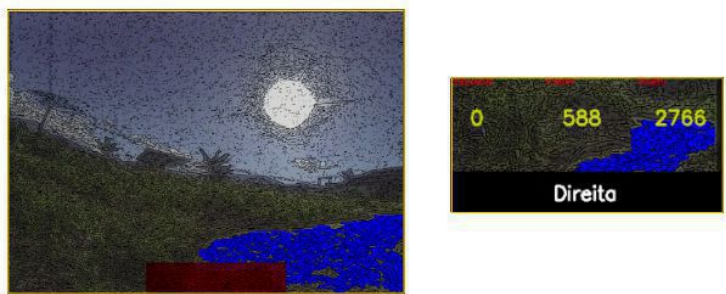


Figura 57 – Detecção do caminho com luz solar forte.



6 CONCLUSÃO

Este trabalho teve como grande motivação, verificar a possibilidade de utilizar a informação de imagem, mais especificamente a visão passiva, para o controle de um veículo autônomo. Verificando assim a possibilidade do uso de técnicas de visão computacional na navegação autônoma, em situações reais e não somente com imagens gravadas. Além disso, imagina-se que em um futuro com veículos autônomos, a tendência pelas pesquisas atuais seria pela utilização de LIDAR (laser) para auxílio na navegação, e entendemos, com base no que foi visto em (STANDARD, 2005) e (GROUP SAFETY PUBLICATION. INTERNATIONAL STANDARD, 2001) que uma grande quantidade desses lasers em público pode representar algum risco à retina quando muito tempo em contato, logicamente são necessários estudos mais aprofundados em relação ao tema e obviamente as pesquisas atuais tem enfoque em possibilidades e ainda tem um longo caminho de adaptações e descobertas até que veículos autônomos se tornem comum ao grande público. O que não impede de verificar o quão possível e realístico é a utilização de visão passiva, como uma alternativa “menos poluente”.

Com base no desenvolvimento da dissertação é possível verificar os resultados da detecção do caminho em três diferentes tipos de terreno, considerados os mais comuns, asfalto, terra e lajota. Foi verificado também a eficácia e adaptação da detecção em áreas de alteração no tipo de terreno. Este trabalho também permite verificar a possibilidade de controlar um veículo autônomo somente com os dados da imagem capturada. Com isso duas afirmações podem ser feitas, que a detecção de caminho se adapta a mudanças de terreno, e que é possível controlar o veículo somente com as informações de imagem.

Entretanto deve-se fazer algumas ponderações. Somente a análise da imagem não oferece segurança suficiente para um veículo autônomo, e este trabalho não buscou afirmar isso, mesmo assim, os estudos e testes aqui feitos mostraram que para um veículo autônomo é necessário maior robustez e redundância de sinais. Então outros sensores podem e devem ser estudados para o auxílio da navegação e segurança do veículo e do ambiente ao redor. Em diversas situações pode ocorrer, como realmente ocorreram, falhas na detecção do caminho, isso pode e vai acabar ocorrendo, mas há possibilidade de ser atenuado, como por exemplo, pode-se estudar formas de se lidar com problema de iluminação, e aplicar outros métodos (como alguns dos estudados na revisão sistemática, por exemplo) para atuarem em conjunto. Mas a verdade é que sendo um ambiente externo, está sujeito a muitas alterações na luminosidade, textura, na intensidade dos *pixels* de uma forma geral, o que

influencia no resultado final. Um agravante ainda maior nesse trabalho, foi o fato da transmissão ter sido realizada por antenas, o que diminuía muito a qualidade da imagem, conseqüentemente limitando a eficácia da detecção.

Termina-se essa etapa do projeto com a certeza de que ainda há muito o que ser feito e estudado. Equipamentos mais adequados e de melhor desempenho devem ser pesquisados, assim como outras utilidades da visão passiva podem ser aplicadas, como por exemplo a detecção de obstáculos.

6.1 TRABALHOS FUTUROS

A detecção de obstáculos é essencial para um veículo autônomo. Uma das tarefas mais importantes em um sistema de navegação autônoma é a percepção do ambiente e dos obstáculos contidos neste ambiente. Então um sistema robusto de veículos inteligentes deve levantar as informações necessárias para a detecção e rastreamento dos obstáculos, como: posição, tamanho e velocidade que cada obstáculo detectado possui (DANESCU et al., 2012).

Muitos métodos de detecção de obstáculos baseados em visão, objetivam detectar apenas certos tipos de obstáculos, como por exemplo: outros carros ou pedestres. Isso acaba ocasionando em sistemas que tem um índice considerável de falsos alarmes de detecção e que perdem obstáculos que deveriam ser detectados (WON; JUNG, 2012). Para a detecção de obstáculos os métodos de Visão Estereoscópica podem ser utilizados. Segundo (ZUREIKI; DEVY; CHATILA, 2007), são técnicas que utilizam as informações capturadas por duas câmeras (fontes de captura) onde, após uma calibração preliminar, estima-se as relações espaciais entre as duas câmeras. Conhecimento este que permite o cálculo das coordenadas 3D de um determinado ponto e sua projeção nas imagens através de uma triangulação simples. Com base nessa projeção, (BLEYER, 2013) diz que é possível calcular a disparidade entre os *pixels*, ou seja, o quanto cada *pixel* é deslocado de uma imagem em relação à outra.

Dentro do cenário atual do trabalho, deve-se ainda considerar uma melhora na confiabilidade da detecção do caminho, e isso pode acontecer de diversas formas. Sendo que com certeza o equipamento utilizado precisa ser melhorado, eliminando por exemplo a queda de qualidade da imagem quando transmitida por antena para o computador que faz todo o processamento. Uma forma de se lidar com isso é que o processamento da imagem seja feito sem necessitar da transmissão do vídeo por antena, com um computador direto no veículo processando os dados e imagens recebidas. Ainda em relação aos equipamentos deve-se melhorar a forma como a bateria do veículo influencia nos resultados de direção e aceleração, o veículo perde muito desempenho

enquanto a bateria gasta, o que dificulta manter um padrão nos valores de controle passados ao veículo.

REFERÊNCIAS

ÁLVAREZ, J. M.; GEVERS, T.; LÓPEZ, A. M. Vision-based road detection using road models. In: **Proceedings of the 16th IEEE international conference on Image processing**. Piscataway, NJ, USA: IEEE Press, 2009. (ICIP'09), p. 2049–2052. ISBN 978-1-4244-5653-6. Disponível em: <<http://dl.acm.org/citation.cfm?id=1819298.1819309>>.

ÁLVAREZ, J. M.; LÓPEZ, A. M. Road detection based on illuminant invariance. **Trans. Intell. Transport. Sys.**, IEEE Press, Piscataway, NJ, USA, v. 12, n. 1, p. 184–193, mar 2011. ISSN 1524-9050. Disponível em: <<http://dx.doi.org/10.1109/TITS.2010.2076349>>.

AMDITIS, A. et al. A situation-adaptive lane-keeping support system: Overview of the safelane approach. **Intelligent Transportation Systems, IEEE Transactions on**, v. 11, n. 3, p. 617–629, 2010. ISSN 1524-9050.

BAI, L.; WANG, Y. Road tracking using particle filters with partition sampling and auxiliary variables. **Computer Vision and Image Understanding**, v. 115, n. 10, p. 1463 – 1471, 2011. ISSN 1077-3142. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314211001421>>.

BAI, L.; WANG, Y.; FAIRHURST, M. An extended hyperbola model for road tracking for video-based personal navigation. **Knowledge-Based Systems**, v. 21, n. 3, p. 265 – 272, 2008. ISSN 0950-7051. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705107001141>>.

BAO, J.; CHEN, Y.; YU, J. An optimized discrete neural network in embedded systems for road recognition. **Engineering Applications of Artificial Intelligence**, v. 25, n. 4, p. 775 – 782, 2012. ISSN 0952-1976. Special Section: Dependable System Modelling and Analysis. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0952197612000206>>.

BLEYER, M. **VU Stereo Vision**. Karlsplatz, Vienna, Austria, 2013. Disponível em: <http://www.ims.tuwien.ac.at/teaching_detail.php?ims_id=188.HQK>. Acesso em: 21 set. 2013.

BROGGI, A.; CATTANI, S. An agent based evolutionary approach to path detection for off-road vehicle guidance. **Pattern Recogn. Lett.**, Elsevier

Science Inc., New York, NY, USA, v. 27, n. 11, p. 1164–1173, aug 2006. ISSN 0167-8655. Disponível em:
<<http://dx.doi.org/10.1016/j.patrec.2005.07.014>>.

CANNY, J. A computational approach to edge detection. **IEEE Trans. Pattern Anal. Mach. Intell.**, IEEE Computer Society, Washington, DC, USA, v. 8, n. 6, p. 679–698, jun 1986. ISSN 0162-8828. Disponível em:
<<http://dx.doi.org/10.1109/TPAMI.1986.4767851>>.

CHEN, C.-L.; TAI, C.-L. Adaptive fuzzy color segmentation with neural network for road detections. **Eng. Appl. Artif. Intell.**, Pergamon Press, Inc., Tarrytown, NY, USA, v. 23, n. 3, p. 400–410, apr 2010. ISSN 0952-1976. Disponível em: <<http://dx.doi.org/10.1016/j.engappai.2009.12.004>>.

DANESCU, R.; NEDEVSCI, S. Probabilistic lane tracking in difficult road scenarios using stereovision. **Intelligent Transportation Systems, IEEE Transactions on**, v. 10, n. 2, p. 272–282, 2009. ISSN 1524-9050.

DANESCU, R. et al. Particle grid tracking system stereovision based obstacle perception in driving environments. **Intelligent Transportation Systems Magazine, IEEE**, v. 4, n. 1, p. 6–20, Spring 2012. ISSN 1939-1390.

DORNAIKA, F. et al. A new framework for stereo sensor pose through road segmentation and registration. **Intelligent Transportation Systems, IEEE Transactions on**, v. 12, n. 4, p. 954–966, 2011. ISSN 1524-9050.

FERNANDES, L. C. et al. Carina intelligent robotic car: Architectural design and applications. **Journal of Systems Architecture**, v. 60, n. 4, p. 372 – 392, 2014. ISSN 1383-7621. Disponível em:
<<http://www.sciencedirect.com/science/article/pii/S1383762113002841>>.

FILHO, O.; NETO, H. **Processamento Digital de Imagens**. São Paulo, Brazil: Ed. Brasport, 1999. 331 p.

GOPALAN, R. et al. A learning approach towards detection and tracking of lane markings. **Intelligent Transportation Systems, IEEE Transactions on**, v. 13, n. 3, p. 1088–1098, 2012. ISSN 1524-9050.

GROUP SAFETY PUBLICATION. INTERNATIONAL STANDARD. **SAFETY OF LASER PRODUCTS. Part 1: Equipment classification, requirements and user's guide**. 2001. Disponível em:
<http://www.microasu.com/download/jqlaser/certificate/IEC_60825_1.pdf>. Acesso em: 18 fev. 2015.

GUO, C.; MITA, S. Drivable road region detection based on homography estimation with road appearance and driving state models. In: **Autonomous Robots and Agents, 2009. ICARA 2009. 4th International Conference on.** [S.l.: s.n.], 2009. p. 204–209.

GUO, C.; MITA, S. Stereovision-based road boundary detection for intelligent vehicles in challenging scenarios. In: **Proceedings of the 2009 IEEE/RSJ international conference on Intelligent robots and systems.** Piscataway, NJ, USA: IEEE Press, 2009. (IROS'09), p. 1723–1728. ISBN 978-1-4244-3803-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=1733023.1733028>>.

GUO, C.; MITA, S.; MCALLESTER, D. Robust road detection and tracking in challenging scenarios based on markov random fields with unsupervised learning. **Intelligent Transportation Systems, IEEE Transactions on,** v. 13, n. 3, p. 1338–1354, 2012. ISSN 1524-9050.

GUO, C.-Z.; YAMABE, T.; MITA, S. Drivable road boundary detection for intelligent vehicles based on stereovision with plane-induced homography. **Acta Automatica Sinica,** v. 39, n. 4, p. 371 – 380, 2013. ISSN 1874-1029. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1874102913600361>>.

ISARD, M.; BLAKE, A. Condensation - conditional density propagation for visual tracking. **International Journal of Computer Vision,** v. 29, n. 1, p. 5–28, 1998. Disponível em: <<http://dx.doi.org/10.1023/A:1008078328650>>.

JOICHEM, T.; POMERLEAU, D.; THORPE, C. Vision-based neural network road and intersection detection and traversal. In: **Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on.** [S.l.: s.n.], 1995. v. 3, p. 344–349 vol.3.

JUSTEN, K. A.; RATEKE, T. **Video Navegação Manual.** [S.l.], 2015. Disponível em: <<http://migre.me/pD64h>>. Acesso em: 05 mar. 2015.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions of the ASME–Journal of Basic Engineering,** v. 82, n. Series D, p. 35–45, 1960.

KIM, B.; SON, J.; SOHN, K. Illumination invariant road detection based on learning method. In: **Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.** [S.l.: s.n.], 2011. p. 1009–1014. ISSN 2153-0009.

KODAGODA, K. R. S.; WIJESOMA, W.; BALASURIYA, A. Cute: curb tracking and estimation. **Control Systems Technology, IEEE Transactions on**, v. 14, n. 5, p. 951–957, 2006. ISSN 1063-6536.

KONG, H.; AUDIBERT, J.-Y.; PONCE, J. General road detection from a single image. **Trans. Img. Proc.**, IEEE Press, Piscataway, NJ, USA, v. 19, n. 8, p. 2211–2220, aug 2010. ISSN 1057-7149. Disponível em: <<http://dx.doi.org/10.1109/TIP.2010.2045715>>.

LEE, S.; LEE, D.; LEE, S. Network-oriented road map generation for unknown roads using visual images and gps-based location information. **Consumer Electronics, IEEE Transactions on**, v. 55, n. 3, p. 1233–1240, 2009. ISSN 0098-3063.

LIDAR. UK, 2015. Disponível em: <<http://www.lidar-uk.com/>>. Acesso em: 18 fev. 2015.

LUNDQUIST, C.; ORGUNER, U.; GUSTAFSSON, F. Extended target tracking using polynomials with applications to road-map estimation. **Signal Processing, IEEE Transactions on**, v. 59, n. 1, p. 15–26, 2011. ISSN 1053-587X.

MARZOTTO, R. et al. A real-time versatile roadway path extraction and tracking on an {FPGA} platform. **Computer Vision and Image Understanding**, v. 114, n. 11, p. 1164 – 1179, 2010. ISSN 1077-3142. <ce:title>Special issue on Embedded Vision</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314210000925>>.

MATSUSHITA, Y.; MIURA, J. On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter. **Robotics and Autonomous Systems**, v. 59, n. 5, p. 274 – 284, 2011. ISSN 0921-8890. <ce:title>Special Issue {ECMR} 2009</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0921889011000297>>.

NEVES, S.; PELAES, E. **Estudo e implementação de técnicas de segmentação de imagens**. Campus Universitário do Guamá, Belém, Pará, 2001. v. 1, n. 2, 1-11 p. Disponível em: <<http://www.lidar-uk.com/>>. Acesso em: 18 fev. 2015.

OBRADOVIC, D. et al. Linear fuzzy space based road lane model and detection. **Knowledge-Based Systems**, v. 38, n. 0, p. 37 – 47, 2013. ISSN 0950-7051. <ce:title>Special Issue on "Advances in Fuzzy Knowledge Systems: Theory and Application</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950705112000032>>.

ONIGA, F.; NEDEVSKI, S. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. **Vehicular Technology, IEEE Transactions on**, v. 59, n. 3, p. 1172–1182, 2010. ISSN 0018-9545.

OSOSINSKI, M.; LABROSSE, F. Automatic driving on ill-defined roads: An adaptive, shape-constrained, color-based method. **Journal of Field Robotics**, p. n/a–n/a, 2013. ISSN 1556-4967. Disponível em: <<http://dx.doi.org/10.1002/rob.21494>>.

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

P.K. Kalra. **Canny Edge Detection**. Delhi, India, 2009. Disponível em: <<http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf>>. Acesso em: 20 fev. 2015.

POMERLEAU, D. A. **Neural Network Perception for Mobile Robot Guidance**. Norwell, MA, USA: Kluwer Academic Publishers, 1993. ISBN 0792393732.

RATEKE, T.; JUSTEN, K. A. **Video Navegação Autônoma**. [S.l.], 2015. Disponível em: <<http://migre.me/pD67v>>. Acesso em: 05 mar. 2015.

ROSENBLUM, M. Neurons that know how to drive. In: **Proceedings of the IEEE Intelligent Vehicles Symposium, Dearborn, MI**. [S.l.: s.n.], 2000. p. 556–562.

SALES, D. O. et al. Adaptive finite state machine based visual autonomous navigation system. **Engineering Applications of Artificial Intelligence**, v. 29, n. 0, p. 152 – 162, 2014. ISSN 0952-1976. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0952197613002406>>.

SCHWORER, I. **Navigation and Control of an Autonomous Vehicle**. Dissertação (Mestrado) — Faculty of the Virginia Polytechnic Institute and State University, Blacksburg, Virginia, EUA, 2005.

SOBIERANSKI, A. C. **Segmentação supervisionada de imagens pela funcional de mumford-shah utilizando métricas de distância não-lineares**. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, Santa Catarina, Brasil, 2010.

SON, T.; MITA, S.; TAKEUCHI, A. Road detection using segmentation by weighted aggregation based on visual information and a posteriori

probability of road regions. In: **Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on**. [S.l.: s.n.], 2008. p. 3018–3025. ISSN 1062-922X.

STANDARD, A. N. **American National Standard for Safe use of Lasers Outdoors**. Orlando, FL, 2005. Disponível em: <https://www.lia.org/PDF/Z136_6_s.pdf>. Acesso em: 18 fev. 2015.

URMSON, C. **Just press go: designing a self-driving vehicle**. May 2014. Disponível em: <<http://googleblog.blogspot.com.br/2014/05/just-press-go-designing-self-driving.html>>. Acesso em: 26 nov. 2014.

URMSON, C. **The latest chapter for the self-driving car: mastering city street driving**. April 2014. Disponível em: <<http://googleblog.blogspot.com.br/2014/04/the-latest-chapter-for-self-driving-car.html>>. Acesso em: 26 nov. 2014.

URMSON, C. et al. Autonomous driving in urban environments: Boss and the urban challenge. **Journal of Field Robotics Special Issue on the 2007 DARPA Urban Challenge, Part I**, Wiley InterScience, v. 25, n. 8, p. 425–466, June 2008.

WANG, Y.; BAI, L.; FAIRHURST, M. Robust road modeling and tracking using condensation. **Intelligent Transportation Systems, IEEE Transactions on**, v. 9, n. 4, p. 570–579, 2008. ISSN 1524-9050.

WANG, Y.; DAHNOUN, N.; ACHIM, A. A novel system for robust lane detection and tracking. **Signal Processing**, v. 92, n. 2, p. 319 – 334, 2012. ISSN 0165-1684. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0165168411002532>>.

WANGENHEIM, A. v. et al. Color image segmentation guided by a color gradient network. **Pattern Recognition Letters**, v. 28, n. 13, p. 1795 – 1803, 2007. ISSN 0167-8655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167865507001560>>.

WEDEL, A. et al. B-spline modeling of road surfaces with an application to free-space estimation. **Intelligent Transportation Systems, IEEE Transactions on**, v. 10, n. 4, p. 572–583, 2009. ISSN 1524-9050.

WON, K.; JUNG, S. Billboard sweep stereo for obstacle detection in road scenes. **Electronics Letters**, v. 48, n. 24, p. 1528–1530, 2012. ISSN 0013-5194.

WU, B.-F.; LIN, C.-T.; CHEN, Y.-L. Dynamic calibration and occlusion handling algorithms for lane tracking. **Industrial Electronics, IEEE Transactions on**, v. 56, n. 5, p. 1757–1773, 2009. ISSN 0278-0046.

ZHAO, Y.; WANG, H.; YAN, R. Unstructured road edge detection and initial positioning approach based on monocular vision. **{AASRI} Procedia**, v. 1, n. 0, p. 486 – 491, 2012. ISSN 2212-6716. <ce:title>AASRI Conference on Computational Intelligence and Bioinformatics</ce:title>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2212671612000777>>.

ZHOU, S. et al. Road detection using support vector machine based on online learning and evaluation. In: **Intelligent Vehicles Symposium (IV), 2010 IEEE**. [S.l.: s.n.], 2010. p. 256–261. ISSN 1931-0587.

ZUREIKI, A.; DEVY, M.; CHATILA, R. Stereo matching using reduced-graph cuts. In: **Image Processing, 2007. ICIP 2007. IEEE International Conference on**. [S.l.: s.n.], 2007. v. 1, p. I – 237–I – 240. ISSN 1522-4880.