

---

Navegação de veículos autônomos em ambientes  
externos não estruturados baseada em visão  
computacional

*Rafael Luiz Klaser*

---



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: \_\_\_\_\_

# Navegação de veículos autônomos em ambientes externos não estruturados baseada em visão computacional

**Rafael Luiz Klaser**

***Orientador: Prof. Dr. Fernando Santos Osório***

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

**USP – São Carlos**  
**Agosto de 2014**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi  
e Seção Técnica de Informática, ICMC/USP,  
com os dados fornecidos pelo(a) autor(a)

K63n Klaser, Rafael Luiz  
Navegação de veículos autônomos em ambientes  
externos não estruturados baseada em visão  
computacional / Rafael Luiz Klaser; orientador  
Fernando Santos Osório. -- São Carlos, 2014.  
81 p.

Dissertação (Mestrado - Programa de Pós-Graduação  
em Ciências de Computação e Matemática  
Computacional) -- Instituto de Ciências Matemáticas  
e de Computação, Universidade de São Paulo, 2014.

1. visão computacional. 2. navegação autônoma. 3.  
mapa de navegabilidade. 4. simulação. 5. ambiente  
externo. I. Osório, Fernando Santos, orient. II.  
Título.



“Pain is temporary. It may last a minute, or an hour, or a day, or a year, but eventually it will subside and something else will take its place. If I quit, however, it lasts forever. That surrender, even the smallest act of giving up, stays with me. So when I feel like quitting, I ask myself, which would I rather live with?” – *Lance Armstrong*



# Agradecimentos

---

---

Agradeço meu orientador prof. Fernando Santos Osório pela oportunidade de trabalhar com um tema desafiador e atual, pela colaboração e apoio durante o projeto. Gostaria de agradecer todos integrantes do Laboratório de Robótica Móvel, que se tornaram amigos e estiveram sempre presentes como um grande grupo.

Este trabalho contou com o suporte financeiro cedido pela **FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo**, processo # **2012/04555-4**.





# Resumo

---

---

Este trabalho apresenta um sistema de navegação autônoma para veículos terrestres com foco em ambientes não estruturados, tendo como principal meta aplicações em campos abertos com vegetação esparsa e em cenário agrícola. É aplicada visão computacional como sistema de percepção principal utilizando uma câmera estéreo em um veículo com modelo cinemático de *Ackermann*. A navegação é executada de forma deliberativa por um planejador baseado em malha de estados sobre um mapa de custos e localização por odometria e GPS. O mapa de custos é obtido através de um modelo de ocupação probabilístico desenvolvido fazendo uso de uma *OctoMap*. É descrito um modelo sensorial para atualizar esta *OctoMap* a partir da informação espacial proveniente de nuvens de pontos obtidas a partir do método de visão estéreo. Os pontos são segmentados e filtrados levando em consideração os ruídos inerentes da aquisição de imagens e do processo de cálculo de disparidade para obter a distância dos pontos. Os testes foram executados em ambiente de simulação, permitindo a replicação e repetição dos experimentos. A modelagem do veículo foi descrita para o simulador físico Gazebo de acordo com a plataforma real CaRINA I (veículo elétrico automatizado do LRM-ICMC/USP), levando-se em consideração o modelo cinemático e as limitações deste veículo. O desenvolvimento foi baseado no ROS (*Robot Operating System*) sendo utilizada a arquitetura básica de navegação deste *framework* a partir da customização dos seus componentes. Foi executada a validação do sistema no ambiente real em cenários com terreno irregular e obstáculos diversos. O sistema apresentou um desempenho satisfatório tendo em vista a utilização de uma abordagem baseada em apenas uma câmera estéreo. Nesta dissertação são apresentados os principais componentes de um sistema de navegação autônoma e as etapas necessárias para a sua concepção, assim como resultados de experimentos simulados e com o uso de um veículo autônomo real.

**Palavras-chave:** Visão computacional, navegação autônoma, mapa de navegabilidade, simulação, ambiente externo.



# Abstract

---

---

**T**his work presents a system for autonomous vehicle navigation focusing on unstructured environments, with the primary goal applications in open fields with sparse vegetation, unstructured environments and agricultural scenario. Computer vision is applied as the main perception system using a stereo camera in a car-like vehicle with Ackermann kinematic model. Navigation is performed deliberately using a path planner based on a lattice state space over a cost map with localization by odometry and GPS. The cost map is obtained through a probabilistic occupation model developed making use of an OctoMap. It is described a sensor model to update the spatial occupancy information of the OctoMap from a point cloud obtained by stereo vision. The points are segmented and filtered taking into account the noise inherent in the image acquisition and calculation of disparity to obtain the distance from points. Tests are performed in simulation, allowing replication and repetition of experiments. The modeling of the vehicle is described to be used in the Gazebo physics simulator in accordance with the real platform CaRINA I (LRM-ICMC/USP automated electrical vehicle) taking into account the kinematic model and the limitations of this vehicle. The development is based on ROS (Robot Operating System) and its basic navigation architecture is customized. System validation is performed on real environment in scenarios with different obstacles and uneven terrain. The system shows satisfactory performance considering a simple configuration and an approach based on only one stereo camera. This dissertation presents the main components of an autonomous navigation system and the necessary steps for its conception as well as results of experiments in simulated and using a real autonomous vehicle.

**Keywords:** Computer vision, autonomous navigation, navigability map, simulation, off-road environment.



# Sumário

---

---

<b>Agradecimentos</b>	<b>iii</b>
<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>Lista de Abreviações</b>	<b>xix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Justificativa . . . . .	3
1.2 Motivação . . . . .	3
1.3 Objetivos . . . . .	4
1.3.1 Objetivo Geral . . . . .	4
1.3.2 Objetivos Específicos . . . . .	4
1.4 Organização do texto . . . . .	4
<b>2 Fundamentação Teórica</b>	<b>7</b>
2.1 Informação Sensorial: Visão Computacional . . . . .	8
2.2 Mapas de Disparidade . . . . .	11
2.3 Espaço Tridimensional . . . . .	12
2.3.1 OctoMap . . . . .	14
2.4 Localização . . . . .	16
2.5 Mapeamento . . . . .	18
2.5.1 Mapa de Navegabilidade . . . . .	18
2.6 Navegação . . . . .	19
2.7 Considerações Finais . . . . .	23

<b>3</b>	<b>Trabalhos Relacionados</b>	<b>25</b>
3.1	LAGR . . . . .	25
3.2	RoBombeiros . . . . .	27
3.3	Projetos do Laboratório de Robótica Móvel . . . . .	28
3.4	Autonomous Prime Movers . . . . .	29
3.5	Considerações Finais . . . . .	30
<b>4</b>	<b>Metodologia</b>	<b>31</b>
4.1	Plataforma de Hardware . . . . .	31
4.2	ROS . . . . .	33
4.3	Simulação Física . . . . .	34
4.4	Percepção Visual . . . . .	36
4.5	Arquitetura do Sistema de Navegação . . . . .	38
4.6	Considerações Finais . . . . .	39
<b>5</b>	<b>Desenvolvimento</b>	<b>41</b>
5.1	Odometria e Localização . . . . .	41
5.2	Cálculo da Disparidade . . . . .	43
5.3	Tratamento da Nuvem de Pontos . . . . .	44
5.3.1	Detecção do chão . . . . .	44
5.3.2	Ruído . . . . .	45
5.3.3	Filtro . . . . .	45
5.3.4	Segmentação da cena . . . . .	46
5.4	Modelo Probabilístico de Ocupação . . . . .	47
5.4.1	Modelo Sensorial . . . . .	48
5.5	Sistema de Navegação . . . . .	50
5.5.1	Mapa de Navegabilidade . . . . .	50
5.5.2	Mapa de Custos . . . . .	50
5.5.3	Planejador Global . . . . .	51
5.5.4	Planejador Local . . . . .	52
5.6	Considerações Finais . . . . .	54
<b>6</b>	<b>Resultados</b>	<b>55</b>
6.1	Testes em Simulação . . . . .	58
6.1.1	Primeiro cenário de teste . . . . .	59
6.1.2	Segundo cenário de teste . . . . .	65
6.2	Validação em Ambiente Real . . . . .	67
6.2.1	Manobra no primeiro ambiente . . . . .	69
6.2.2	Primeira manobra no segundo ambiente . . . . .	70
6.2.3	Segunda manobra no segundo ambiente . . . . .	71
6.3	Vídeos . . . . .	72
<b>7</b>	<b>Conclusão</b>	<b>73</b>
7.1	Principais Contribuições . . . . .	74
7.2	Trabalhos Futuros . . . . .	75







---

# Lista de Figuras

---

2.1	Distorções óticas: sem distorção ( <i>esquerda</i> ), distorção tipo “barril” ( <i>centro</i> ), distorção tipo “almofada” ( <i>direita</i> ) . . . . .	9
2.2	Exemplo de composição ótica de lentes . . . . .	9
2.3	Geometria epipolar (Faugeras, 1993) . . . . .	10
2.4	Retificação (Fusiello <i>et al.</i> , 2000) . . . . .	10
2.5	Sistema de visão tridimensional embarcado baseado em mapa de disparidade (Khaleghi <i>et al.</i> , 2008) . . . . .	12
2.6	Estéreo: (a) imagem RGB (b) disparidade referencial <i>University of Tsukuba</i> . . . . .	12
2.7	Nuvem de pontos a partir de câmera estéreo - <i>Willow Garage</i> . . . . .	13
2.8	Mapa de ocupação ( <i>OctoMap</i> ) (Hornung <i>et al.</i> , 2013) . . . . .	13
2.9	Segmentação por <i>stixels</i> (Badino <i>et al.</i> , 2009) . . . . .	13
2.10	<i>Octree</i> : a) Espaço subdividido por octantes, b) Representação em forma de árvore . . . . .	15
2.11	<i>Ray-casting</i> : são obtidos os pontos que representam um raio da origem até um destino . . . . .	16
2.12	Modelo sensorial para a marcação da ocupação do espaço . . . . .	16
2.13	Arquitetura de mapeamento: os dados volumétricos provenientes dos sensores produzem um mapa local de custos, que atualiza um mapa global de custos composto por informações mapeadas a priori e provenientes de um veículo aéreo (Kelly <i>et al.</i> , 2006) . . . . .	19
2.14	Mapa de ocupação planar: Em branco o espaço livre, em preto o espaço ocupado e em cinza o espaço sem informação de ocupação - <i>ROS</i> . . . . .	20
2.15	Mapa topológico do metrô da cidade de São Paulo - <i>Governo do Estado de São Paulo</i> . . . . .	20
2.16	VFH - (a) Gráfico do histograma, onde $H'(k)$ representam as forças de repulsão aos obstáculos nas direções $k$ . (b) Representação espacial em relação aos obstáculos (retângulos em preto) - A área circulada representa a posição do robô. (Borenstein & Koren, 1991) . . . . .	21
2.17	Planejamento de trajetória por amostras - <i>SBPL</i> . . . . .	22
2.18	Malha de expansão de amostras de trajetórias em um terreno irregular (Howard & Kelly, 2007) . . . . .	22

2.19	Árvore de expansão de menor custo (T-RRT) em um mapa de elevação convertido em custos. Em verde o caminho de menor custo entre os pontos em amarelo. Em azul a expansão . . . . .	22
3.1	Veículos participantes de edições do ELROB - <a href="http://www.elrob.org">www.elrob.org</a> (divulgação) . . .	26
3.2	Veículo do projeto LAGR - <i>NIST</i> . . . . .	26
3.3	Mapa de navegabilidade, em amarelo o caminho livre, em preto o espaço livre, em tons de cinza os obstáculos (Konolige <i>et al.</i> , 2009) . . . . .	27
3.4	RoBombeiros - Navegação em ambiente não estruturado e estratégia de combate a incêndio florestal (Pessin, 2008) . . . . .	28
3.5	Topologia da RNA no simulador RoBombeiros . . . . .	28
3.6	Diagrama de blocos do sistema de navegação proposto em Mendes & Wolf (2011)	29
3.7	Veículo autônomo para o auxílio de colheita em pomares (Bergerman <i>et al.</i> , 2012)	30
4.1	CaRINA I - Carro elétrico automatizado . . . . .	32
4.2	IMU com GPS integrado Xsens MTi-G - <i>Xsens</i> . . . . .	32
4.3	Câmera estéreo Point Grey Bumblebee 2 - <i>Pointgrey</i> . . . . .	32
4.4	Sistemas de referência de coordenadas . . . . .	34
4.5	Representação em árvore, a esfera amarela representa a raiz e as setas em magenta a relação filho-pai . . . . .	34
4.6	CaRINA I - modelo virtual . . . . .	35
4.7	CaRINA I - cenário teste de ambiente não estruturado . . . . .	35
4.8	Geometria Ackermann de esterçamento modelada em simulação . . . . .	36
4.9	Efeitos da modelagem da suspensão no ambiente simulado . . . . .	36
4.10	Sistema de visão: <b>a</b> ) imagem (apenas a esquerda), <b>b</b> ) mapa de disparidade calculado, <b>c</b> ) nuvem de pontos, <b>d</b> ) Marcação da ocupação espacial em um OctoMap	37
4.11	Arquitetura geral do sistema de navegação - (adaptado de: <a href="http://www.ros.org">www.ros.org</a> ) . . . . .	39
5.1	Odometria e localização por GPS . . . . .	42
5.2	Odometria e localização por <i>dead-reckoning</i> . . . . .	42
5.3	Imagem obtida pela câmera estéreo (somente esquerda) . . . . .	43
5.4	Cálculo da disparidade entre as imagens (esquerda / direita). . . . .	43
5.5	Nuvem de pontos . . . . .	44
5.6	Em preto, os pontos segmentados como chão . . . . .	44
5.7	Ruído ondular (em azul) na nuvem de pontos gerada pela câmera estéreo . . . . .	45
5.8	Estrutura real (árvore) e a projeção da nuvem de pontos . . . . .	45
5.9	Em azul os pontos ajustados pelo filtro, em vermelho os pontos fora do intervalo	46
5.10	Projeção dos pontos na imagem real . . . . .	46
5.11	Agrupamentos marcados com diferentes cores (algumas cores se repetem) . . . . .	46
5.12	Foto do cenário real . . . . .	46
5.13	Ocupação espacial probabilística em tons de cinza. Tons escuros representam maior probabilidade, tons claros menores probabilidades . . . . .	47
5.14	Distância <i>versus</i> disparidade - Adaptado de (Bradski & Kaehler, 2008) . . . . .	49
5.15	Probabilidade diferenciada para cada intervalo de distância para contemplar a imprecisão do cálculo da distância pela disparidade . . . . .	49

5.16	Mapa de navegabilidade (2D) . . . . .	50
5.17	Mapa de custos em tons de cinza, sendo o branco o maior custo, o preto o menor	50
5.18	Primitivas de trajetórias básicas levando em consideração o modelo cinemático do veículo . . . . .	52
5.19	Rotação das primitivas para permitir a expansão em 360 graus . . . . .	52
5.20	O traçado em vermelho representa a trajetória planejada até o destino escolhido desviando dos obstáculos marcados em rosa . . . . .	53
5.21	Janela de navegação local. O traçado em vermelho representa a trajetória planejada, em azul, o traçado do controle referente ao esterçamento e velocidade naquele instante . . . . .	53
6.1	Campo de visão do sistema de percepção . . . . .	56
6.2	Ocupação espacial probabilística representada pela OctoMap . . . . .	58
6.3	Visualização dos obstáculos e o comportamento do sistema de percepção e localização, em a) pode ser vista a representação pela OctoMap da ocupação espacial e o mapa de navegabilidade, em b), o mapa de custos. Em verde, são representadas as inflações dos custo dos obstáculos detectados . . . . .	59
6.4	Imagens do cenário em simulação para o experimento de teste . . . . .	60
6.5	Exemplos de cenários de pomares . . . . .	65
6.6	Imagens do cenário em simulação para o segundo experimento de teste . . . . .	65
6.7	Visualização do comportamento da navegação no segundo experimento simulado. O início do percurso está marcado pelo círculo e o final pelo quadrado, em azul . . . . .	66
6.8	Primeiro ambiente real de validação . . . . .	67
6.9	Imagem de satélite do primeiro ambiente de validação - <i>Google Maps</i> . . . . .	67
6.10	Segundo ambiente real de validação . . . . .	68
6.11	Imagem de satélite do segundo ambiente de validação - <i>Google Maps</i> . . . . .	68
6.12	Detecção do plano do chão (em preto) e marcação dos obstáculos (em cores). Visualização (a) e projeção da visualização na cena (b) . . . . .	68
6.13	Visualização da manobra autônoma no primeiro ambiente de validação . . . . .	69
6.14	Visualização da primeira manobra autônoma no segundo ambiente de validação	70
6.15	Visualização da segunda manobra autônoma no segundo ambiente de validação	71



# Lista de Tabelas

---

---

5.1	Probabilidades de cada intervalo de distância no modelo sensorial . . . . .	49
6.1	Parâmetros do veículo . . . . .	57
6.2	Principais sistemas de referência de coordenadas . . . . .	57
6.3	Dados do experimento em simulação . . . . .	61
6.4	Visualização do experimento em simulação . . . . .	62
6.5	Videos disponibilizados na Internet . . . . .	72



# Lista de Abreviações

---

---

**AD\*** Anytime Dynamic A-Star

**CaRINA** Carro Robótico Inteligente para Navegação Autônoma

**CCD** Charge-coupled Device

**DEM** Digital Elevation Map

**DGPS** Differential Global Positioning System

**DRC** DARPA Robotics Challenge

**DTED** Digital Terrain Elevation Description

**DWA** Dynamic Window Approach

**FPS** Frames Per Second

**GPS** Global Positioning System

**INCT-SEC** Instituto Nacional de Ciência e Tecnologia - Sistemas Embarcados Críticos

**IMU** Inertial Measurement Unit

**ODE** Open Dynamics Engine

**PR2** Personal Robot 2

**RANSAC** Random Sampling Consensus

**RGB-D** Red, Gree, Blue, Disparity

**RMA** Robô Móvel Autônomo

**RNA** Rede Neural Artificial

**ROS** Robot Operating System

**SAD** Sum of Absolute Differences

**SDF** Simulation Description Format

**SLAM** Simultaneous Location And Mapping

**T-RRT** Transitional Rapidly exploring Random Tree

**TF** Transform Frames

**UAV** Unmanned Aerial Vehicle

**UGV** Unmanned Ground Vehicle

**URDF** Unified Robot Description Format

**VFH** Vector Field Histogram

**VO** Visual Odometry



---

# Introdução

---

Na robótica móvel, a navegação é a capacidade do veículo de se locomover no ambiente, podendo ser guiada (teleoperada), autônoma ou híbrida (semiautônoma). A navegação autônoma é totalmente dependente do aparato sensorial acessível ao robô, pois a percepção e estímulos do ambiente precisam ser capturados para que este possa reagir de acordo.

As características físicas do robô também vão limitar seu comportamento em relação a sua mobilidade (restrições cinemáticas), onde o aparato sensorial e motor também devem ser propícios para a execução de suas tarefas. A capacidade de se deslocar de forma autônoma e segura depende certamente de um determinado grau de “inteligência” do dispositivo de controle e navegação do robô. Através do uso de técnicas de Inteligência Artificial se busca traduzir em algoritmos estes processos de controle e navegação inteligente. Estes algoritmos podem ser baseados nos princípios clássicos de otimização e de busca assim como em técnicas de aprendizado de máquina.

A capacidade de percepção e locomoção é própria dos animais. Em robôs móveis, para mimetizar essa capacidade, dotam-se os mesmos de aparatos motores e sensoriais, podendo estes, ser ou não inspirados na fisiologia dos animais (Webb, 2000). Tais aparatos são fundamentais e definem a capacidade, características e especificidades dos movimentos. A visão é um subsistema biofísico sensorial o qual podemos associar intuitivamente às especificidades e características das estratégias de locomoção nos animais. Portanto, implementar um sistema de visão artificial com capacidade sensorial em um robô móvel, significa dotá-lo de um sistema

análogo de percepção, que irá permitir que este realize tarefas de movimentação e deslocamento como nos animais.

Para Marr (1982), a visão é acima de tudo uma tarefa de processamento de informações, porém, como a informação é representada a partir de imagens, o tratamento destas é um dos principais problemas que precisa ser investigado. Desta forma, a constituição de um sistema de visão computacional adequado para um determinado problema, como a navegação robótica, ainda é um tema aberto em pesquisa.

A visão provê um conjunto vasto de informações, como cores, texturas, formas e movimentos e exige uma grande capacidade de processamento; estima-se que um terço do cérebro humano seja dedicado à visão. Até que a imagem seja formada no receptor sensorial, a câmera de vídeo no caso da visão artificial, ocorrem transformações que provocam a perda da informação geométrica da cena. Uma transformação é a projeção perspectiva, que reduz em uma representação bidimensional a cena tridimensional.

Recuperar essa informação tridimensional e as características geométricas da cena é a principal função da visão estéreo. Em visão computacional, a informação tridimensional resultante dos métodos computacionais tem elevado ruído associado. Estes ruídos podem estar associados tanto a erros numéricos como por aproximações, estimação de valores e simplificações do modelo matemático que descreve a formação da imagem. Este trabalho, portanto, utiliza uma abordagem probabilística para tratar a informação visual e contemplar o ruído através de incertezas em um mapa de navegabilidade.

A navegação autônoma é talvez um dos principais aspectos a ser alcançado por um robô móvel. Esta característica confere ao veículo a capacidade de executar tarefas de locomoção de forma autônoma. No contexto de veículos autônomos voltados a ambientes não estruturados ou semi estruturados, a aplicação por exemplo na área agrícola, tem grande apelo para o avanço da mecanização de tarefas como plantio, colheita, pulverização e demais atividades afim.

Aplicações mais abrangentes e concretas dos Robôs Móveis Autônomos (RMA) ainda dependem de avanços nas pesquisas científicas e tecnológicas. Por outro lado, já é possível visualizar a sua viabilidade através da combinação de abordagens bem sucedidas em diversas áreas para diversos tipos de aplicações junto a sociedade, tanto do ponto de vista econômico quanto do ponto de vista das tecnologias que já estão sendo embarcada nestes dispositivos. Exemplos disto são os veículos exploradores espaciais, equipamentos de agricultura de precisão, como colheitadeiras e até mesmo em projetos recentes com veículos urbanos.

## 1.1 Justificativa

Processos como segmentação, detecção de bordas, e extração de características locais já são amplamente utilizados em aplicações baseadas em imagem, mas no caso da navegação robótica em ambientes não estruturados é necessário o uso de informações de percepção espacial (3D), a fim de identificar obstáculos e outros elementos que possam prejudicar a navegação do robô. Algoritmos de tratamento de informação tridimensional tem, em geral, um custo computacional elevado. Com a capacidade de processamento dos computadores atuais, algoritmos mais complexos e custosos se tornam viáveis, permitindo a concepção de um sistema de visão computacional mais eficaz.

Com o uso de câmeras estéreo associadas a técnicas para a geração dos mapas de disparidade é possível obter dados semelhantes aos sensores do tipo *rangefinder*, baseados em sonar, laser ou infravermelho, porém com uma resolução espacial superior na ordem de *megapixels*. Além disto, as câmeras são dispositivos de baixo custo se comparadas, por exemplo, com dispositivos como os sensores a laser de medida de distância. Até mesmo as câmeras estéreo possuem atualmente um custo relativamente baixo em relação a estes tipos de sensores pois a princípio são constituídas a partir de duas câmeras comuns.

## 1.2 Motivação

Este trabalho tem por motivação o desenvolvimento de um sistema de navegação autônoma baseado em visão computacional. Apesar de existirem sensores mais sofisticados e com maior precisão de captura de informações tridimensionais do ambiente, a aplicação de visão estéreo baseada em câmeras de vídeo torna o sistema mais simples em termos de equipamentos utilizados. Com foco em ambientes externos e não estruturados, este tipo de sistema robótico tem vasta aplicabilidade. O interesse por essas aplicações na área da robótica móvel tem sido demonstrado tanto pelo surgimento de grupos de pesquisa como pelas conferências e publicações na área.

Um exemplo de aplicação deste tipo de sistema pode ser visto no trabalho de Pessin (2008) onde foi avaliada a utilização de veículos autônomos no combate a incêndios florestais. Este tipo de aplicação requer que um (ou mais) veículo se desloque de um ponto de origem (base) até um destino determinado (foco de incêndio) utilizando coordenadas de GPS desviando de obstáculos e se deslocando em um terreno irregular e não estruturado.

Aplicações como esta de combate a incêndios florestais em ambientes externos não estruturados, entre outras, como por exemplo, monitoramento ambiental, transporte militar de cargas, aplicações agrícolas, são exemplos das possíveis aplicações do trabalho aqui proposto.

## 1.3 Objetivos

### 1.3.1 Objetivo Geral

Desenvolver um método de navegação autônoma, reativo/deliberativo, focado em ambientes externos não estruturados, com base em um mapa de navegabilidade com informação espacial (tridimensional) construído a partir de visão estéreo.

### 1.3.2 Objetivos Específicos

Os objetivos específicos deste projeto de mestrado, que se apresentam como um desdobramento do objetivo geral descrito acima, são:

- Extrair referenciais a partir de um par de câmeras, constituindo um sistema de visão binocular (estéreo);
- Gerar um mapa de navegabilidade a partir de informações visuais;
- Desenvolver um mecanismo de navegação, baseado nas informações de GPS e do sistema de visão, capaz de desviar de obstáculos e dirigir o veículo até um destino determinado de forma autônoma;
- Aplicar o sistema de navegação autônoma em um veículo real em ambiente externo não estruturado, com o terreno levemente irregular e vegetação esparsa.

## 1.4 Organização do texto

O texto está organizado conforme os capítulos abaixo:

- No *Capítulo 2* são apresentados os principais conceitos referente aos temas abordados.
- No *Capítulo 3* são apresentados alguns trabalhos envolvendo navegação autônoma em ambientes não estruturados, assim como aplicações de veículos autônomos nestes cenários.

- 
- No *Capítulo 4* é descrita a metodologia e as ferramentas utilizadas no desenvolvimento do projeto.
  - No *Capítulo 5* é apresentado o desenvolvimento do projeto e as etapas para a construção do sistema de navegação autônoma.
  - No *Capítulo 6* são apresentados os resultados obtidos, os testes e as validações executadas. Também, são listados conteúdos externos para a visualização de todo o desenvolvimento do projeto.
  - No *Capítulo 7* é feita a conclusão desta dissertação e apresentadas continuidades possíveis de serem abordadas em relação ao tema tratado.



---

## Fundamentação Teórica

---

Um veículo autônomo tem como problemas básicos as questões referentes ao mapeamento, localização e navegação, as quais são fortemente influenciadas pelas características do ambiente, do tipo de aplicação e da configuração do robô (sensores e atuadores) (Wolf *et al.*, 2009).

Quando se trata de navegação em ambiente externo, o desconhecimento do ambiente e da sua dinâmica tornam o mapeamento e localização mais críticos. Em relação à forma como o ambiente se apresenta, pode se definir como sendo estruturados ou não estruturados. Ambientes estruturados apresentam certos padrões como por exemplo um cenário urbano, onde as ruas são delimitadas e há um certo ordenamento das estruturas, como sinalizações e construções. Um cenário não estruturado não contém padrões bem definidos, como vias e estruturas de certa forma ordenadas. Um exemplo de ambiente não estruturado é um campo aberto com vegetação.

Quando o ambiente não é estruturado depara-se com a falta de referenciais elementares (linhas, paredes, trilhas ou vias de deslocamento bem definidas) e com a baixa possibilidade de imposição de restrições. Em ambientes semiestruturado podemos considerar a existência de algum referencial e impor alguma restrição, como a existência de caminhos bem definidos e guias, porém estes devem ser considerados intermitentes, ou seja, o sistema deve ser mais tolerante à perda momentânea de referenciais.

As pesquisas em veículos terrestres autônomos têm crescido nos últimos anos e já apresentam aplicações práticas principalmente pela indústria automobilística para uso em ambiente urbano visando o aumento da segurança no uso de automóveis e na redução de acidentes. Os

desenvolvimentos mais recentes podem ser vistos em (Luettel *et al.*, 2012). A aplicação de veículos terrestres autônomos em ambientes semiestruturados e não estruturados tem também chamado a atenção. Um exemplo deste tipo de utilização são os veículos adotados em aplicações agrícolas.

## 2.1 Informação Sensorial: Visão Computacional

Um veículo móvel terrestre se desloca apoiado sobre um plano de suporte. Desta forma, o reconhecimento do chão representa uma necessidade e se caracteriza como um importante desafio ao se considerar um ambiente externo não estruturado composto por um solo com vegetação. Além da detecção do chão, é necessária uma classificação das regiões deste plano de suporte (solo) como sendo, por exemplo, uma superfície transitável ou não transitável: devido à presença de vegetação, obstáculos, buracos e demais impedimentos.

As câmeras de vídeo são adequadas para capturar tais informações que caracterizam o ambiente, principalmente, se forem usadas câmeras estéreo, das quais é possível extrair uma informação de profundidade dos elementos da cena capturada. A utilização de câmeras de vídeo demanda algumas tarefas elementares, como o ajuste de foco, a utilização de lentes adequadas ao ângulo de visão necessário e uma devida calibragem.

A calibragem das câmeras é um processo fundamental para se obter um sistema de visão computacional com representação tridimensional a partir de um par de câmeras independentes (binocular). Neste processo são calculados (estimados) os parâmetros internos e externos de cada câmera. Estes parâmetros são essenciais para a reconstrução tridimensional a partir das imagens capturadas pela câmera estéreo (Faugeras, 1993).

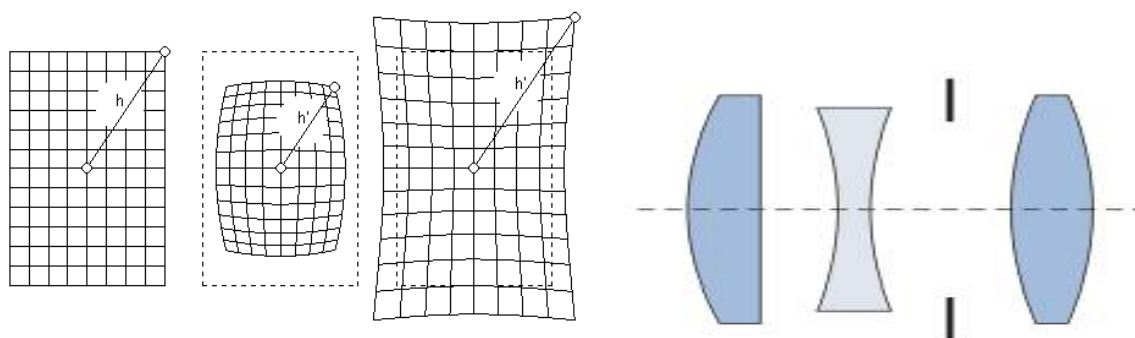
A estimação dos parâmetros internos (intrínsecos) consiste em se obter a distância focal e o ponto central da imagem, que por questões de alinhamento entre lentes e sensor pode não ser o pixel central da imagem, influenciando fortemente os demais processos de estimação (Willson & Shafer, 1994). Estes dois parâmetros definem como a imagem é formada por uma câmera estilo estenopeica (*pin-hole*) e a sua projeção perspectiva. Outros parâmetros também internos, não associados com o modelo de projeção e sim com as características físicas das lentes utilizadas, também são estimados no processo de calibragem.

No caso das câmeras de vídeo existe a necessidade de utilização de lentes, tanto para reduzir a imagem a ser projetada no sensor (filme/CCD) como para aproximá-las. As lentes comumente são estruturas com curvatura esférica, característica esta que provoca distorções óticas na imagem projetada. Estes efeitos são chamados de distorções radiais, produzindo imagens como



demonstrado na *Figura 2.1*, onde esta distorção pode ser em forma de “barril” (*barrel*) ou “almofada” (*pincushion*) (Weng *et al.*, 1992).

As distorções óticas influenciam fortemente de forma negativa o processo de correspondência entre as imagens para extrair a informação tridimensional, sendo grande fonte de ruído quando não corrigidas. É interessante notar que, matematicamente, o processo de correção das distorções óticas se baseia na curvatura esférica da lente, porém as lentes são formadas por uma composição de várias camadas com curvaturas diferentes (*Figura 2.2*), que ainda podem apresentar defeitos no processo de fabricação. Portanto, o modelo matemático utilizado na correção é sempre uma aproximação global do aparato ótico, fonte por si só de imprecisão (Willson & Shafer, 1994).



**Figura 2.1:** Distorções óticas: sem distorção (*esquerda*), distorção tipo “barril” (*centro*), distorção tipo “almofada” (*direita*)

**Figura 2.2:** Exemplo de composição ótica de lentes

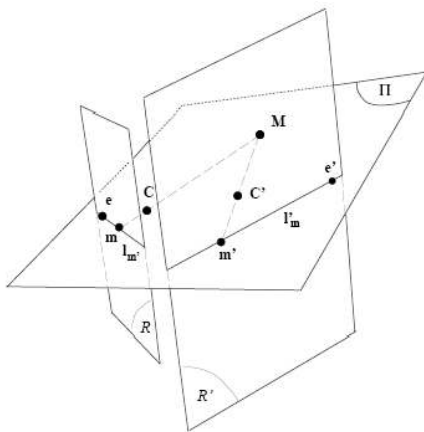
De posse dos parâmetros internos associados às distorções é necessário um pré-processamento para a sua correção que ao ser aplicado produz uma imagem mais própria para os algoritmos de correspondência que formarão a imagem tridimensional. Os parâmetros internos são também chamados de parâmetros do modelo, isto é, como a imagem é formada.

Um outro modelo de câmera geralmente associado a veículos móveis são as câmeras omnidirecionais, capazes de produzir uma imagem em 360 graus, útil para veículos com esse grau de liberdade de movimentos. A calibragem deste tipo de câmera é ainda mais complexa.

O segundo conjunto de parâmetros são os parâmetros externos (extrínsecos), também chamados de parâmetros de pose. Este conjunto diz respeito à posição da câmera em relação ao ambiente, no caso, a um sistema de coordenadas referencial. Estes são os principais parâmetros para a reconstrução tridimensional, pois são utilizados para se chegar a uma correspondência entre os pixels de cada uma das duas imagens, se as imagens estiverem alinhadas horizontalmente. Esta correspondência entre os pontos é obtida pelos princípios da geometria epipolar (*Figura 2.3*) (Faugeras, 1993).

De posse dessa correspondência, a partir do deslocamento destes pontos (disparidade) e um processo de triangulação, pode ser construída uma representação tridimensional em um mapa de profundidade. Fundamentalmente, são determinados os planos formados pelos pontos em comum em cada imagem (referenciais obtidos pela calibragem) e o ponto central de cada imagem. Esses planos no espaço (ambiente) são coincidentes, ou seja, o mesmo plano de corte. Esta projeção do plano em cada imagem definem retas, chamadas retas epipolares. Desta forma é possível fazer a correspondência dos demais pontos (não referenciais). Este alinhamento das imagens é chamado de retificação (Fusiello *et al.*, 2000), como pode ser visto na *Figura 2.4*. Pela geometria epipolar todos os planos formados pelos pontos referenciais serão concorrentes e as suas intersecções se dão em uma reta em comum, chamada linha base (*baseline*). Esta linha base determina o alcance da visão (profundidade).

Com isso, ao extrair a informação tridimensional de um ponto pelo processo de triangulação a partir da calibragem do par estéreo, a informação resultante contém as coordenadas espaciais do ponto na unidade de medida adotada. Desta forma, é possível saber a posição real de um determinado ponto, podendo ser extraída uma nuvem de pontos 3D do ambiente (*3D point cloud*). Porém, para se obter as coordenadas tridimensionais de um determinado ponto no espaço é necessário saber onde esse ponto se encontra em cada imagem (Mundy & Zisserman, 1994). Uma abordagem para extrair a terceira coordenada espacial tendo por base duas imagens da mesma cena em poses diferentes e a devida calibragem é chamada de disparidade (Qian & Qian, 1997). Este processo é computacionalmente custoso e existem diversas técnicas para a sua implementação.



**Figura 2.3:** Geometria epipolar (Faugeras, 1993)



**Figura 2.4:** Retificação (Fusiello *et al.*, 2000)

## 2.2 Mapas de Disparidade

Os mapas de disparidade se tornaram uma ferramenta bastante utilizada para a percepção tridimensional. Para robôs móveis autônomos os mapas de disparidade são muito úteis, principalmente na detecção e desvio de obstáculos. O veículo pode extrair uma informação de profundidade que servirá como parâmetro para a navegação, por exemplo, uma região frontal de maior profundidade pode representar um caminho sem obstáculos, livre de objetos e elementos localizados em frente a câmera. A implementação direta em hardware dos algoritmos de cálculo do mapa de disparidade a partir de um par de imagens (câmera estéreo) permite a aplicação em tempo real e de forma embarcada (*Figura 2.5*) (Khaleghi *et al.*, 2008).

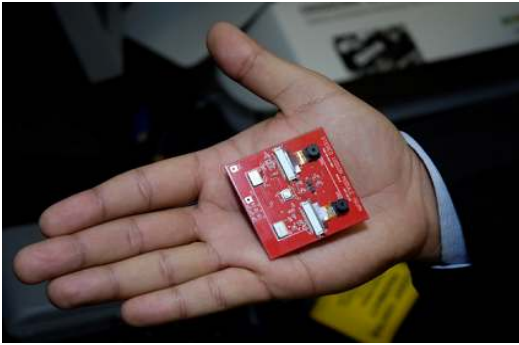
A ideia fundamental do mapa de disparidade é fornecer uma informação de profundidade relativa, mapeada diretamente na imagem bidimensional a partir do valor do pixel. Usualmente são utilizadas imagens em tons de cinza (8 *bits*), fornecendo então até 255 níveis de profundidade. Na *Figura 2.6*, o mapa de disparidade está representando os obstáculos mais próximos por tons mais claros e os obstáculos mais distantes por tons mais escuros. Desta forma, os pixels da imagem passam a representar no lugar da informação de cor ou luminosidade uma informação de profundidade sobre os elementos da cena. Os dispositivos sensores que fornecem uma imagem colorida juntamente com o mapa de disparidade têm sido denominados de dispositivos RGB-D (Imagem RGB + *Depth* / mapa de disparidade).

O sensor Kinect<sup>1</sup> da Microsoft fornece imagens RGB-D com uma precisão de 2047 níveis de profundidade. As avaliações da qualidade do mapa de disparidade produzido pelo Kinect (Khoshelham & Elberink, 2012; Pedro & de Paula Caurin, 2012) já mostraram que a relação entre o nível de profundidade e distância não é linear, onde a distância real dos objetos é gradualmente maior do que a distância estimada a medida que se afastam da câmera. O Kinect é utilizado atualmente como base de comparação pois é uma aplicação em larga escala comercial desta técnica e produz resultados práticos satisfatórios. O Kinect utiliza o método de luz estruturada para capturar a profundidade, tendo uma câmera e um emissor de infravermelho, sendo a disparidade calculada pela diferença entre os pontos da luz estruturada capturada pela câmera e o padrão desta estrutura de pontos de luz infravermelho.

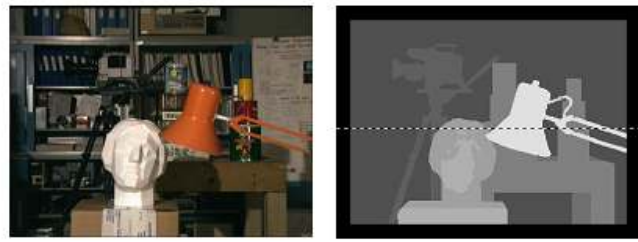
Em sistemas onde as câmeras são estacionárias (fixas em um local), como na aplicação tradicional do Kinect em jogos, técnicas de extração e supressão do fundo podem ser utilizadas para aumentar a robustez do método (Silveira *et al.*, 2010). No caso das aplicações em robótica móvel, as câmeras estão em constante movimento não sendo possível essa abordagem e produzindo efeitos negativos na imagem que dificultam a localização dos pontos em cada ima-

---

<sup>1</sup>Microsoft Kinect - <http://www.xbox.com/en-us/kinect>



**Figura 2.5:** Sistema de visão tridimensional embarcado baseado em mapa de disparidade (Khaleghi *et al.*, 2008)



(a)

(b)

**Figura 2.6:** Estéreo: (a) imagem RGB (b) disparidade referencial  
*University of Tsukuba*

gem para o cálculo da disparidade, fazendo-se necessários outros métodos para minimizar estes efeitos. Outra característica a ser apontada em relação ao Kinect é sua inviabilidade de uso no ambiente externo, ou seja, com iluminação natural - isto se deve ao fato da tecnologia empregada se basear em luz estruturada e requerer iluminação artificial (infravermelho) a qual sofre interferência na presença de luz solar. Em ambiente externo são utilizadas câmeras convencionais, onde o conjunto estéreo pode ser constituído tanto por câmeras independentes permitindo maior versatilidade de poses, como no uso de câmeras do tipo STOC (*STereo On a Chip*) que são mais estáveis em relação à manutenção da calibragem.

## 2.3 Espaço Tridimensional

As abordagens de navegação autônoma de veículos terrestres mais usuais se baseiam em grande parte na restrição de uma visão planar do ambiente, ou seja, um mundo plano. De fato, isto simplifica os modelos de representação do espaço, consciente das suas limitações em relação à fidelidade de representação do mundo real. Para ambientes controlados e até mesmo problemas reais restritos a abordagem planar é bem sucedida. Já a abordagem tridimensional é mais geral do que a abordagem planar pois descreve o espaço em todas as suas reais dimensões. Portanto, ela contempla o caso planar como caso particular. Por outro lado, o custo computacional associado se torna inerentemente mais elevado devido aos aumento da complexidade e dimensionalidade dos modelos.

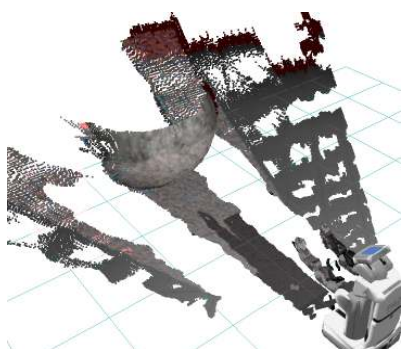
No que diz respeito a recursos computacionais, a computação embarcada atual já viabiliza diversas aplicações onde modelos mais completos e complexos eram impraticáveis até pouco tempo atrás. Isto se traduz não só em maior realismo quando se trata da relação com o utilizador

do sistema, como também uma maior precisão devido aos modelos de representação espacial mais completos. É neste sentido que as abordagens tridimensionais, no que diz respeito à percepção do ambiente onde se inserem os robôs móveis, têm sido buscadas pela comunidade científica atualmente.

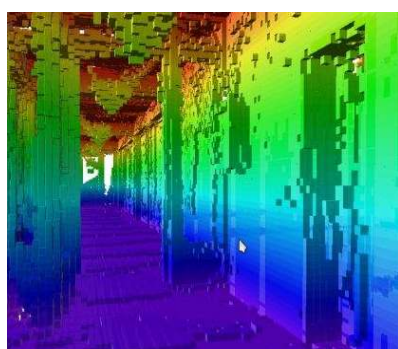
Os sensores *rangefinder* tem desempenhado papel importante na percepção espacial por produzirem dados precisos a um baixo custo computacional pois não há a necessidade de transformação do dado capturado, como ocorre no caso das câmeras. Por outro lado são sensores extremamente especializados, com características que por vezes limitam sua aplicação real - são equipamentos sensíveis, com custo ainda elevado e geralmente não podem ser cobertos por uma proteção, por exemplo. Já câmeras de vídeo são de uso geral sendo largamente utilizadas em diversas aplicações.

Conforme descrito na *Seção 2.2*, a partir de uma câmera estéreo é possível extrair a informação tridimensional do ambiente a partir da disparidade entre as imagens. Tais dados são compostos em um nuvem de pontos (*Figura 2.7*), e assim como para os sensores tipo *rangefinder*, os métodos aplicados à interpretação e classificação se dão a partir destes pontos.

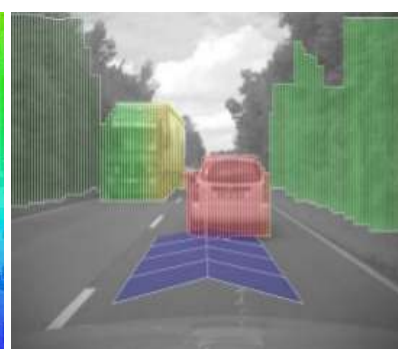
Uma nuvem de pontos nada mais é do que um conjunto de pontos no espaço sem informação de correlação, apenas os dados de pontos no espaço. No caso das câmeras, a informação cromática pode ainda ser agregada a estes pontos. As câmeras apresentam outra vantagem por não só produzirem pontos tridimensionais como sendo possível extrair características que permitem uma interpretação da cena não necessariamente tridimensional.



**Figura 2.7:** Nuvem de pontos a partir de câmera estéreo - Willow Garage



**Figura 2.8:** Mapa de ocupação (*OctoMap*) (Hornung *et al.*, 2013)



**Figura 2.9:** Segmentação por *stixels* (Badino *et al.*, 2009)

Devido à sua dimensionalidade, a informação a ser tratada tem complexidade cúbica comparada a complexidade quadrática no caso planar. O tratamento da informação é volumétrico levando à necessidade de estruturas de dados correspondentes. Largamente utilizada em computação gráfica, a representação por *voxels* é uma abordagem tradicional, pois permite discretizar

o espaço em sólidos cúbicos, reduzindo assim o espaço de busca. Uma técnica eficiente para a representação de mapas tridimensionais é apresentada em (Hornung *et al.*, 2013) e se baseia no conceito de *octrees*. Nesta abordagem o espaço é representado por um *octomap* (Figura 2.8), de acordo com a ocupação do espaço. A informação da ocupação do espaço é obtida através de uma árvore estruturada de octantes discretizados. A árvore pode ser fornecida de forma recursiva por subdivisão destes octantes, podendo gerar então um mapa do espaço em diferentes resoluções.

Outra abordagem de representação da informação tridimensional da cena em agrupamentos por volumes é apresentada no trabalho de (Badino *et al.*, 2009). Neste trabalho é introduzido o conceito de *stixel*, que representa um determinado objeto a partir de sua projeção a partir do solo. O *stixel* é então um paralelepípedo (Figura 2.9) que representa uma região ocupada na cena. Esta abordagem tem sido aplicada em reconhecimento de objetos na cena, como veículos e pessoas (Benenson *et al.*, 2012). O *stixel* é dado por uma base retangular e uma altura tendo o plano do chão como origem (representação também chamada de 2,5 dimensões). Apesar de ser uma representação restritiva do espaço tridimensional tem se demonstrado vantajosa principalmente em relação ao seu custo computacional (Benenson *et al.*, 2011).

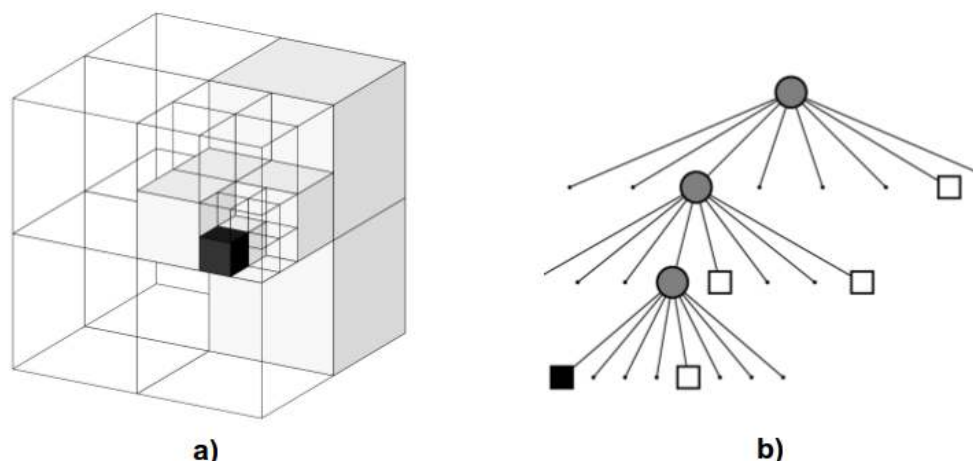
### 2.3.1 OctoMap

OctoMap é um método apresentado em (Hornung *et al.*, 2013) para representar a ocupação espacial em um modelo probabilístico. É baseado em uma *octree*, uma estrutura de dados bem conhecida pela comunidade de computação gráfica, utilizada para subdividir o espaço por octantes em uma estrutura em árvore (Figura 2.10). As principais vantagens do OctoMap é fornecer uma estrutura de dados atualizável, flexível, compacta e permitir uma representação adequada e completa de um modelo tridimensional.

A principal ideia por trás do OctoMap é armazenar a probabilidade de ocupação do espaço em seus nós da *octree*. Para que seja possível a atualização constante destas probabilidades e de maneira direta, são computadas e armazenadas o *log* da razão de possibilidades (*logOdds*). o *log* da razão de possibilidade de uma probabilidade é definido pela equação 2.1, sendo  $p$ , a probabilidade.

$$\logOdds(p) = \ln \frac{p}{1-p}, p \in (0, 1) \quad (2.1)$$

Desta forma, a cada leitura do sensor, ou seja, a cada nova percepção do espaço, é possível somar-se incrementos na probabilidade de um espaço estar ocupado (acima de 50%) ou livre (abaixo de 50%). O OctoMap é inicializado com o valor de *logOdds* em 0, que corresponde a



**Figura 2.10:** Octree: a) Espaço subdividido por octantes, b) Representação em forma de árvore

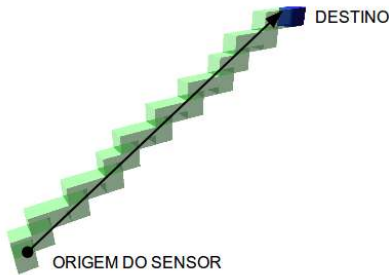
uma probabilidade de ocupação de 50%, ou simplesmente, uma ocupação espacial desconhecida. A partir deste estado inicial é possível continuamente atualizar a probabilidade de estar livre ou ocupado somando-se as novas percepções.

Como é possível somar os incrementos infinitamente, isto pode levar a um excesso de confiança sobre um determinado espaço estar livre ou ocupado. Para contornar esse efeito os autores propuseram uma política de atualização com um travamento. Este travamento força o valor de atualização em um limiar inferior e um limiar superior, limitando em um intervalo de confiança. Os valores predefinidos são de 12% e 97%, equivalente a  $-2$  e  $3,5$  em  $\log Odds$ . Esta estratégia de travamento dos valores de atualização tem também uma importância na característica de compactação proposta pelos autores do OctoMap, assim, quando todos os nós de um octante atingem o mesmo valor podem ser aglutinados em apenas um nó.

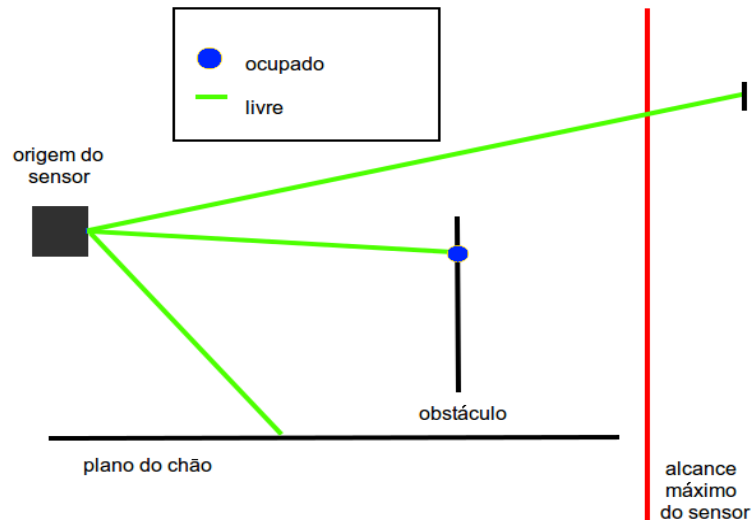
Por se tratar de um modelo probabilístico, a cada observação efetuada pelo sensor é atribuída ao OctoMap o nível de confiança daquela observação. A forma como o dado é capturado pelo sensor também é levada em conta para determinar a ocupação espacial, desta forma é definido um modelo sensorial para atualizar a OctoMap.

O modelo sensorial definido pelos autores em (Hornung *et al.*, 2013) se baseia no comportamento de um sensor a laser. No caso, o modelo sensorial adotado é baseado em raios a partir de um ponto de origem. Desta forma, para cada ponto observado, o caminho a partir da origem até o ponto é marcado como espaço possivelmente vazio e o ponto em si é marcado como espaço possivelmente ocupado. Estes raios são calculados tridimensionalmente utilizando o método de *ray-casting* (Roth, 1982), conforme exemplo apresentado na *Figura 2.11*. A *Figura 2.12* apresenta, em verde, as diferentes formas como o espaço livre pode ser marcado a partir dos

raios projetados. Em azul, o espaço marcado como ocupado. A linha em vermelho representa o alcance máximo do sensor.



**Figura 2.11:** *Ray-casting*: são obtidos os pontos que representam um raio da origem até um destino



**Figura 2.12:** Modelo sensorial para a marcação da ocupação do espaço

## 2.4 Localização

Localização de um veículo terrestre é usualmente definida como a habilidade de se determinar a sua posição bidimensional ( $x$ ;  $y$ ) e a orientação ( $\theta$ ) dentro de um determinado ambiente. A informação de localização de uma entidade móvel é essencial para determinar seu posicionamento em relação ao ambiente. No contexto de navegação, o posicionamento é fundamental para que se mantenha o curso em uma trajetória determinada.

Para a construção de um mapa a partir da exploração em ambiente desconhecido torna-se necessário que se saiba a posição onde os dados estão sendo capturados com base em um determinado sistema de referência de coordenadas. Uma abordagem para a localização (determinação da posição) é a utilização de um sistema de posicionamento global por satélite (GPS - *Global Positioning System*). Este tipo de sistema provê uma localização absoluta em um sistema de referência global (latitude e longitude).

Outra abordagem de localização é a partir da odometria de veículo, na qual o posicionamento é dado pelos deslocamentos e orientações do veículo. Este tipo de abordagem provê uma localização relativa (a um ponto de início). Ambas abordagens estão sujeitas a erros de medição e a ruídos inerentes ao processo de aquisição dos dados.



Já no caso de dispositivos como o odômetro, utilizados para manter um controle da localização por *dead-reckoning* (Dudek & Jenkin, 2000), a confiabilidade é ainda menor pois há grande tendência de discrepância uma vez que o erro é cumulativo. Estas discrepâncias podem ser resultantes de derrapagens das rodas, erros de estimação da real distância percorrida ou mesmo das alterações involuntárias de direção do veículo.

Uma possível melhoria da precisão do GPS pode ser conseguida usando um sistema de correção DGPS (GPS diferencial), onde antenas em solo são utilizadas para aumentar a precisão do local estimado pelos sinais dos satélites. No entanto, nem sempre este tipo de abordagem é possível pela falta desta infraestrutura no local desejado, além de possuir um custo bem mais elevado em relação a abordagem tradicional.

No Brasil, o IBGE<sup>2</sup> disponibiliza serviços de posicionamento de precisão através da RBMC<sup>3</sup> (Rede Brasileira de Monitoramento Contínuo). O GPS tradicional possui um erro que pode chegar até 15 metros (Bevly & Cobb, 2010) da posição estimada informada em relação a posição real, enquanto um DGPS pode reduzir este erro médio para menos de 1 metro, podendo chegar a poucos centímetros com serviços de correção comerciais mais sofisticados (Ozguner *et al.*, 2011). É interessante observar que em aplicações de GPS em roteiros rodoviários (e urbanos) a existência de um mapa georreferenciado permite um certo ajuste de coerência que visualmente dá a impressão de uma maior precisão, porém em campo aberto esta abordagem não é possível.

A localização e o mapeamento é uma necessidade recorrente nas aplicações em robótica móvel. A caracterização do veículo como autônomo de fato pode vir a ser dada essencialmente pela sua capacidade de tratar esta questão. Para que um robô possa ir de um ponto ao outro do ambiente ele deve conhecer, pelo menos, a sua localização e a localização do seu ponto de destino. Esta localização está associada a um mapa e a uma posição relativa conhecida.

Um problema central na robótica móvel é quando não há o conhecimento da localização inicial do veículo e para se localizar de forma relativa há a necessidade de uma mapa que também é inexistente a priori. Este tipo de problema é tratado através de uma abordagem denominada de SLAM (*Simultaneous Localization and Mapping*), onde não há nenhuma informação inicial de localização e mapeamento, e uma depende da outra para ser obtida. Um algoritmo proposto para solucionar o SLAM é apresentado em (Thrun *et al.*, 2004) que utiliza uma abordagem probabilística onde as estimativas tanto da posição como do mapeamento são atualizadas uma a partir da outra a medida que dados sensoriais são coletados.

A visão computacional tem sido aplicada em soluções para o tratamento de SLAM como fonte de informação sensorial. Outros equipamentos, como sensor a laser utilizado em (Lategahn *et al.*, 2011) e até mesmo sensores inerciais (IMU - *Inertial Measurement Unit*), como

<sup>2</sup>IBGE – <http://www.ibge.gov.br>

<sup>3</sup>RBMC – <http://www.ibge.gov.br/home/geociencias/geodesia/rbmc/rbmc.shtm>

apresentado em (Angermann & Robertson, 2012), são aplicados e combinados com o propósito de aumentar a acurácia destas medições.

## 2.5 Mapeamento

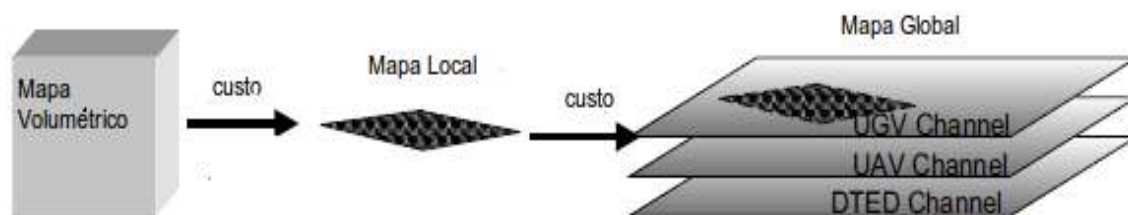
Os mapas têm papel importante na tarefa de navegação (*Seção 2.6*), tanto quando fornecidos previamente como quando construídos no momento em que o veículo se desloca no ambiente. O mapa para navegação geralmente se caracteriza como local ou global (Thrun, 1998). O mapa global provê informações mais abrangentes sobre o espaço onde o veículo irá se deslocar, principalmente fornecendo informações sobre obstáculos estáticos e zonas navegáveis preferenciais, como estradas. Já os mapas locais têm o papel de expressar o ambiente dentro do raio de atuação dos sensores, ou seja, uma noção localizada próxima ao veículo. Este mapa local contém a visão mais atual do ambiente, contemplando obstáculos dinâmicos.

Quando não há informação prévia que possa ser utilizada como mapa global é usual uma navegação prévia para o reconhecimento do ambiente antes que se efetuem tarefas específicas de navegação. Este reconhecimento inicial pode ser feito tanto de forma autônoma apenas vagando no ambiente como por uma navegação teleoperada. Em Kelly *et al.* (2006), o mapa local é utilizado para atualizar um mapa global que contém informações a priori do ambiente e atualizado também por dados capturados por um outro veículo. Neste caso, o mapa global ainda é utilizado para agregar as informações de mapeamento e compartilhar tais informações.

Na *Figura 2.13* pode ser vista esta arquitetura de mapeamento do ambiente, onde um mapa volumétrico gerado pela percepção é convertido em um mapa local de navegação a partir de custos calculados e o mapa local atualiza os custos de um mapa global. O mapa global é uma composição de informações de elevação do terreno (DTED - *Digital Terrain Elevation Description*) obtidas a priori, informações capturadas por um veículo aéreo (UAV - *Unmanned Aerial Vehicle*) e dados armazenados da navegação do próprio veículo (UGV - *Unmanned Ground Vehicle*).

### 2.5.1 Mapa de Navegabilidade

Os mapas de navegabilidade expressam as informações básicas para se ter um conhecimento necessário para se deslocar em um ambiente de forma deliberativa, ou seja, de forma planejada. Para este planejamento, o mapa de navegabilidade pode ser transformado em outras representações, como no caso do mapa de custos. Estes mapas de custo são utilizados por algoritmos de busca por minimização para encontrar um caminho para navegação. Algoritmos de planejamento de navegação por árvore de expansão espacial como o T-RRT (Jaillet *et al.*, 2008) e



**Figura 2.13:** Arquitetura de mapeamento: os dados volumétricos provenientes dos sensores produzem um mapa local de custos, que atualiza um mapa global de custos composto por informações mapeadas a priori e provenientes de um veículo aéreo (Kelly *et al.*, 2006)

baseados em malha de estados como em Howard & Kelly (2007), se baseiam na informação de mapa de custos e no princípio de encontrar o caminho de menor custo.

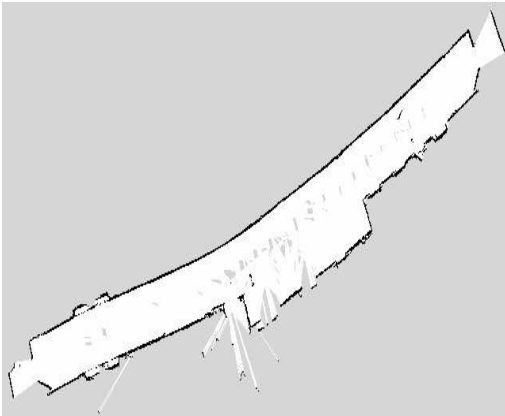
Os mapas de navegabilidade podem se apresentar em diversas formas, como carta de navegação marítima, mapas rodoviários, mapas topológicos e semânticos ou mapas que marquem apenas as áreas navegáveis e não navegáveis. Mapas que marcam as áreas navegáveis são também chamados de mapas de ocupação e podem ser convertidos diretamente para mapas de custos. A figura *Figura 2.14* apresenta um exemplo de mapa de ocupação, onde em preto está marcado o espaço ocupado, o branco o espaço livre e o cinza o espaço desconhecido.

Estes mapas podem apresentar a informação em forma métrica, contendo a localização em termos de coordenadas ou de forma lógica, organizando a informação em uma forma estruturada por um conjunto de regras (Thrun, 1998). Também, podem conter uma combinação das duas representações (Thrun *et al.*, 1998). A *Figura 2.15* é um exemplo de mapa topológico, que representa as linhas do metrô da cidade de São Paulo.

## 2.6 Navegação

A navegação em campo aberto se torna um problema de tomada de decisões difícil, devido à ausência de um mapa global atualizado, a falta de referenciais locais e o possível excesso de ruído dos sensores pode produzir pouca informação útil para o sistema. Apesar do veículo estar buscando um destino preestabelecido, dirigir-se sempre em linha reta pode não ser a melhor decisão, ou mesmo pode ser impossível devido aos obstáculos presentes neste caminho. Além disto, a indicação da posição atual (localização) pode não ser totalmente confiável.

O algoritmo básico de navegação de um veículo autônomo pode se basear em um princípio estratégico similar ao utilizado por uma pessoa portadora de deficiência visual, ou seja, passos controlados e o constante monitoramento do ambiente ao seu redor. Uma pessoa consegue



**Figura 2.14:** Mapa de ocupação planar: Em branco o espaço livre, em preto o espaço ocupado e em cinza o espaço sem informação de ocupação - ROS



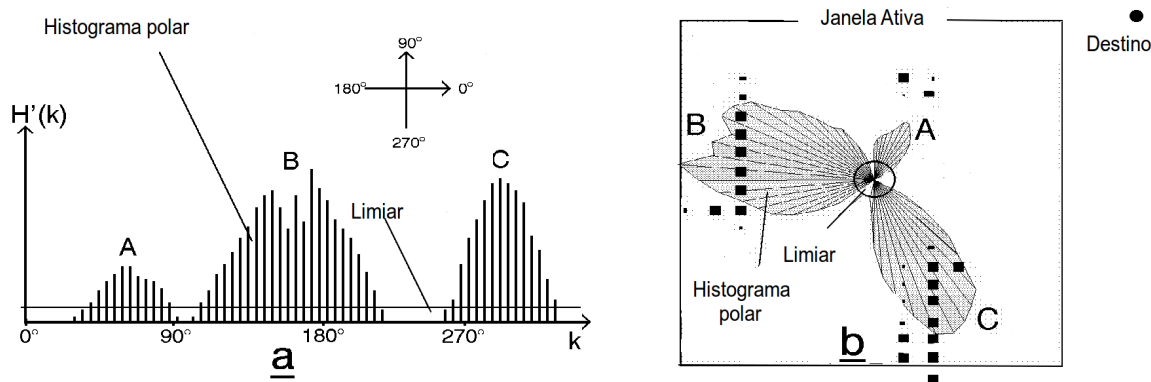
**Figura 2.15:** Mapa topológico do metrô da cidade de São Paulo - Governo do Estado de São Paulo

achar, de modo intuitivo e racional, um caminho em direção a um determinado destino e evitar colisões com obstáculos em seu caminho. No caso de um veículo autônomo, este não possui essa capacidade intrínseca de navegação e desvio de obstáculos, devendo ser dotado de algum recurso computacional e/ou físico para tal. Existem diversos algoritmos clássicos de estratégia de navegação, como por exemplo, o algoritmo do *bug* (Choset *et al.*, 2005) e suas variantes (Taylor & LaValle, 2009), que se baseiam na detecção de bordas (obstáculos) e executam a locomoção aproximando-se a elas e fazendo o seu contorno.

A navegação autônoma de um veículo pode então ser executada de forma reativa, onde a percepção do sistema tem o principal papel de detecção de obstáculos que obstruem ou impedem o movimento apenas a cada passo dado. Já em uma navegação deliberativa se espera que haja a determinação de uma trajetória que guie os movimentos planejando um maior número de passos. Antecipar os passos que o veículo irá executar para alcançar o objetivo permite otimizar o caminho, gerar trajetórias coerentes com a dinâmica de movimento do veículo e com menor custo em número de manobras e tempo.

Ambas abordagens, reativa e deliberativa, são aplicáveis conjuntamente em um sistema de navegação autônoma. Em Luetzel *et al.* (2012) são definidos três tipos de abordagens de navegação; a navegação global, a navegação reativa e a navegação guiada que é uma forma intermediária entre o deliberativo (global) e o reativo. A navegação reativa está associada mais diretamente ao controle do veículo, tendo a percepção uma influência direta na atuação do veículo. A navegação deliberativa está mais associada ao mapeamento do ambiente e do planejamento de trajetórias, por isso é definida como global.

O método VFH (*Vector Field Histogram*) é um exemplo desta abordagem intermediária e foi projetado para ser utilizado em tempo real diretamente a partir dos dados dos sensores sem um mapeamento, gerando um histograma polar (*Figura 2.16*) onde os picos representam a proximidade dos objetos em relação ao veículo e os vales os espaços livres. Idealizado para ser utilizado com sensores do tipo sonar, leva em conta as questões inerentes a ruídos próprios desta classe de sensor e as características cinemáticas do veículo.

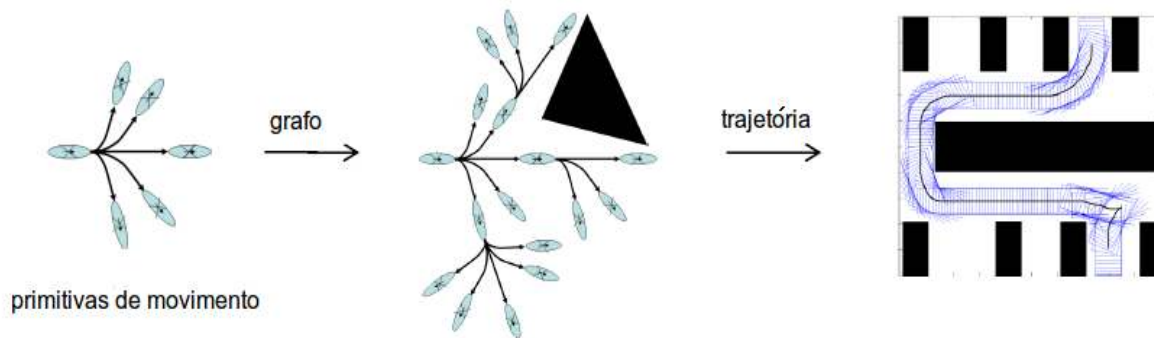


**Figura 2.16:** VFH - (a) Gráfico do histograma, onde  $H'(k)$  representam as forças de repulsão aos obstáculos nas direções  $k$ .  
 (b) Representação espacial em relação aos obstáculos (retângulos em preto) - A área circulada representa a posição do robô.  
 (Borenstein & Koren, 1991)

Enquanto o VFH tem um comportamento mais reativo, em um cenário onde um mapa global mesmo que incompleto é fornecido, pode ser adotado um método deliberativo para o planejamento da trajetória até um destino. Já a navegação global requer um planejamento (global), que consiste em determinar uma trajetória válida para o veículo, tomando por base um mapa de navegabilidade. O planejamento leva em consideração o modelo cinemático do veículo para que a trajetória seja factível. Neste planejamento os possíveis obstáculos e obstruções também devem ser avaliados afim de evitar colisões e bloqueios do veículo.

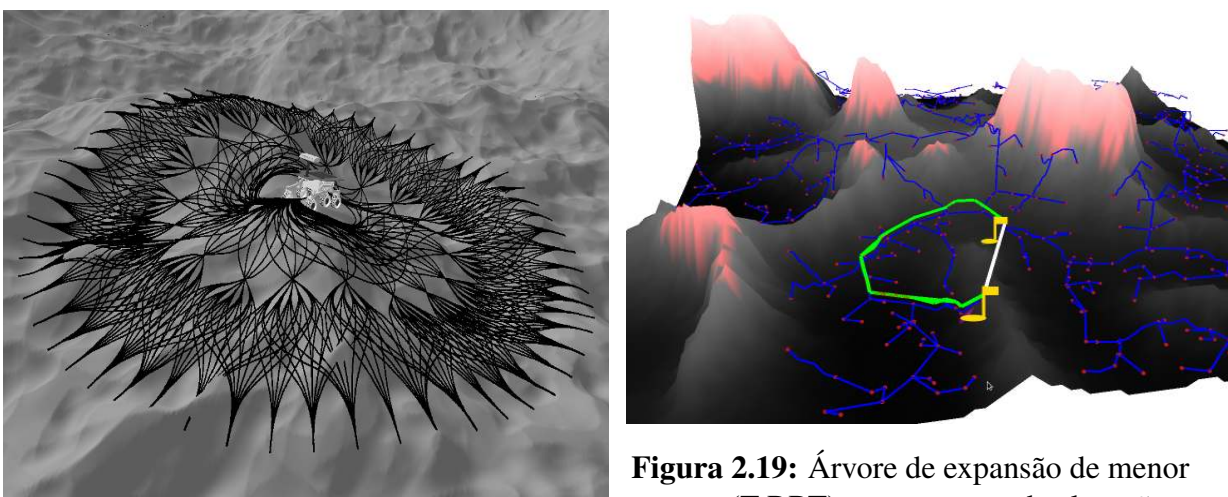
Os algoritmos de geração de trajetória baseados em amostras tem sido adotado em robótica móvel com maior ênfase principalmente pela aplicabilidade prática, este tipo de abordagem leva em consideração a cinemática do veículo gerando trajetórias realistas com custo computacional baixo (Sucan *et al.*, 2012). Estes métodos inicialmente constroem um grafo baseado em amostras de trajetórias que o veículo pode executar, onde geralmente se limita a algumas manobras afim de reduzir as ramificações do grafo (Bhattacharya *et al.*, 2012). Após esta etapa pode ser executado um método gráfico de busca por um caminho de menor custo sobre o mapa de navegabilidade traduzido em custos, como por exemplo o  $A^*$ . Na *Figura 2.17* pode ser visto um exemplo de planejamento baseado na construção de uma malha de amostras de trajetórias. Um

exemplo deste tipo de malha de expansão de amostras pode ser visto na *Figura 2.18* referente ao trabalho apresentado em Howard & Kelly (2007).



**Figura 2.17:** Planejamento de trajetória por amostras - *SBPL*

Outra abordagem que busca um caminho de menor custo sobre um mapa de custos é através de árvores de expansão. No caso do algoritmo T-RRT (Jaillet *et al.*, 2008), não é levado em conta a dinâmica do veículo para traçar a rota, pois se trata de um método apenas de busca e não necessariamente de planejamento de trajetória para veículos. Porém, em um mapa topográfico, georreferenciado, a elevação do terreno e demais elementos do mapa podem ser convertidos em custos a partir de uma função ponderada. Desta forma, o caminho retornado pelo T-RRT, pode ser utilizado em um mapa prévio para geração de pontos de GPS intermediários para a navegação em uma grande área. A *Figura 2.19* mostra uma expansão utilizando T-RRT em um mapa de elevação (DEM), os traçados em azul são a expansão, o caminho em verde é o de menor custo entre os pontos em amarelo, em branco, o caminho de menor distância.



**Figura 2.18:** Malha de expansão de amostras de trajetórias em um terreno irregular (Howard & Kelly, 2007)

**Figura 2.19:** Árvore de expansão de menor custo (T-RRT) em um mapa de elevação convertido em custos. Em verde o caminho de menor custo entre os pontos em amarelo. Em azul a expansão

## 2.7 Considerações Finais

Neste capítulo foram apresentados conceitos sobre percepção e visão computacional, criação de mapas de disparidade, representações espaciais e sobre mapas globais, mapas locais e mapas de navegabilidade. Também foram apresentados conceitos sobre localização e planejamento de trajetórias que permitem a navegação autônoma. Neste trabalho, em função da opção por ambientes externos e não estruturados, serão considerados elementos como a localização por GPS, complementada pelo uso de odometria, que irá permitir estimar a posição aproximada atual e de destino do veículo.

Para o planejamento e navegação do veículo autônomo com desvio de obstáculos, serão adotados métodos de percepção baseados em visão estéreo que permitem criar representações locais (mapas locais e mapas de navegabilidade), adotando técnicas como o uso da OctoMap, nuvem de pontos e planejamento de trajetória.

No capítulo seguinte serão apresentados trabalhos relacionados que serviram de base e inspiração para o presente trabalho.





---

## Trabalhos Relacionados

---

A navegação de veículos autônomos em ambientes externos e não estruturados está associada geralmente a aplicações diversificadas, desde aplicações para fins militares, de resgate em caso de catástrofes naturais, em aplicações agrícolas e de produção e até exploração espacial.

A própria inexistência desse tipo de equipamento para operar no desastre da usina nuclear de Fukushima ocorrido recentemente em 2011 fez com que as pesquisas e desenvolvimento motivadas neste tipo de aplicação ganhasse um novo foco de atenção. Exemplo disto é o programa DRC (*Darpa Robotics Challenge*) lançado em 2012 pelo Departamento de Defesa dos Estados Unidos (DARPA), com o propósito de fomentar o desenvolvimento de robôs humanoides e de veículos terrestres capazes de executarem tarefas complexas em situações de desastres.

Outra iniciativa envolvendo o desenvolvimento deste tipo de equipamento é o ELROB (*European Robotic Trial*), uma competição anual europeia que ocorre desde 2006, sendo intercalada anualmente entre objetivo militar e objetivo civil. Exemplos de veículos que participaram de edições do ELROB podem ser vistos na *Figura 3.1*.

### 3.1 LAGR

O LAGR (*Learning Applied to Ground Robots*) (Jackel *et al.*, 2006) foi um projeto do departamento de defesa americano (DARPA) que ocorreu nos anos de 2004 a 2008. O intuito do



**Figura 3.1:** Veículos participantes de edições do ELROB - [www.elrob.org](http://www.elrob.org) (divulgação)

projeto foi acelerar o desenvolvimento e pesquisas em veículos para ambientes não estruturados e percepção baseada em visão. Foi criada uma plataforma robótica e um sistema básico de navegação onde as equipes participantes do projeto deveriam apresentar melhorias que foram comparadas entre si como forma de avaliação.

A plataforma consistia em um veículo de médio porte, dotado de *encoders* nas rodas, sensores de contato (*bumpers*) frontais, uma unidade de medida inercial (IMU), quatro computadores embarcados, um GPS, dois pares de câmeras estéreo e dois sensores de distância por infravermelho. Foram distribuídos cerca de 30 veículos (*Figura 3.2*) entre as equipes participantes no período em que o projeto ocorreu.



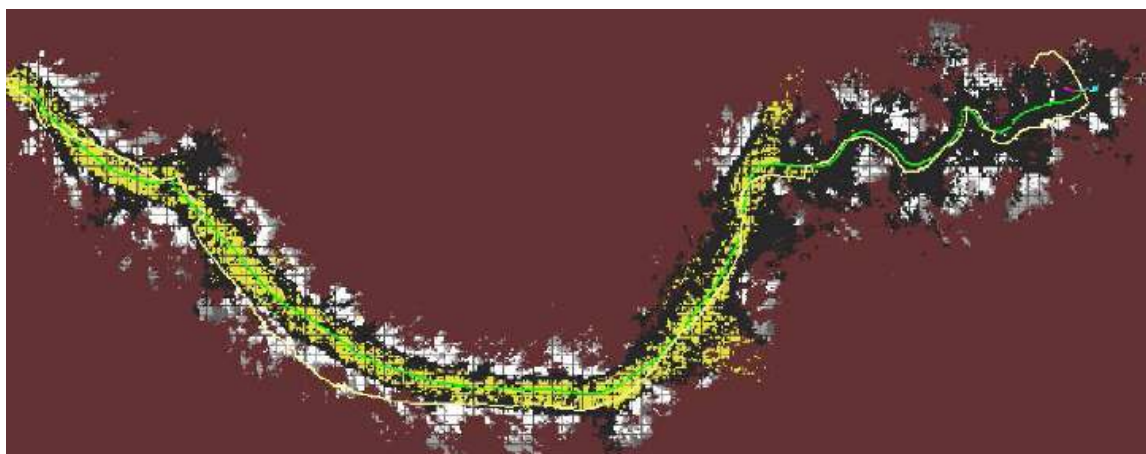
**Figura 3.2:** Veículo do projeto LAGR - NIST

A arquitetura básica do sistema do veículo consistia em um controle de navegação baseado no algoritmo DWA (Fox *et al.*, 1997) e o planejamento de navegação baseado na execução

do algoritmo  $D^*$  (Stentz, 1994) em um mapa de custos para buscar um caminho. O trabalho apresentado em Konolige *et al.* (2009) obteve resultados que se destacaram entre os demais. Para a construção do mapa de custos o sistema de percepção visual utilizava um classificador nas imagens para detectar áreas navegáveis e não navegáveis, onde a partir da visão estéreo era computada a posição espacial das regiões classificadas.

O classificador básico distinguia a cena em uma classe obstáculo e três classes livre; o plano do chão, o caminho e o espaço livre. As classes foram convertidas em custos por uma função ponderada entre os quatro pesos retornadas pelo classificador. Um exemplo do mapa de navegabilidade codificado em cores pode ser visto na *Figura 3.3*. A discretização utilizada para o mapa foi em células de 20 centímetros.

Para aumentar a acurácia da localização e reduzir erros, tanto provenientes do GPS como por erros na odometria pelas rodas provocados por derrapagens, foi desenvolvido um método de odometria visual (VO) que permitiu maior precisão na construção do mapa de navegabilidade.



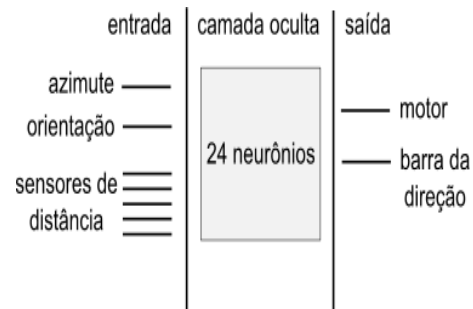
**Figura 3.3:** Mapa de navegabilidade, em amarelo o caminho livre, em preto o espaço livre, em tons de cinza os obstáculos (Konolige *et al.*, 2009)

## 3.2 RoBombeiros

No sistema RoBombeiros (Pessin, 2008) foi implementada uma solução baseada em coordenadas GPS para localização e uma rede neural artificial (RNA) para o controle e navegação. O sistema foi projetado para ambientes externos onde os veículos tinham como meta o combate a incêndios florestais. Este sistema foi testado apenas em simulação que consistia em ambientes não estruturados onde a vegetação predominante era composta por árvores. Os veículos foram capazes de desviar dos obstáculos adequadamente e se dirigir ao destino estabelecido (*Figura 3.4*).



**Figura 3.4:** RoBombeiros - Navegação em ambiente não estruturado e estratégia de combate a incêndio florestal (Pessin, 2008)



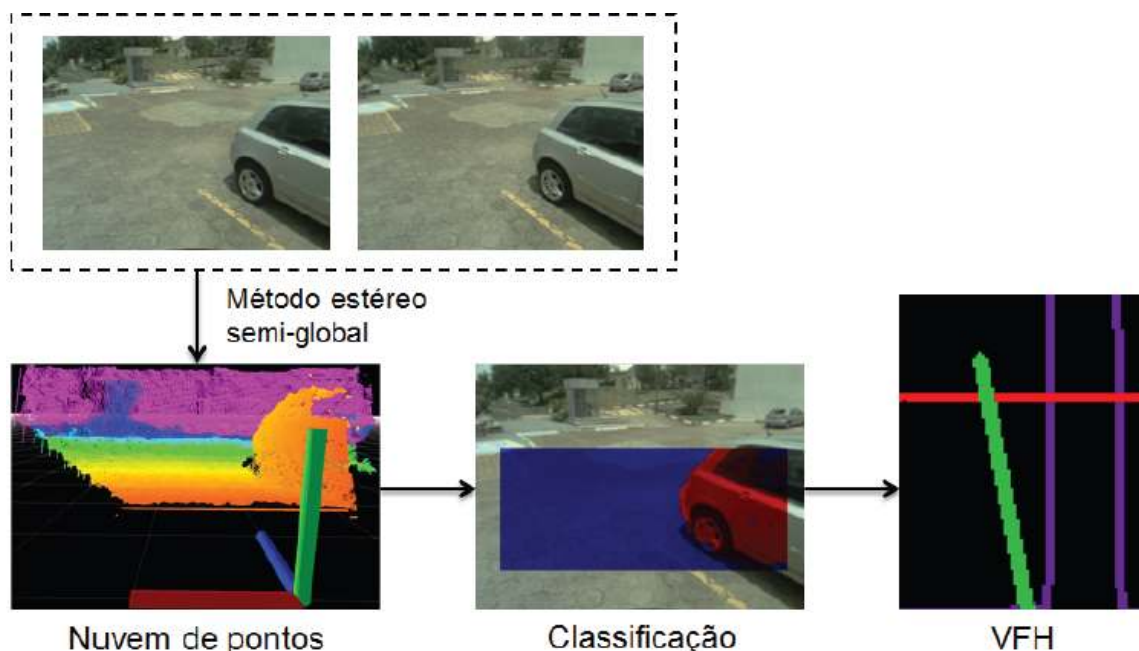
**Figura 3.5:** Topologia da RNA no simulador RoBombeiros

No caso dos RoBombeiros, o controle pela rede neural era de um comportamento basicamente reativo, onde ao mesmo tempo em que direcionava o veículo para o destino gerava comandos de desvio na presença de obstáculos próximos. Esta rede neural foi projetada usando como entradas a orientação do veículo dada por uma bússola e a sua orientação em relação ao ponto de destino calculada pelas posições de GPS, assim como informações sensoriais de distâncias de objetos presentes no raio de ação do sensor laser utilizado. Desta forma a rede neural foi treinada para gerar comandos de aceleração e esterçamento que controlavam a navegação do veículo autônomo. A topologia da rede neural utilizada no projeto RoBombeiros pode ser vista na *Figura 3.5*.

### 3.3 Projetos do Laboratório de Robótica Móvel

No trabalho de Mendes & Wolf (2011), foi utilizado um método semiglobal para calcular o mapa de disparidade das imagens capturadas pela câmera estéreo. O método semiglobal foi adotado por produzir nuvens de pontos mais densas sendo capaz de detectar pequenas saliências como guias da rua. Dada a nuvem de pontos foi utilizada uma abordagem baseada em RANSAC (Fischler & Bolles, 1981) para detectar o plano do chão, a partir deste plano foi aplicado um método de detecção de obstáculos baseado em cone (Talukder *et al.*, 2002).

Para a navegação autônoma foi utilizado o VFH (Borenstein & Koren, 1991) para desvio de obstáculos e o uso de GPS para localização. A *Figura 3.6* mostra o diagrama do sistema proposto de navegação autônoma baseada em visão.



**Figura 3.6:** Diagrama de blocos do sistema de navegação proposto em Mendes & Wolf (2011)

### 3.4 Autonomous Prime Movers

No trabalho de Bergerman *et al.* (2012) foi apresentada uma plataforma chamada APM (*Autonomous Prime Movers*) voltada para automação de colheita em pomares. Este é um exemplo de aplicação em ambiente externo em um ambiente semiestruturado. O veículo foi equipado com três sensores laser em alturas diferentes que permitem uma percepção frontal em três planos diferentes com um ângulo de 270 graus.

Para localização foi utilizado um sistema de GPS com correção utilizando duas antenas. O sistema de navegação consiste em manter o veículo em um determinado curso e velocidade constante, detectando obstáculos que bloqueiam o caminho. A mesma plataforma foi utilizada em Zhang *et al.* (2012) para trafegar em corredores de plantações utilizando um sistema autônomo baseado em visão monocular.

Na *Figura 3.7* pode ser visto o veículo em operação. Para a aplicação específica em pomares o veículo opera em modos determinados de acordo com cada situação, sendo, trafegar mantendo-se no corredor cercado pelas árvores do pomar, trocar de corredor no fim de cada trajeto fazendo uma meia volta e entrar em um novo corredor. Um aspecto interessante da

aplicação deste veículo é a necessidade da adequação do ambiente para que este possa ser utilizado (foi adotada uma poda específica nas plantações, formando uma espécie de "parede" com as árvores). Porém, esta adequação além de criar uma certa estruturação que o sistema pode se beneficiar, também é condizente com um melhor aproveitamento do espaço e aumento da produtividade do pomar.



**Figura 3.7:** Veículo autônomo para o auxílio de colheita em pomares (Bergerman *et al.*, 2012)

### 3.5 Considerações Finais

Os trabalhos apresentados nesta seção demonstram diferentes possibilidades de aplicações dos sistemas automatizados de navegação em ambientes semiestruturados e não estruturados.

É importante destacar que alguns destes trabalhos são baseados no uso de sensores a laser, porém o tipo de aplicações neles descritos pode ser tratado pela proposta apresentada nesta dissertação de mestrado, que faz uso de um sistema de percepção baseado em visão. Os trabalhos relacionados foram descritos em ordem cronológica e servem de base para o desenvolvimento deste projeto.

Alguns trabalhos ainda se encontram ativos, como no caso do projeto *Autonomous Prime Movers* e das competições do ELROB, que no ano de 2014 tem como objetivo aplicações militares.

---

# Metodologia

---

A construção de sistemas de navegação autônoma requer a aplicação de diversas técnicas e algoritmos, também, o uso de diversos sensores faz com que haja a necessidade da fusão dos dados capturados (Luettel *et al.*, 2012).

A simulação desempenha um importante papel no desenvolvimento destes sistemas, sendo a modelagem do sistema em um ambiente virtual um passo importante para viabilizar a execução de experimento em simulação. Os experimentos simulados são replicáveis e repetíveis mais facilmente, provendo um numero maior de dados para análises do comportamento dos algoritmos aplicados. Grande parte dos algoritmos aplicados na robótica móvel dependem de um ajuste fino de parâmetros para se adequarem às características do ambiente. Com a simulação, estes parâmetros e seus efeitos podem ser melhor compreendidos e ajustados para o caso real. Um ambiente de simulação permite que um número maior de pesquisadores tenham acesso a mesma plataforma podendo conduzir experimentos em paralelo e em componentes isoladamente.

## 4.1 Plataforma de Hardware

Este trabalho foi desenvolvido junto ao LRM<sup>1</sup> (Laboratório de Robótica Móvel). Diversos trabalhos relacionados ao desenvolvimento de veículos autônomos e robôs móveis inteligentes

---

<sup>1</sup>LRM - Laboratório de Robótica Móvel do ICMC/USP - <http://www.lrm.icmc.usp.br>

vêm sendo pesquisados e desenvolvidos neste laboratório, destacando-se, a pesquisa e uso de sistemas de navegação baseados em visão computacional. O LRM possui atualmente duas plataformas de teste para aplicações de veículos móveis autônomos que foram adquiridas pelo INCT-SEC: os veículos CaRINA<sup>2</sup> I e II (Fernandes *et al.*, 2014). Para realizar as atividades de validação em ambiente real foi utilizada a plataforma CaRINA I (*Figura 4.1*) que possui a automatização dos seus controles de aceleração e esterçamento. A arquitetura completa da plataforma do CaRINA I foi descrita em (Fernandes *et al.*, 2012).

Foram ainda utilizados uma unidade de medida inercial (IMU) com GPS integrado, modelo Xsens MTi-G (*Figura 4.2*), uma câmera estéreo modelo Point Grey Bumblebee 2 (*Figura 4.3*) e um computador Intel Core i7 Quad-Core 3, 5 GHz com 16 Gb de memória. A câmera estéreo tem um afastamento horizontal entre as lentes de 12 centímetros. Esta câmera possui lentes com distância focal de 6 milímetros, o que dá um campo de visão de 43 graus. A resolução utilizada foi de 640x480 pixels.



**Figura 4.1:** CaRINA I - Carro elétrico automatizado



**Figura 4.2:** IMU com GPS integrado Xsens MTi-G - Xsens



**Figura 4.3:** Câmera estéreo Point Grey Bumblebee 2 - Pointgrey

A marcha reversa do veículo CaRINA I não se encontra automatizada. Como não é possível inverter a marcha do veículo de forma autônoma foi limitado ao planejamento de trajetórias e ao controle apenas de manobras para frente. Esta limitação é prejudicial a um sistema de navegação em ambiente não estruturado, visto que manobras evasivas acabam sendo necessárias, mas por ser uma limitação da plataforma de validação foi limitado também no ambiente simulado para que o comportamento da navegação e as trajetórias planejadas fossem equivalentes. Também,

<sup>2</sup>CaRINA - Carro Robótico Inteligente para Navegação Autônoma - <http://www.lrm.icmc.usp.br/carina>



foi apenas utilizada uma câmera estéreo instalada na frente do veículo permitindo apenas uma percepção frontal do ambiente. Estas limitações são apenas dos equipamentos utilizados, não afetando a arquitetura do sistema proposto nem as técnicas de navegação aplicadas, apenas limitando o campo de visão e a mobilidade do veículo em termos de manobras.

## 4.2 ROS

O desenvolvimento do sistema baseou-se no *framework* do ROS (*Robot Operating System*) (Quigley *et al.*, 2009), versão “G” (Groovy). O ROS se caracteriza por um conjunto de ferramentas e bibliotecas para o desenvolvimento de aplicações robóticas. Sendo disponibilizado como *Open Source*, tem sido adotado mundialmente por laboratórios de pesquisa e desenvolvimento em robótica. Produtos comerciais também fornecem *drivers* e ferramentas dentro dos padrões do ROS. Um exemplo de plataforma completa é o robô PR2<sup>3</sup> e os produtos da Clearpath Robotics<sup>4</sup> com plataformas terrestres, aéreas e aquáticas.

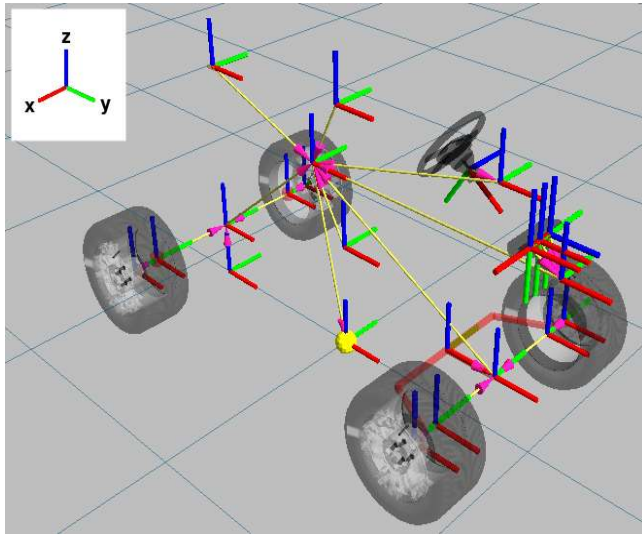
O *framework* ROS conta como principais componentes um sistema distribuído, permitindo através de um mecanismo de troca de mensagens, um desacoplamento entre os módulos de um sistema robótico, como por exemplo, um subsistema de percepção e um subsistema de navegação. Desta forma também permite uma abstração entre o *hardware* real e o equivalente virtual em simulação. Outro componente central no ROS é o TF (*Transform*) responsável por armazenar todos os sistemas de referências existentes no robô, como por exemplo a posição dos sensores e partes móveis (*Figura 4.4*). Mantém constantemente atualizadas as transformações (rotação e translação) necessárias para passar de um sistema de referência para outro, operação importante para por exemplo fundir dados de sensores. Como essas transformações podem ser feitas de um sistema de referência para qualquer outro a relação entre sistemas deve ser única, desta forma, a estrutura do TF é em forma de árvore conforme a *Figura 4.5*.

Dentre as diversas ferramentas disponibilizadas pelo ROS, a ferramenta de visualização RViz permite obter de forma gráfica diversas informações do sistema, como dados dos sensores, mapeamentos e posicionamento calculados pelo robô e ter uma visão geral do comportamento do mesmo em relação ao ambiente. As principais linguagens de programação utilizadas são C++ e Python, porém é possível implementar componentes em outras linguagens como Java e Lisp. O sistema operacional utilizado foi o Ubuntu Linux, versão 12.04.

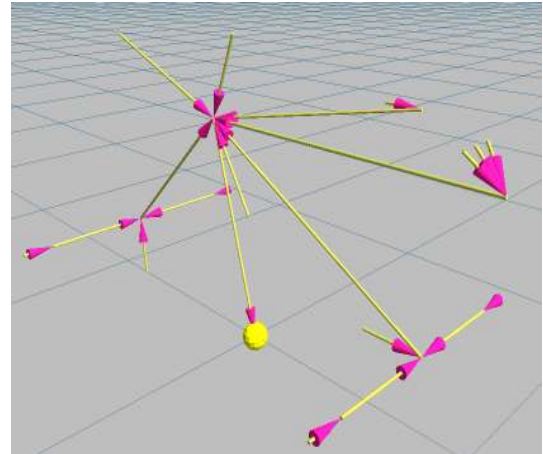
---

<sup>3</sup>PR2 - Personal Robot 2 - <http://www.willowgarage.com>

<sup>4</sup>Clearpath Robotics - <http://www.clearpathrobotics.com>



**Figura 4.4:** Sistemas de referência de coordenadas



**Figura 4.5:** Representação em árvore, a esfera amarela representa a raiz e as setas em magenta a relação filho-pai

## 4.3 Simulação Física

Para este trabalho a simulação desempenhou um papel essencial nos testes e na avaliação das técnicas utilizadas. Para tal foi utilizado o simulador físico 3D Gazebo<sup>5</sup> (Koenig & Howard, 2004), versão 1.5, que é disponibilizado como ferramenta na plataforma ROS. Este simulador tem sido adotado por demais grupos de pesquisa que utilizam ROS (Meyer *et al.*, 2012) e também, foi o simulador adotado no DRC<sup>6</sup> (*DARPA Robotics Challenge*). A modelagem do sistema em um ambiente virtual é um passo fundamental para executar experimentos em um simulador. Os experimentos em simulação podem ser replicados e repetidos mais facilmente, desta forma, uma maior quantidade de dados pode ser extraída para análises e verificação do comportamento dos algoritmos. A maioria dos algoritmos utilizados no contexto da robótica móvel autônoma requerem uma parametrização e um refino de diversos parâmetros para se adequarem às necessidades da aplicação. Este refino está diretamente associado aos comportamentos desejados que devem ser testados. Com o uso de simulação estes parâmetros e seus efeitos podem ser melhor observados antes de se levar o experimento para o ambiente real.

Para a plataforma CaRINA I foram descritos os modelos de simulação, do veículo e de alguns cenários de testes (*Figura 4.6* e *Figura 4.7*). Para a modelagem em ambiente simulado foi feito o levantamento com o veículo real dos parâmetros dos sensores - câmera estéreo, IMU, GPS, *encoders* - que foram utilizados para criar os modelos destes sensores no simulador. Para

<sup>5</sup>Simulador Gazebo - <http://www.gazebosim.org>

<sup>6</sup>DRC - <http://www.theroboticschallenge.org>

a modelagem de robôs na plataforma ROS é utilizada uma linguagem para a descrição chamada URDF (*Unified Robot Description Format*). O URDF é uma linguagem de descrição em XML onde basicamente são informados os componentes, sensores e relações entre as partes do robô, cada qual tendo um sistema de referência de coordenadas próprio. Estas relações podem ser móveis ou fixas, determinando assim transformações entre cada sistema de referência. Toda informação sensorial e de controle está associada a origem do seu sistema de referência. A descrição em URDF tem uma relação direta ao sistema TF do ROS, sendo assim a própria descrição do modelo do robô em URDF define as relações de transformação dos sistemas de referência de coordenadas.



**Figura 4.6:** CaRINA I - modelo virtual



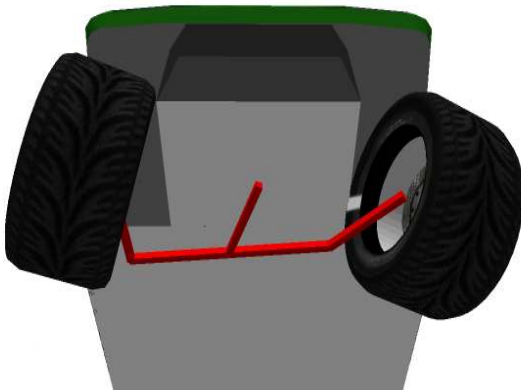
**Figura 4.7:** CaRINA I - cenário teste de ambiente não estruturado

A biblioteca ODE<sup>7</sup> (*Open Dynamics Engine*) utilizada pelo simulador Gazebo para calcular a simulação física utiliza o conceito de ligações (*links*) e juntas (*joints*) para descrever as estruturas do corpo físico sendo simulado. Essa biblioteca apenas executa simulação física de corpos rígidos. Uma junta descreve a relação entre duas ligações e tanto junta como ligação podem ter propriedades físicas associadas, como por exemplo coeficiente de fricção e massa. Para descrever as relações entre ligações e juntas de um corpo físico, uma representação em grafo é mais adequada pois permite representar laços, por exemplo. A descrição em URDF é limitada neste sentido. O simulador Gazebo tem sua própria linguagem de descrição para modelar os elementos da simulação chamada SDF (*Simulation Description Format*). A SDF é de certa forma compatível com a URDF, onde é possível descrever as características específicas da SDF a partir de extensões na URDF. Desta forma, uma única descrição do modelo do robô pode ser compartilhada entre o ROS e o Gazebo. Nestas extensões, principalmente podem ser definidos sensores simulados ligados a um sistema de referência e atuadores. Também é possível

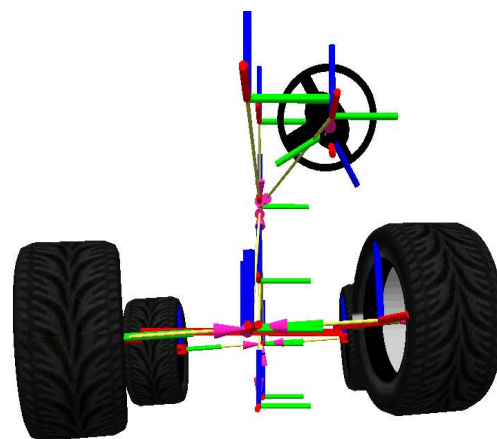
<sup>7</sup>ODE - <http://www.ode.org>

descrever algumas relações mecânicas entre ligações e juntas que são limitadas pela estrutura em árvore da URDF (TF). No modelo desenvolvido para o veículo CaRINA I, a modelagem da geometria Ackermann de esterçamento (*Figura 4.8*) é um caso onde a estrutura em árvore não é suficiente para descrever a restrição física, sendo usado neste caso as extensões em SDF.

Outra necessidade na modelagem para simulação foi capturar os efeitos da suspensão do veículo. Para tal, foi também adicionado ao modelo a capacidade de refletir a dinâmica do veículo em terrenos com irregularidades e deflexões. Outro efeito importante de ser capturado em simulação que permite se aproximar do veículo real são as vibrações e inclinações do veículo no terreno, que afetam as imagens capturadas pela câmera, desta forma, foi possível produzir em simulação as condições adversas que afetam a percepção do sistema. Também, estes efeitos afetam diretamente as leituras da unidade inercial (IMU) simulada, condizendo mais próximo da realidade principalmente em curvas, acelerações e frenagens. Na *Figura 4.9* pode ser visualizada a configuração dos diversos sistemas de referência de coordenadas modelados do veículo em relação à suspensão em um terreno irregular.



**Figura 4.8:** Geometria Ackermann de esterçamento modelada em simulação



**Figura 4.9:** Efeitos da modelagem da suspensão no ambiente simulado

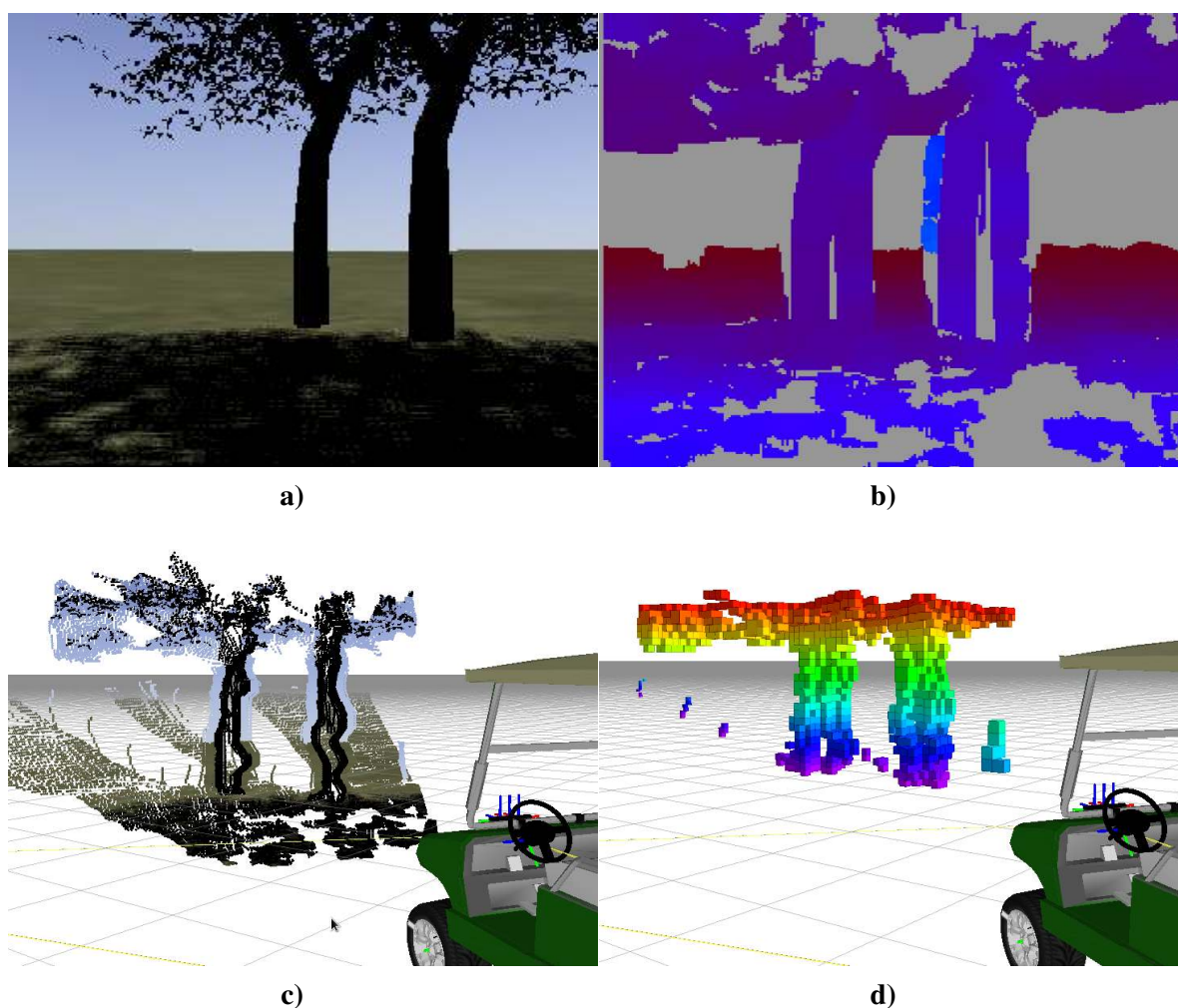
## 4.4 Percepção Visual

O sistema de visão consiste primeiramente no cálculo das disparidades dos pixels entre as imagens das duas câmeras. É aplicado o algoritmo de *Block Matching* (Faugeras *et al.*, 1993) que consiste numa busca local pela correspondência entre os pixels dentro de uma janela a partir da soma das diferenças absolutas (SAD). Foi utilizada a implementação existente na biblioteca OpenCV<sup>8</sup>. O algoritmo escolhido tem um custo computacional comparativamente baixo

<sup>8</sup>OpenCV - <http://opencv.org>

(Scharstein & Szeliski, 2002), apresentado uma qualidade suficiente para o sistema de percepção proposto. Baseado no mapa de disparidade e nos parâmetros de calibração, as coordenadas 3D de cada pixel são computadas e gerada uma nuvem de pontos. Para a manipulação e filtragem da nuvem de pontos foram utilizados algoritmos implementados na biblioteca PCL (*Point Cloud Library*) (Rusu & Cousins, 2011).

Após o mapa de disparidade ser computado pelo método estéreo, as coordenadas tridimensionais para cada ponto que foi calculada a disparidade são estimadas baseado nos parâmetros da câmera. Este passo resulta em uma nuvem de pontos tridimensionais representando a cena observada pelas câmeras. Para a detecção de obstáculo são utilizadas apenas as informações geométricas dos pontos, ou seja, as coordenadas espaciais. As informações fotométricas, como a cor, não são utilizadas. A *Figura 4.10* apresenta de forma geral os passos do sistema de percepção visual.



**Figura 4.10:** Sistema de visão: **a)** imagem (apenas a esquerda), **b)** mapa de disparidade calculado, **c)** nuvem de pontos, **d)** Marcação da ocupação espacial em um OctoMap

O modelo sensorial adotado para a câmera também é baseado em raios a partir de uma origem assim como nos sensores laser. Câmeras não possuem explicitamente um intervalo máximo de distância para a geração dos raios e nem todos os pontos das imagens são pareados para se calcular a disparidade, levando assim, a distâncias desconhecidas em regiões do mapa de disparidade.

Foi adotada a distância de 45 metros para definir como distância máxima, tanto para pontos desconhecidos como aqueles que ultrapassam a distância máxima. São traçados então raios para cada ponto tridimensional calculado na nuvem de pontos tendo como origem o centro da câmera. É marcado então como espaço ocupado o ponto final do raio, todos os demais pontos até a origem são marcados como livre. Estes raios são gerados pelo método *ray-casting*, que na implementação atual retorna a lista dos *voxels* da *octree* que o raio projetado atravessa. Desta forma, é possível atualizar as probabilidades de ocupação do espaço no OctoMap.

## 4.5 Arquitetura do Sistema de Navegação

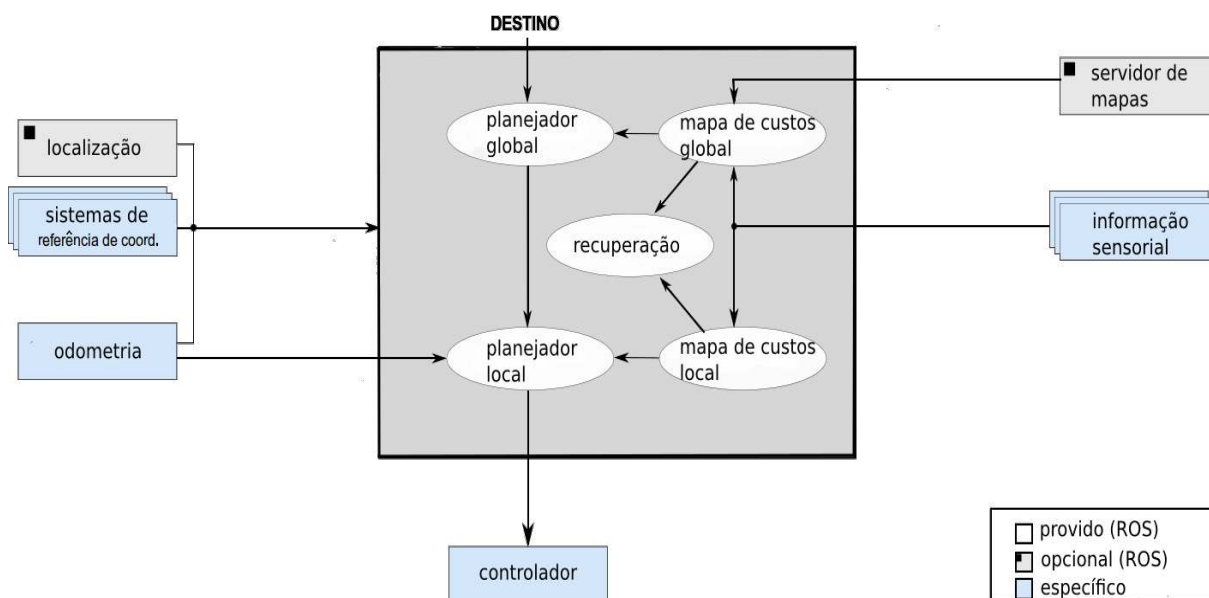
Para o desenvolvimento do sistema e execução no ambiente real também foi utilizada a plataforma ROS (Quigley *et al.*, 2009), pois ela foi projetada tendo como um dos propósitos a padronização de uma interface para dispositivos reais e simulados. Isto permite o desenvolvimentos dos algoritmos de forma independente da fonte dos dados.

A arquitetura se baseia em um comportamento deliberativo em relação ao mapeamento global e um comportamento reativo/deliberativo associado ao mapeamento local. O serviço de mapa fornece informações de ocupação do espaço. O usual é classificar o espaço como livre, ocupado e desconhecido. Estes atributos são utilizados no cálculo dos custos utilizado pelo algoritmo de geração de trajetória. Neste projeto foi adotada uma abordagem probabilística, então este mapa de ocupação representa as probabilidades de um espaço estar obstruído por um obstáculo.

Estas probabilidades são convertidas em custos que são mantidos pelos mapas de custos e atualizados de acordo com as informações sensoriais que são constantemente analisadas. Os mapas de custos são obtidos a partir da detecção de obstáculos realizada com o sistema de visão. O sistema de visão estéreo obtém uma *point cloud* representando os obstáculos, onde esta *point cloud* é tratada e salva em uma estrutura de dados representando o mapa local de ocupação probabilístico (OctoMap), usado para permitir o planejamento da trajetória e o desvio de obstáculos durante a navegação.

A arquitetura do sistema de navegação é baseada no modelo geral do ROS conforme a *Figura 4.11*. Nesta arquitetura os componentes específicos dizem respeito ao veículo e sensores

utilizados, que neste trabalho foram implementados para a plataforma CaRINA I. Já os componentes providos pela plataforma ROS requerem as devidas parametrizações e a escolha dos algoritmos adequados de acordo com as necessidades das tarefas a serem executadas. O *Capítulo 5* descreve o desenvolvimento e customização desta arquitetura.



**Figura 4.11:** Arquitetura geral do sistema de navegação - (adaptado de: [www.ros.org](http://www.ros.org))

## 4.6 Considerações Finais

Neste capítulo foram apresentadas as principais ferramentas e materiais e métodos utilizados no desenvolvimento do projeto. Foi descrita a arquitetura geral do sistema de navegação e o processo utilizado como fonte de informação para se obter a percepção do ambiente. Foi enfatizado o uso de simulação através de um simulador físico e da modelagem do veículo e dos sensores semelhante aos equipamentos reais. A simulação foi utilizada como principal ferramenta de testes e para replicação e repetição de experimentos sem comprometer os equipamentos e produzir um número maior de informações para a análise dos algoritmos. O *Capítulo 5*, a seguir, descreve o detalhamento dos componentes e métodos aplicados e desenvolvidos neste trabalho.

No final do *Capítulo 7* é apresentada uma relação de vídeos (*Tabela 6.5*) com as etapas do desenvolvimento do sistema e resultados de execuções em cenários diversos.





---

# Desenvolvimento

---

A seguir, são apresentados os mecanismos utilizados para criar o mapa de navegação e sua aplicação para ambientes não estruturados em um veículo autônomo. A abordagem se baseia em visão estéreo como principal fonte de informação sensorial para perceber o ambiente.

Por se tratar de uma fonte ruidosa de informação devido às transformações necessárias para reconstruir a informação tridimensional a partir de imagens bidimensionais, a construção do mapa se baseia em uma abordagem probabilística, onde o espaço é representado pelo grau de certeza de estar ocupado e de ser não navegável. Esta informação é convertida diretamente para custos, que são utilizados para o planejamento da trajetória utilizando um algoritmo de busca.

O controle da navegação leva em consideração as características cinemáticas do veículo e mantém o mesmo em curso na rota planejada.

## 5.1 Odometria e Localização

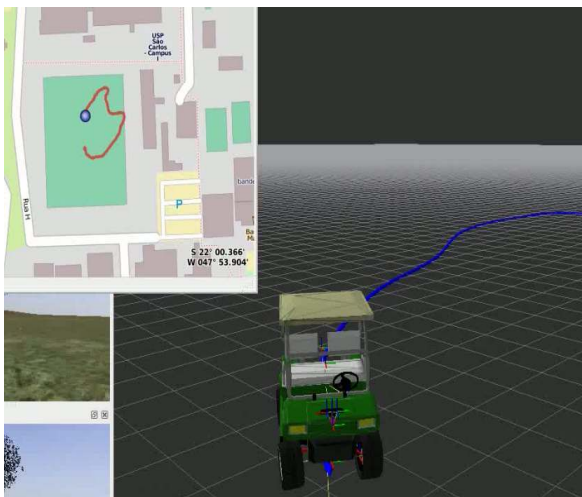
Uma das necessidades básicas para a navegação autônoma é o conhecimento da localização do veículo. Para esta localização a odometria do veículo tem papel importante em informar o deslocamento e pose do veículo durante o seu movimento.

A abordagem para a localização do veículo neste projeto é através do uso do GPS, onde também é possível obter os dados de odometria a partir deste sensor. Um dos problemas associados ao GPS é a localização mais precisa em pequenos deslocamentos como manobras de

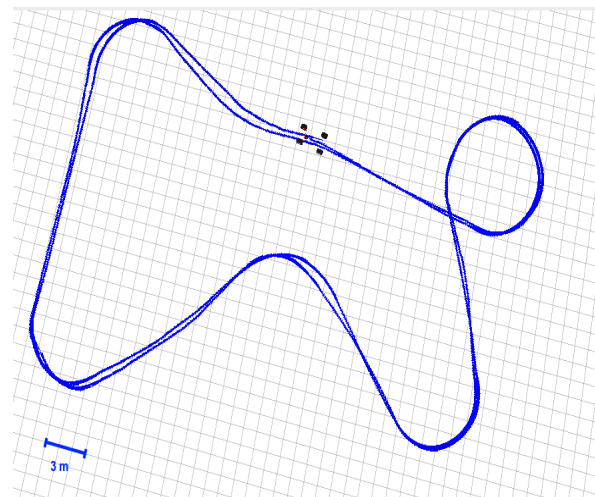
desvio de obstáculos. Já uma odometria por *dead-reckoning* baseada em *encoder* nas rodas do veículo permite obter uma odometria mais precisa nestas condições, porém diverge rapidamente em maiores distâncias.

A abordagem utilizada foi a integração das duas odometrias, pelo *encoder* e pelo GPS) através de um filtro estendido de Kalman (EKF) (Maybeck, 1990) que estima e atualiza a odometria. O GPS está sendo simulado em ambiente virtual assim como o *encoder*, sendo que a odometria e localização nos ambientes real e simulado são equivalentes. A *Figura 5.1* apresenta um exemplo da simulação de GPS e da localização do veículo durante a navegação.

Foram efetuados experimentos com a odometria baseada em *encoder* na plataforma CARINA I que levaram à necessidade de melhorias neste módulo do veículo. Foi desenvolvido um novo módulo de odometria baseado no *encoder* da roda e da barra da direção integrado com os dados da unidade de medida inercial (IMU) que está sendo utilizada no projeto. O novo módulo também tem ajustes no cálculo da geometria Ackermann levando em consideração as quatro rodas do veículo resultando em uma melhor estimativa da odometria conforme descrito em Lee *et al.* (2010). Na *Figura 5.2* é possível visualizar esta odometria por *dead-reckoning* em um percurso com curvas e terreno irregular em simulação. Foram efetuadas duas voltas contornando obstáculos terminando no local de partida, tendo uma acurácia suficiente para a localização do veículo.



**Figura 5.1:** Odometria e localização por GPS



**Figura 5.2:** Odometria e localização por *dead-reckoning*

De acordo com a arquitetura descrita no *Capítulo 4*, o sistema de navegação é dependente da informação sensorial para fornecer dados aos mapas de custos. A extração da informação sensorial baseada em visão é composta por etapas que serão descritas nas próximas seções.

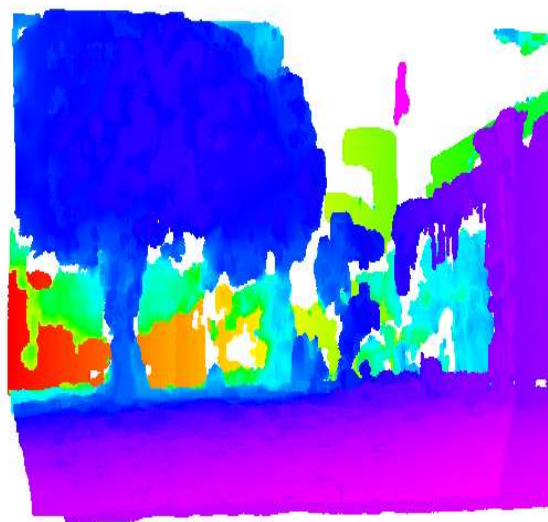
## 5.2 Cálculo da Disparidade

Nesta etapa é efetuado o cálculo da disparidade entre os pixels das imagens relacionadas. A principal dificuldade deste cálculo é fazer a correspondência entre cada pixel, isto é, verificar qual pixel em cada imagem corresponde ao mesmo ponto na cena. Algoritmos que fazem esta busca são de forma geral classificados como busca local, semiglobal e global, onde o custo computacional tende a ser maior nos métodos globais (Scharstein & Szeliski, 2002). Foram testados dois métodos, um local e um semiglobal.

O método local *Block Matching* tem se demonstrado suficiente para gerar a disparidade com um custo computacional que permite aplicação em tempo real. Na *Figura 5.3* é apresentada uma imagem de um cenário virtual, a *Figura 5.4* representa o mapa de disparidade para esta cena, onde as cores que tendem para o vermelho tem uma disparidade menor e representam o que está mais distante na cena, já as cores que tendem para o violeta representam o que está mais próximo na cena. As regiões em branco são partes da imagem onde não foi possível encontrar uma correspondência, geralmente ocasionado quando há muita similaridade na textura do ponto e seus vizinhos, como no caso do céu na imagem.



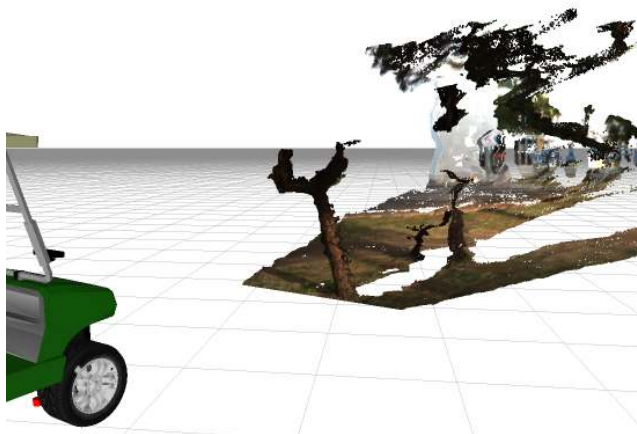
**Figura 5.3:** Imagem obtida pela câmera estéreo (somente esquerda)



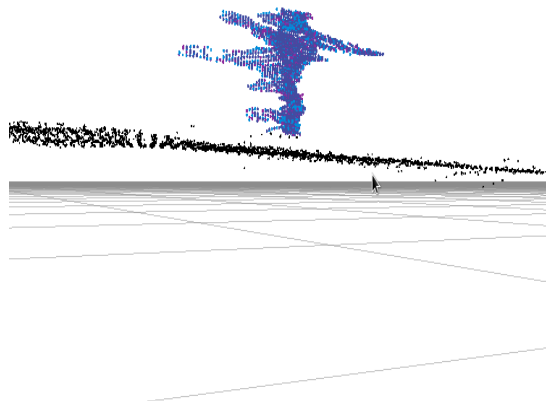
**Figura 5.4:** Cálculo da disparidade entre as imagens (esquerda / direita).

## 5.3 Tratamento da Nuvem de Pontos

Uma nuvem de pontos é construída a partir das disparidades calculadas no passo anterior. Enquanto a disparidade é apenas uma distância relativa em pixels de cada ponto correlacionado, a nuvem de pontos fornece a informação tridimensional métrica da posição do ponto no espaço. Porém, a nuvem de pontos é uma representação não estruturada dos dados da cena, sendo então necessário uma forma de armazenar e fazer buscas nestes dados de forma estruturada (*Figura 5.5*).



**Figura 5.5:** Nuvem de pontos



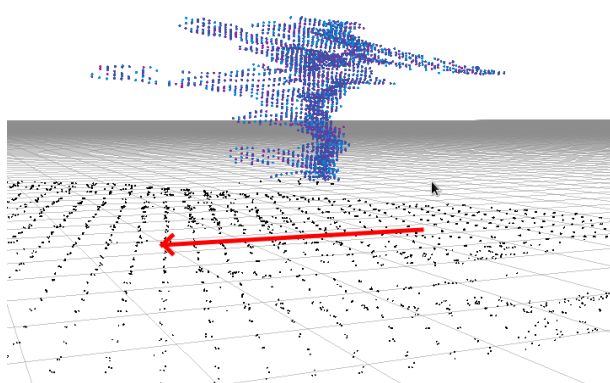
**Figura 5.6:** Em preto, os pontos segmentados como chão

### 5.3.1 Detecção do chão

A primeira etapa de processamento da nuvem de pontos consiste em determinar os pontos que pertencem ao chão. Os pontos compreendidos como o chão são excluídos da análise de obstáculos. Para segmentar o plano do chão é utilizado o método de RANSAC (RANDOM Sample Consensus) (Fischler & Bolles, 1981). O RANSAC busca pelo maior número de pontos que se ajustam a um modelo dentro de uma tolerância. No caso do chão é utilizada a equação do plano como modelo. Na *Figura 5.6* é possível ver em preto os pontos extraídos pelo RANSAC que se ajustam ao modelo e são marcados como o chão.

### 5.3.2 Ruído

As nuvens de pontos geradas por câmeras estéreo tendem a apresentar um tipo de ruído característico ao longo do eixo da projeção perspectiva. Este ruído causa um espalhamento ondular no eixo  $Z$  (profundidade) dos pontos que representam uma determinada estrutura. Este efeito é mais presente em estruturas distantes da câmera, visto que a disparidade entre pontos nos objetos distantes é cada vez menor a medida que se afastam da câmera. Erros numéricos, erros de casamento entre pontos pelo método estéreo aplicado e baixa resolução das imagens produzem esse tipo de ruído. Na *Figura 5.7* é possível observar o ruído nos pontos em azul que representam a estrutura de uma árvore (*Figura 5.8*).



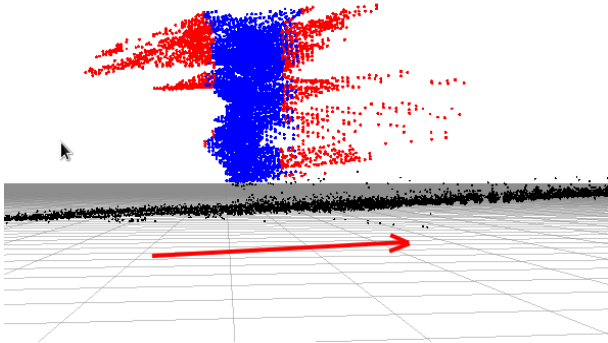
**Figura 5.7:** Ruído ondular (em azul) na nuvem de pontos gerada pela câmera estéreo



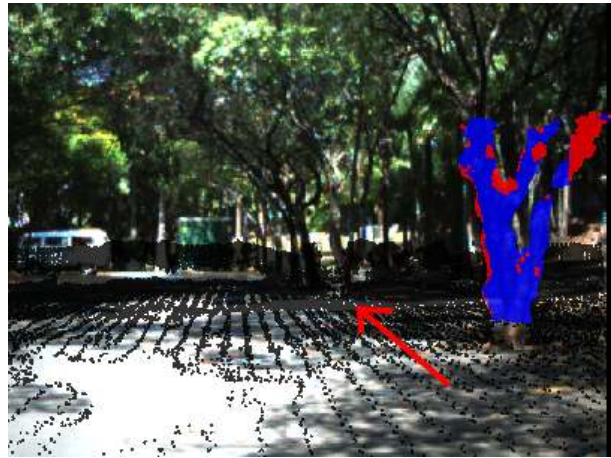
**Figura 5.8:** Estrutura real (árvore) e a projeção da nuvem de pontos

### 5.3.3 Filtro

Para tratar este ruído foi desenvolvido um filtro que normaliza as distâncias em relação ao eixo  $Z$ . Esta normalização consiste em obter a média e o desvio padrão das distâncias dos pontos de cada estrutura agrupada e ajustar as distâncias dentro do intervalo de um desvio padrão. O efeito do filtro pode ser visto na *Figura 5.9* onde os pontos em vermelho estão fora do intervalo e os pontos em azul são os pontos ajustados pelo filtro. Na *Figura 5.10* pode ser vista a projeção dos pontos na imagem real.



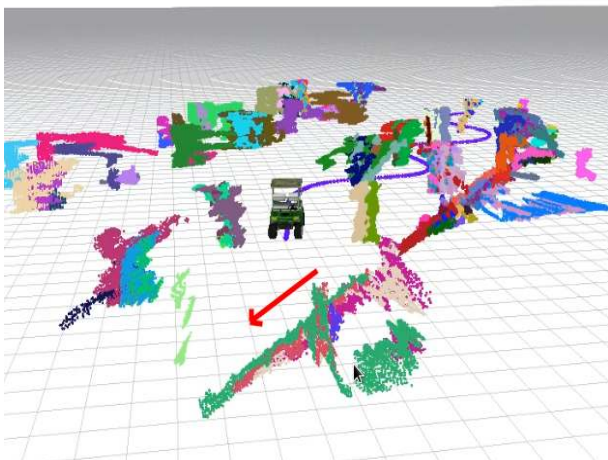
**Figura 5.9:** Em azul os pontos ajustados pelo filtro, em vermelho os pontos fora do intervalo



**Figura 5.10:** Projeção dos pontos na imagem real

### 5.3.4 Segmentação da cena

A nuvem de pontos gerada pelo mapa de disparidade contém os pontos de todas as estruturas da cena, para se aplicar o filtro descrito anteriormente é necessário isolar independentemente cada estrutura. Devido a esta necessidade é aplicado um método de segmentação baseado no algoritmo apresentado em Rabbani *et al.* (2006), que separa as estruturas a partir de uma busca por vizinhos em regiões com densidades maiores de pontos, separando assim cada estrutura em conjuntos de pontos (*clusters*). Na *Figura 5.11* pode se ver alguns elementos da cena separados pela segmentação. A *Figura 5.12* apresenta uma foto real da cena.



**Figura 5.11:** Agrupamentos marcados com diferentes cores (algumas cores se repetem)

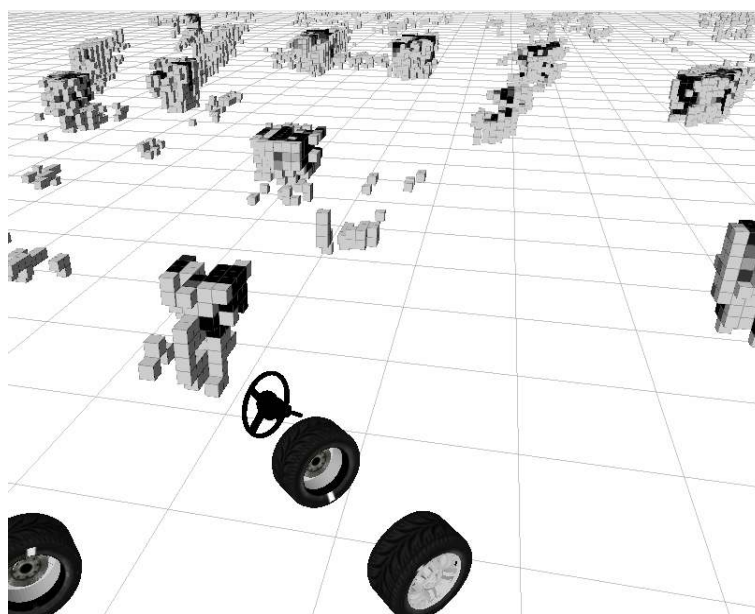


**Figura 5.12:** Foto do cenário real

## 5.4 Modelo Probabilístico de Ocupação

Tendo os segmentos isolados da cena e a devida filtragem dos dados esta informação é armazenada como espaço ocupado no ambiente. Porém, esta informação não é binária, ou seja, ocupado ou livre. É utilizada uma abordagem probabilística da ocupação do espaço, desta forma, a toda informação obtida através da visão estéreo é atribuída uma probabilidade. Para armazenar e atualizar constantemente estas probabilidades é utilizada uma *octree* (Hornung *et al.*, 2013).

A *octree* divide os espaço em volumes cúbicos, discretizando o espaço. É aplicado uma discretização do espaço de 10 centímetros. Esta discretização define o tamanho máximo que a estrutura de dados irá armazenar e a capacidade de mapeamento. Na *Figura 5.13* pode ser visualizado o espaço discretizado e as probabilidades de ocupação mapeadas em tons de cinza, onde o escuro representa maior probabilidade e o claro menores.



**Figura 5.13:** Ocupação espacial probabilística em tons de cinza. Tons escuros representam maior probabilidade, tons claros menores probabilidades

Uma necessidade comum em um sistema de navegação onde há o planejamento de trajetória sobre um mapa é o tratamento das áreas desconhecidas ou não mapeadas. Uma abordagem conservadora é considerar tais espaços não explorados como proibitivos para o veículo, ou pelo menos, a serem evitados pelo planejador. Desta forma, é atribuído a estas regiões do mapa um custo infinito a fim de inibir o planejador de traçar uma rota que atravessasse esses espaços. Neste projeto foi assumida uma abordagem otimista no sentido de permitir o veículo explorar

livremente o espaço. Sendo assim, foi atribuído um custo intermediário, que em termos de probabilidade de ocupação representa 50% de certeza para o espaço desconhecido. A atualização da OctoMap é feita somando-se o  $\log Odds$  da probabilidade conforme descrito no *Capítulo 2, Subseção 2.3.1*.

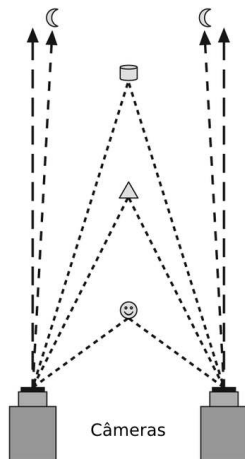
### 5.4.1 Modelo Sensorial

A atualização da OctoMap depende do modelo sensorial adotado. Para a atualização das probabilidades e o preenchimento do espaço com tais certezas, as leituras dos sensores são convertidas em pontos no espaço com alguma informação agregada, além da sua localização tridimensional. O modelo sensorial simples utilizado para nuvens de pontos produzidas por câmeras estéreo segue a mesma abordagem utilizada em sensores do tipo *rangefinder*, como sensores laser. Os pontos lidos são marcados como presença de obstáculo e todos os demais pontos a partir deste até a origem do sensor são marcados como ausência de obstáculo. Estes raios são calculados através do método *ray-casting*, conforme já citado no *Capítulo 2, Subseção 2.3.1*.

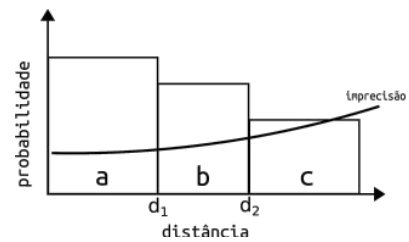
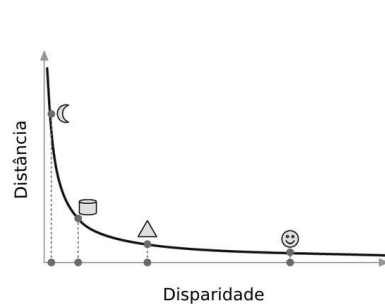
Neste trabalho, este modelo sensorial foi estendido para contemplar certas características próprias das câmeras de vídeo e do método de extração das coordenadas tridimensionais. Um efeito característico é a baixa granularidade da disparidade calculada para objetos distantes em relação a objetos próximos, isto porque a disparidade é inversamente proporcional à distância, também há uma não linearidade nesta razão. A figura *Figura 5.14* demonstra estas relação entre distância e disparidade. São definidos 3 intervalos em relação à distância, o intervalo próximo (*a*), o intervalo médio (*b*) e o intervalo distante (*c*). Estes intervalos são definidos por 2 distâncias ( $d_1$  e  $d_2$ ), conforme *Figura 5.15*. Para cada intervalo é atribuído uma probabilidade que é utilizada para atualizar a OctoMap. Desta forma, é contemplada a imprecisão da disparidade e o erro provocado no cálculo da distância dos pontos através da redução da probabilidade a medida que pontos são detectados mais distantes da câmera.

O alcance máximo definido para o sensor foi de 45 metros e a probabilidade utilizada para marcar as leituras do espaço livre foi de 48%, equivalente em  $\log Odds$  a aproximadamente  $-0,08$ . A probabilidade do espaço para ser considerada obstáculo foi definida em 85%, equivalente em  $\log Odds$  a aproximadamente  $1,735$ . As distâncias  $d_1$  e  $d_2$  foram definidas em 7 metros e 12 metros, respectivamente. As probabilidades utilizadas são apresentadas na *Tabela 5.1*, onde consta também o valor equivalente em  $\log Odds$  e a quantidade de leituras necessárias para atingir os 85% de probabilidade de ocupação. Foram utilizadas probabilidades diferentes para o ambiente simulado e o ambiente real pois existem diferenças na qualidade da textura das imagens e na quantidade de quadros por segundo (FPS) em cada configuração.





**Figura 5.14:** Distância *versus* disparidade  
- Adaptado de (Bradski & Kaehler, 2008)



**Figura 5.15:** Probabilidade diferenciada para cada intervalo de distância para contemplar a imprecisão do cálculo da distância pela disparidade

**Tabela 5.1:** Probabilidades de cada intervalo de distância no modelo sensorial

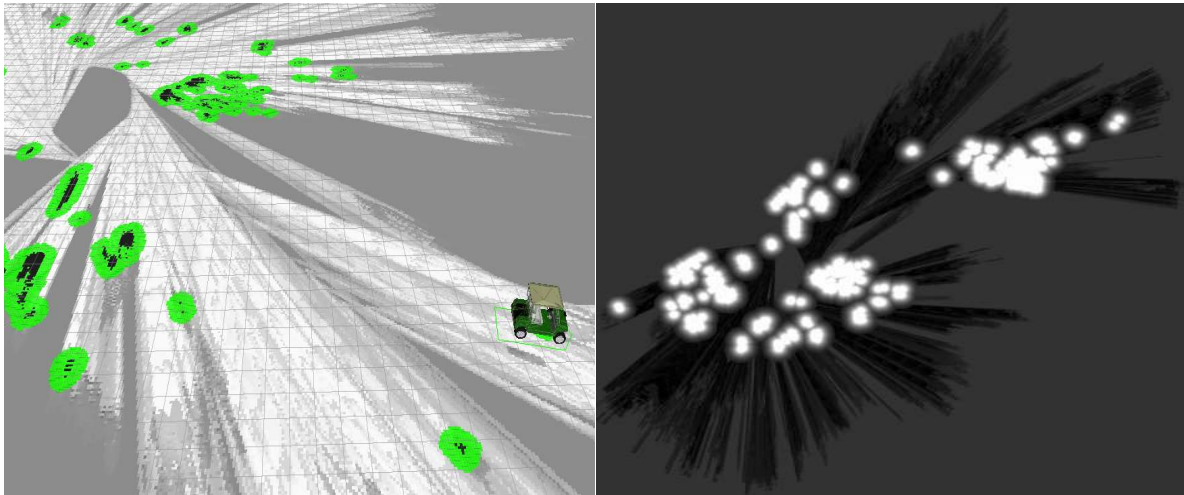
Intervalo	em %	em $\log Odds$	Leituras necessárias para atingir 85%
<b>Ambiente simulado</b>			
a	70,425%	0,8676	2 (1,99)
b	64,1%	0,5797	3 (2,99)
c	58,6%	0,3475	5 (4,99)
<b>Ambiente real</b>			
a	56,5%	0,2481	7 (6,99)
b	54,5%	0,1736	10 (9,99)
c	52,41%	0,0965	18 (17,98)

Tendo como exemplo a necessidade de 6 leituras para marcar um obstáculo, o sistema de percepção operando a uma frequência de 4 Hz levaria 1,5 segundos para marcar o obstáculo. Se esta frequência fosse alterada para 12 Hz, seriam necessários 0,5 segundos. Com as taxas de frequência que o sistema tem capacidade de operar é possível ajustar a velocidade máxima que o veículo pode atingir dada a necessidade de se detectar um obstáculo a uma determinada distância. A uma velocidade de 3 m/s e obstáculos sendo marcados a 0,5 segundos é possível perceber obstáculos em um deslocamento de 1,5 metros.

Tendo em vista o modelo cinemático do veículo é possível determinar em até que distância mínima um obstáculo pode ser detectado para que o veículo execute a manobra de desvio ou frenagem com segurança. Portanto, estas probabilidades podem ser ajustadas de acordo com outros parâmetros do sistema, como a frequência que são capturadas as imagens e calculada a disparidade e a velocidade máxima do veículo tal que um obstáculo seja detectado a uma distância mínima desejada.

## 5.5 Sistema de Navegação

O sistema de navegação integra a percepção e a atuação do veículo. A partir da localização e do mapa de navegabilidade convertido em mapa de custos o planejador de trajetória calcula o plano de navegação deliberativa. Este plano é seguido pelo módulo de controle do veículo que também é responsável pela navegação reativa.



**Figura 5.16:** Mapa de navegabilidade (2D)

**Figura 5.17:** Mapa de custos em tons de cinza, sendo o branco o maior custo, o preto o menor

### 5.5.1 Mapa de Navegabilidade

A partir dos dados já agrupados em uma *octree* é possível analisar de forma estruturada a ocupação do espaço, assim, gerando um mapa local de navegabilidade baseado na informação de ocupação ou não do espaço, conforme *Figura 5.16*. O mapa de navegabilidade também reflete a probabilidade de ocupação do espaço, sendo desta forma um mapa probabilístico de navegabilidade. Como o planejamento de trajetória é efetuado sobre um mapa bidimensional, a ocupação espacial é projetada no plano utilizando o valor máximo encontrado na altura (dimensão  $Z$ ).

### 5.5.2 Mapa de Custos

Um mapa de navegação basicamente contém as informações de espaço ocupado não navegável, espaço livre navegável e o espaço cuja navegabilidade é desconhecida. Os algoritmos de planejamento de navegação em geral se baseiam na informação de mapas de custos e buscam

encontrar o caminho de menor custo. Um mapa de custos pode ser obtido diretamente do mapa de navegabilidade, atribuindo-se valores de custos aos estados, navegável, não navegável e desconhecido. O mapa de navegabilidade desenvolvido neste trabalho contém a probabilidade de ocupação do espaço, podendo ser ajustado às classes navegável/não navegável por intervalos de percentuais. Porém, para transformar em mapa de custos as probabilidades podem ser utilizadas diretamente.

Foi considerada neste trabalho a probabilidade de ocupação de 85% como não navegável. Esta informação é importante pois é necessário garantir que o planejador não projete rotas tanto nestes pontos como em uma região de segurança das dimensões do veículo. Para garantir esta segurança os pontos no mapa de custo considerados não navegáveis geram uma região de inflação dos custos no seus vizinhos. Esta inflação é uma expansão dos custos em torno dos obstáculos levando em consideração a dimensão lateral do veículo, afim de evitar que haja um planejamento de trajetória rente ao obstáculo. Na *Figura 5.17* é possível ver um mapa de custos gerado pelo sistema. É possível observar zonas circulares onde os custos foram inflados. Além disto, neste mapa os custos são mapeados em tons de cinza onde o branco é o custo máximo (255) e o preto o mínimo (0).

### 5.5.3 Planejador Global

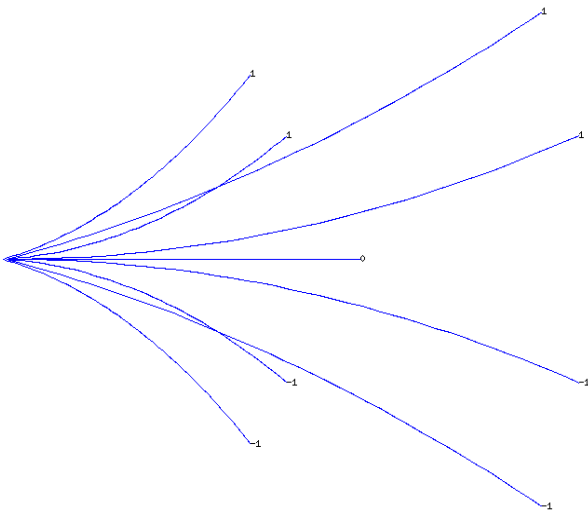
Neste trabalho foi utilizado um planejador de trajetórias baseado em malha de estados que gera uma trajetória de acordo com o modelo cinemático do veículo. No caso, a trajetória planejada leva em consideração as manobras que o veículo consegue executar, assim como a velocidade. Na *Figura 5.20* é possível observar uma tomada do sistema de planejamento em execução. As áreas em rosa são os obstáculos, em azul os custos inflados, a seta em vermelho o destino que foi dado ao veículo, o traçado em vermelho representa a trajetória planejada.

Para tal, foram parametrizadas 9 trajetórias básicas geradas a partir de comandos de esterçamento e velocidade levando em consideração o modelo cinemático do veículo. Os ângulos de esterçamento utilizados para gerar as primitivas foram  $-32$ ;  $-22$ ;  $-15$ ;  $0$ ;  $15$ ;  $22$  e  $32$  graus, conforme a *Figura 5.18*. São aplicadas 16 rotações de  $22,5$  graus nestas amostras para que se possa obter uma expansão em  $360$  graus, conforme *Figura 5.19*. Desta forma, são geradas 144 primitivas básicas de movimento do veículo que são expandidas em uma malha de estados possíveis do veículo se posicionar. Esta malha é projetada sobre o mapa de custos gerando um grafo com custos associados.

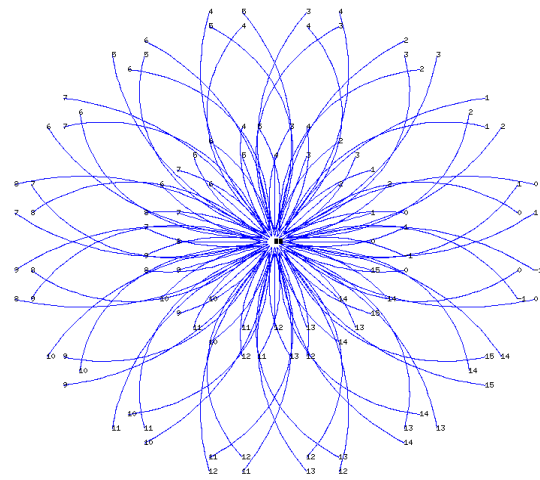
A trajetória é escolhida por base em uma busca por um caminho no grafo de menor custo. Foi utilizada a biblioteca SBPL<sup>1</sup> também disponibilizada no ROS. O algoritmo de busca utili-

<sup>1</sup><http://http://www.sbpl.net/>

zado foi o AD\* (*Anytime Dynamic A-Star*). O AD\*, descrito em (Likhachev *et al.*, 2008), tem como característica a capacidade de incrementalmente executar o planejamento do caminho de menor custo, gerando um resultado intermediário quase ótimo dentro de um tempo limite até produzir o resultado final. O algoritmo é capaz de fazer o replanejamento aproveitando o resultado anterior em caso de atualização dos custos. Em (Magalhães *et al.*, 2013) foi utilizada essa abordagem para a tarefa do estacionamento autônomo.



**Figura 5.18:** Primitivas de trajetórias básicas levando em consideração o modelo cinemático do veículo



**Figura 5.19:** Rotação das primitivas para permitir a expansão em 360 graus

#### 5.5.4 Planejador Local

O planejamento local é responsável por manter o veículo em curso no plano da trajetória global e reagir a situações inesperadas não previstas, como o desvio de obstáculos dinâmicos, responsável pela camada reativa do sistema de navegação conforme visto na *Seção 2.6*.

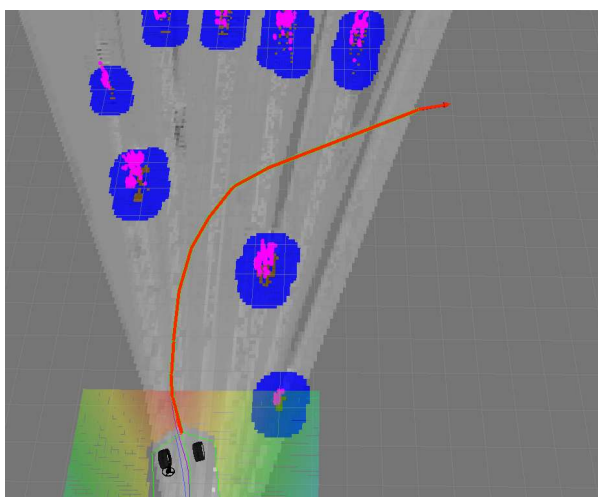
O controle irá calcular a velocidade e o ângulo de esterçamento necessários para o veículo alcançar a próxima posição  $(x; y; \theta)$  determinada pelo planejador local. Foi adotado o algoritmo DWA (*Dynamic Window Approach*) (Fox *et al.*, 1997) para o planejamento local e controle com adaptação para levar em conta o modelo cinemático do veículo respeitando o modelo de esterçamento de Ackermann.

Originalmente o DWA reprojeta semicírculos que representam cada comando de esterçamento e velocidade enquanto que esta adaptação reprojeta a trajetória de acordo com a geometria Ackermann. Os parâmetros de velocidade máxima e limites de esterçamento aplicados foram de 3 m/s e  $-32$  a 32 graus respectivamente. Para o DWA criar as amostras e escolhe o

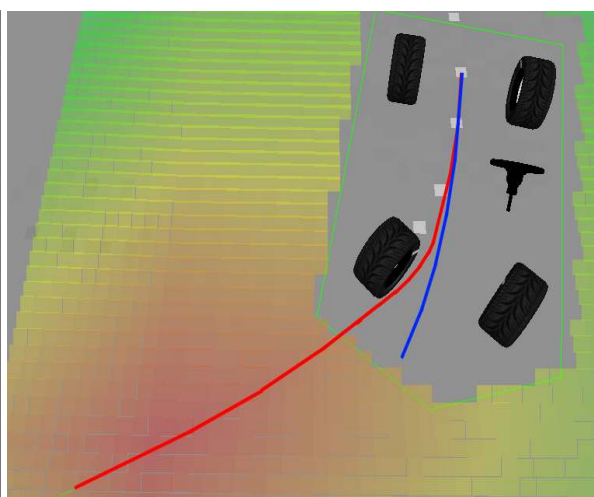
traçado ao qual serão gerados os comandos de controle também são necessárias a discretização dos parâmetros de aceleração e esterçamento.

Estão sendo utilizados 11 níveis de aceleração e o limite de esterçamento é subdividido igualmente em 37 ângulos. Também, uma janela que representa uma área em torno do veículo que será considerada para o planejamento. Esta janela recebe uma cópia dos custos do mapa global centrada na posição do veículo. Está sendo utilizada uma janela quadrada de 8 metros de lado. As trajetórias que terminam fora da janela são descartados, assim como planos que entram em rota de colisão com obstáculos representados pelos custos do mapa de custo. Este mapa de custo local também pode ser atualizado diretamente pelos dados dos sensores do veículo permitindo um comportamento reativo.

Uma função objetivo é utilizada para escolher a trajetória a ser aplicada pelo controle. Esta função objetivo gera um mapa de atração para o local de menor custo conforme pode ser visto na *Figura 5.21*. Nesta imagem, as cores que tendem para o vermelho representam o menor custo, e as cores que tendem para o verde o maior custo. O traçado em vermelho representa o plano global, o traçado em azul representa a trajetória que equivale aos comandos executados pelo controle naquele instante de tempo que melhor posiciona o veículo em relação ao plano global dado o estado do veículo. A função objetivo aplicada utiliza a estratégia apresentada em (Brock & Khatib, 1999), favorecendo o menor custo sobre a trajetória do planejador global. Desta forma, o veículo respeita o traçado obtido pelo planejamento global executado apenas os desvios necessário para evitar colisões.



**Figura 5.20:** O traçado em vermelho representa a trajetória planejada até o destino escolhido desviando dos obstáculos marcados em rosa



**Figura 5.21:** Janela de navegação local. O traçado em vermelho representa a trajetória planejada, em azul, o traçado do controle referente ao esterçamento e velocidade naquele instante

## 5.6 Considerações Finais

A localização é uma necessidade elementar para a construção de mapas métricos de navegabilidade e a possibilidade de planejamento por trajetórias uma vez que a posição do veículo em relação ao mapa é que define os comandos de controle e desvios de obstáculos.

A partir do cálculo da disparidade entre as imagens da câmera estéreo é recuperada a informação tridimensional da cena. A partir de uma nuvem de pontos os obstáculos são detectados e é construído um modelo de ocupação espacial probabilístico que permite a sua atualização contínua. Esta constante atualização permite ao modelo uma robustez em relação ao ruído inerente da informação tridimensional proveniente das nuvens de pontos produzidas a partir de métodos que utilizam câmera estéreo. A partir deste modelo de ocupação é construído um mapa de navegabilidade utilizado por um planejador de trajetórias que conduz o veículo em uma navegação deliberativa.

A abordagem de segmentar a nuvem de pontos para aplicar o filtro desenvolvido que reduz os efeitos do ruído introduzido pelo método de gerar os pontos tridimensionais pela câmera estéreo possibilitou a marcação do espaço ocupado com maior grau de certeza permitindo a detecção de obstáculos. Com o modelo sensorial descrito foi possível ajustar estes graus de certeza. Esta detecção de obstáculos permitiu a construção de um mapa de navegabilidade no qual é efetuado o planejamento de trajetória para a navegação autônoma. Esta trajetória se baseia em primitivas básicas de movimentos válidos para o veículo de acordo com o seu modelo cinemático.

O sistema foi testado em simulação e validado em ambiente real conforme demonstrado no *Capítulo 6*, a seguir.

---

## Resultados

---

Neste trabalho foi necessário tratar diferentes aspectos do sistema de navegação e do veículo para atingir o objetivo de se chegar a um sistema de navegação autônomo completo. A primeira necessidade foi obter um mecanismo de localização pois todo sistema depende desta informação. A localização por GPS foi aplicada como referência global e foi utilizada conjuntamente uma localização a partir de *encoders* e unidade inercial por *dead-reckoning* conforme descrito na *Seção 5.1*.

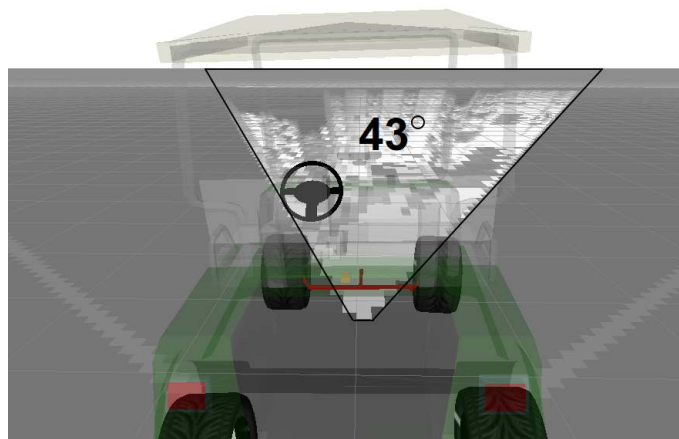
O objetivo não foi de se fazer um mapeamento do local e sim obter um mapa de navegabilidade para executar um planejamento de trajetória, desta forma, a localização mesmo com alguma imprecisão e erros acumulados ao longo do tempo pode ser reinicializada assim como o mapa de navegabilidade a cada manobra completada entre posições intermediárias.

Uma forma de se obter posições intermediárias para traçar trajetórias menores em um deslocamento do veículo de maior extensão é a partir da execução de um algoritmo de expansão como o T-RRT visto na *Seção 2.6*, gerando assim pontos de GPS intermediários. A cada ponto alcançado, o sistema de localização e mapeamento de navegabilidade poderia ser então reinicializado, traçando-se uma nova trajetória até o próximo ponto intermediário. Com esta abordagem, o mapa de navegabilidade somente precisa ser mantido enquanto estiverem sendo executadas as manobras para alcançar cada objetivo.

A única câmera estéreo utilizada limitou o campo de visão do veículo, como pode ser visto na *Figura 6.1*, a abertura angular de 43 graus, reduzindo a visibilidade do veículo e compro-

metendo manobras com curvas fechadas. Porém, foi adotado este equipamento e apenas uma câmera por questões de simplicidade, sendo suficiente para demonstrar a capacidade de geração de um mapa probabilístico de navegabilidade a partir de um sistema de visão estéreo.

A utilização da OctoMap para armazenar e manipular os dados de ocupação, conforme visto na *Subseção 2.3.1*, torna a aplicação independente do sensor utilizado para gerar a nuvem de pontos para marcar os obstáculos e o espaço livre detectados. Desta forma, é possível fazer a fusão dos dados de sensores independentes, sendo apenas necessários a posição do sensor no sistema de referências de coordenadas e o modelo sensorial aplicado para atualizar as probabilidades de ocupação do espaço. Com isso, é possível adotar a abordagem de se utilizar mais câmeras para aumentar o campo de visão do veículo.



**Figura 6.1:** Campo de visão do sistema de percepção

Conforme descrito na *Seção 4.1*, o veículo CaRINA I não possui automatizada a inversão da marcha, o que limita o veículo de executar manobras evasivas e de recuperação para corrigir a trajetória. Esta característica foi mantida no modelo simulado, resultando em alguns casos de testes de o veículo cancelar a execução da trajetória por impossibilidade de se mover para frente. Os principais parâmetros utilizados para modelar o veículo em simulação e para descrever o modelo cinemático constam na *Tabela 6.1*. O posicionamento dos sensores e os principais sistemas de referência de coordenadas utilizados para executar as transformações dos dados em relação a localização no mapa do ambiente e ao próprio veículo estão descritos na *Tabela 6.2*.

A sequência das etapas do sistema desenvolvido foram descritas no *Capítulo 5*. A partir da localização obtida e a segmentação dos obstáculos pelo sistema de percepção utilizando os dados provenientes da câmera estéreo pôde ser construído o mapa de navegabilidade. Na *Figura 6.2* pode ser vista a marcação dos obstáculos pela OctoMap e a probabilidade de ocupação representada pelas cores em tons de cinza, sendo as cores mais escuras as maiores probabilidades de ocupação.



**Tabela 6.1:** Parâmetros do veículo

Parâmetro	Valor
Masa (sem rodas)	350 Kg
Velocidade máxima	10 Km/h
Distância entre eixos	1,64 m
Distância entre rodas traseiras	1,06 m
Distância entre rodas dianteiras	0,96 m
Diâmetros da roda	0,48 m
Largura da roda	0,24 m
Massa da roda	5 Kg
Altura	1,85 m
Esterçamento máximo (limitado a)	32 graus

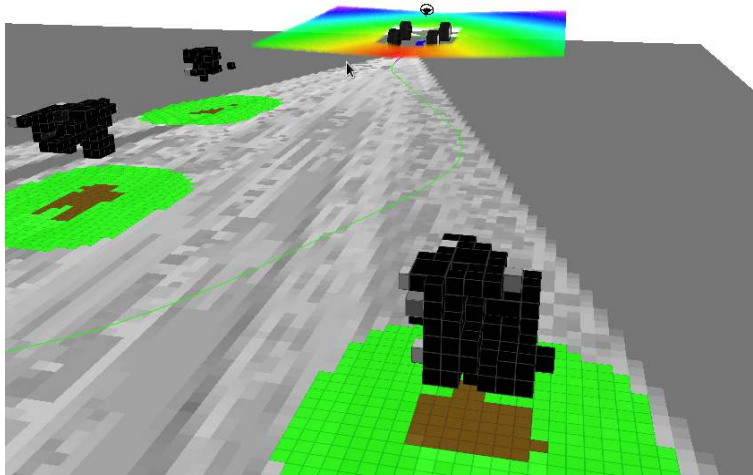
**Tabela 6.2:** Principais sistemas de referência de coordenadas

Referência	Translação relativa ao centro* (x; y; z), em cm	Rotação relativa ao centro* (x; y; z), em graus
centro do veículo*	(0; 0; 0)	(0; 0; 0)
GPS / IMU	(-0,82; 0; 0,97)	(0; 0; 0)
câmera estéreo	(0,97; 0; 0,965)	(-90; 0; -90)
odometria	(-0,82; 0; 0)	(0; 0; 0)

Com o sistema de localização desenvolvido foi possível marcar com sucesso o posicionamento dos obstáculos assim como atualizar as probabilidades de ocupação espacial. No sistema de visualização, foram marcados os obstáculos nas mesmas posições do ambiente simulado para possibilitar a validação, conforme pode ser visto na *Figura 6.3*, a marcação dos obstáculos

pelo sistema de percepção e localização condizem com a posição real dos obstáculos. Nesta figura, em a), pode ser vista a ocupação probabilística espacial representada pela OctoMap e o mapa de navegabilidade projetado no plano do chão, juntamente com a marcação dos obstáculos na posição real. Em b), pode ser visto o mapa de custos relativo ao mapa de navegabilidade juntamente com a marcação dos obstáculos nas posições reais.

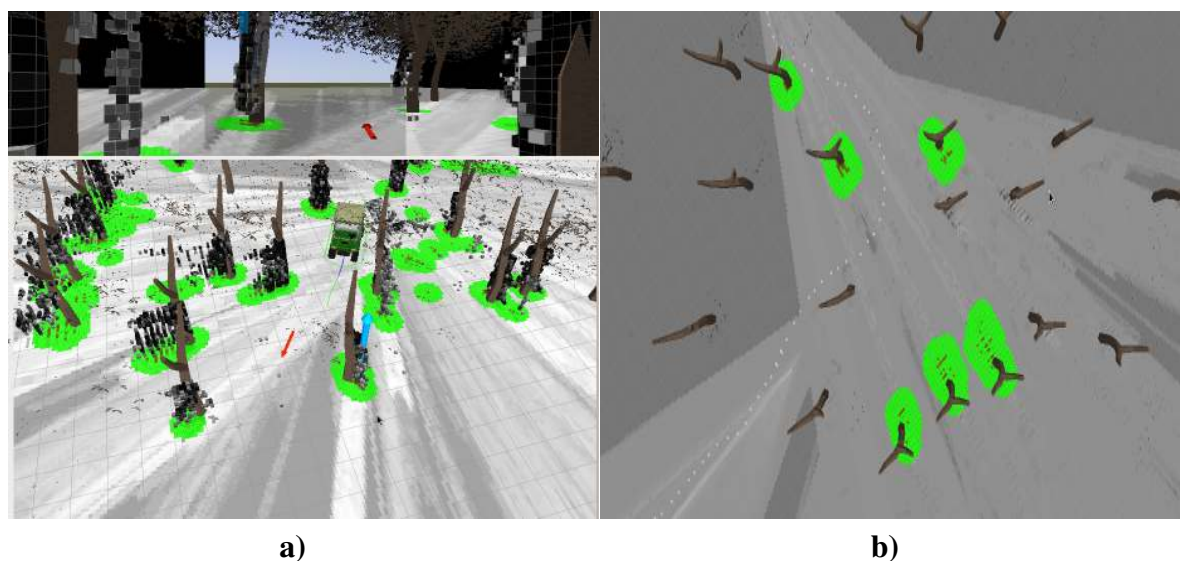


**Figura 6.2:** Ocupação espacial probabilística representada pela OctoMap

## 6.1 Testes em Simulação

A simulação desempenhou um papel importante em todo o desenvolvimento do projeto, permitindo a execução de testes de forma mais versátil e rápida, sem comprometer os equipamentos a possíveis danos. O sistema de percepção em simulação foi ajustado para operar em 5 Hz. O sistema de localização, navegação e controle foi ajustado para operar em 10 Hz.

O subsistema do ROS responsável pelas transformações entre sistemas de referência de coordenadas foi ajustado para operar em 50 Hz. Estas frequências de amostragem foram ajustadas de acordo com a capacidade de processamento dos equipamentos utilizados para as simulações. Foram utilizados dois computadores Intel Core i5, com 4 Gb de memória ligados em rede *gigabit*. Em um equipamento foi executado o simulador e a visualização, sendo o outro dedicado ao sistema de percepção e navegação do veículo. A velocidade máxima do veículo foi ajustada para 1,5 m/s.



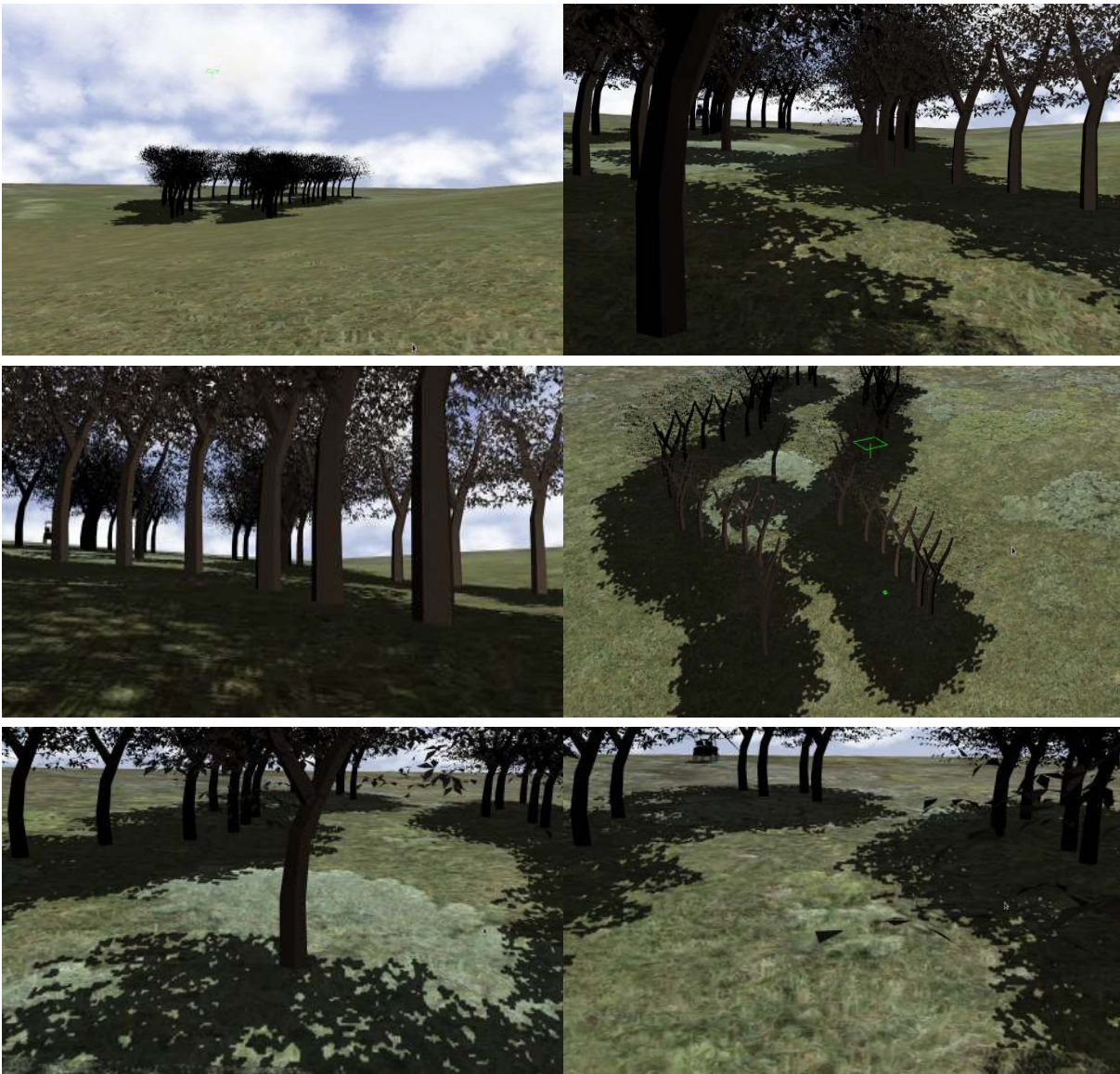
**Figura 6.3:** Visualização dos obstáculos e o comportamento do sistema de percepção e localização, em a) pode ser vista a representação pela OctoMap da ocupação espacial e o mapa de navegabilidade, em b), o mapa de custos. Em verde, são representadas as inflações do custo dos obstáculos detectados

### 6.1.1 Primeiro cenário de teste

Para o primeiro teste em simulação foi modelado um terreno irregular com aclividade e desníveis. Foi criado um percurso cercado por árvores que obrigam o veículo a fazer o desvio de obstáculos para atingir o destino estabelecido. A *Figura 6.4* apresenta algumas imagens deste cenário de teste.

O experimento consistiu em destinar o veículo a uma posição a 55 metros a frente em linha reta. O caminho não é totalmente livre nesta direção levando à necessidade de se planejar uma trajetória que desvie dos obstáculos. Alguns obstáculos foram posicionados propositalmente em oclusão, sendo que a visibilidade dos mesmo para o veículo só se torna presente no decorrer do percurso, forçando o replanejamento da trajetória. Este cenário pode ser considerado semiestruturado uma vez que existem pelo menos dois caminhos viáveis bem delimitados pelos obstáculos.

Porém, não foi adicionado ao sistema nenhum conhecimento prévio em relação a essa estruturação. Não foi inserido nenhum elemento dinâmico na cena, com exceção dos efeitos de sombra simulados e da própria variação da dinâmica do veículo em relação às irregularidades do terreno e das variações dos comandos de controle fazendo com que cada simulação seja independente. Foram efetuadas 24 execuções no mesmo cenário sem alterações na configuração dos obstáculos e na posição de destino. O posicionamento inicial de veículo tem pequenas variações de orientação e posição devida a própria inicialização da simulação ocasionadas pela



**Figura 6.4:** Imagens do cenário em simulação para o experimento de teste

irregularidade do terreno, o que contribui para que cada simulação tenha um componente inicial não determinista.

Na *Tabela 6.3* são exibidas as métricas destas execuções e o resultado final de cada navegação. A *Tabela 6.4* apresenta as imagens da visualização de cada experimento no final da execução. Nestas imagens, em azul está marcada a trajetória executada pelo veículo, em laranja os planejamentos de trajetória calculados durante o percurso. Em verde são os obstáculos detectados.

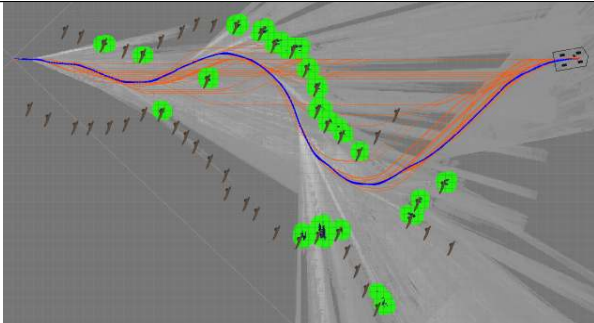
A marcação das posições das árvores na visualização tem o propósito de verificar a consistência do mapeamento em relação ao ambiente. Em alguns casos, o veículo não finalizou o percurso. Nos casos em que o veículo parou, o planejador não conseguiu traçar uma nova trajetória em relação à posição que o veículo se deslocou pois não tem a capacidade de executar uma manobra evasiva em marcha ré. Já nos casos em que houve a colisão, o obstáculo não foi mapeado a tempo ou não havia visibilidade suficiente. O objetivo deste experimento foi testar as limitações do sistema criando situações propositais de dificuldade.

**Tabela 6.3:** Dados do experimento em simulação

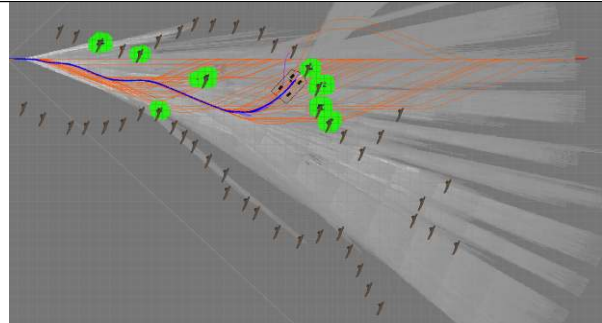
Ordem de execução	Distância (m)	Tempo (s)	Velocidade média (m/s)	Obstáculos detectados	Resultado final
1	66,67	104,25	0,639	22	<i>concluiu</i>
2	—	—	—	8	<i>parou</i>
3	65,21	103,50	0,630	23	<i>concluiu</i>
4	—	—	—	21	<i>colidiu</i>
5	64,43	133,70	0,482	19	<i>concluiu</i>
6	59,84	113,21	0,528	8	<i>concluiu</i>
7	63,98	96,00	0,667	21	<i>concluiu</i>
8	65,07	97,60	0,667	20	<i>concluiu</i>
9	64,49	96,70	0,667	25	<i>concluiu</i>
10	—	—	—	19	<i>colidiu</i>
11	63,14	94,90	0,665	20	<i>concluiu</i>
12	—	—	—	9	<i>colidiu</i>
13	64,00	105,20	0,608	23	<i>concluiu</i>
14	—	—	—	10	<i>colidiu</i>
15	58,42	99,10	0,589	8	<i>concluiu</i>
16	64,18	96,90	0,662	20	<i>concluiu</i>
17	—	—	—	9	<i>colidiu</i>
18	—	—	—	9	<i>parou</i>
19	63,82	99,30	0,642	17	<i>concluiu</i>
20	64,49	102,30	0,630	26	<i>concluiu</i>
21	—	—	—	18	<i>colidiu</i>
22	63,89	96,50	0,662	18	<i>concluiu</i>
23	65,83	104,70	0,628	23	<i>concluiu</i>
24	65,87	104,21	0,632	25	<i>concluiu</i>

**Tabela 6.4:** Visualização do experimento em simulação

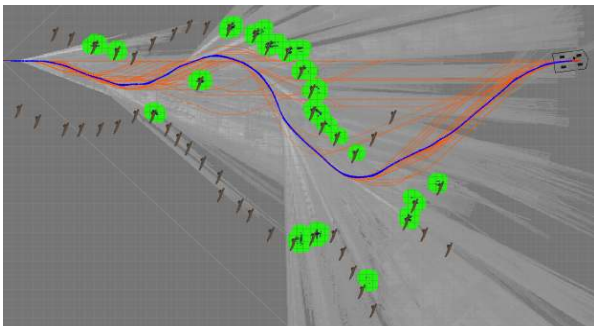
## Imagens



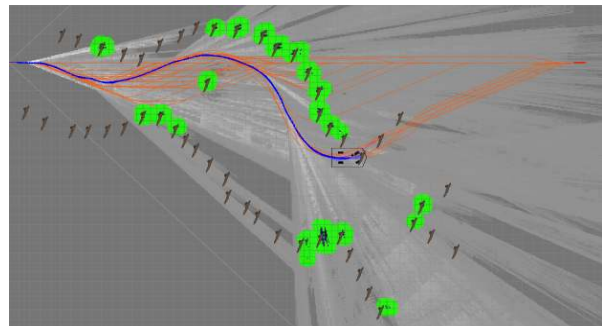
1



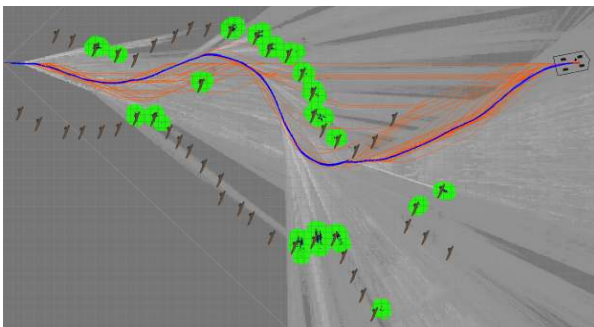
2



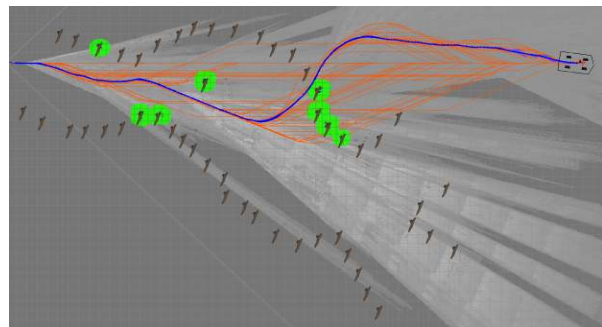
3



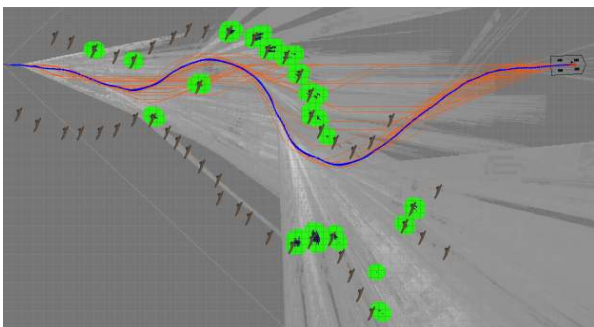
4



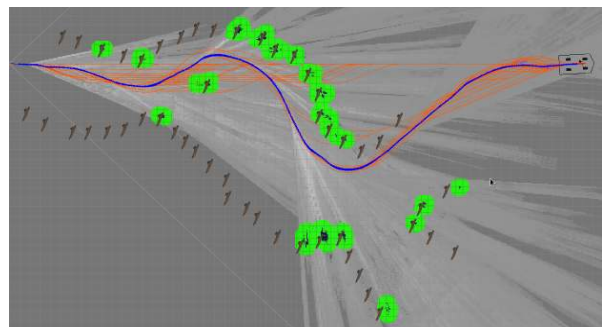
5



6



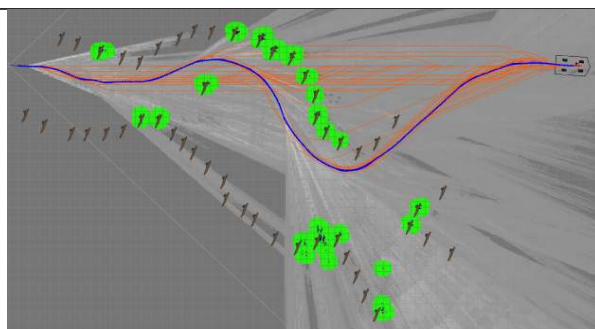
7



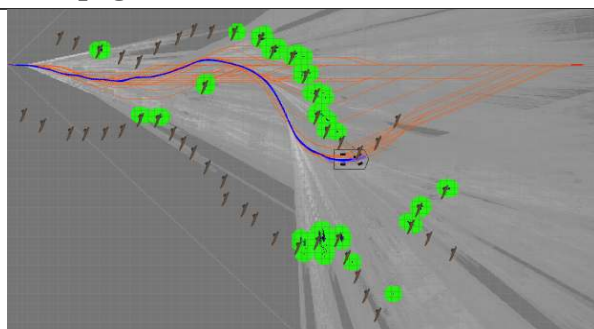
8

Continua na próxima página

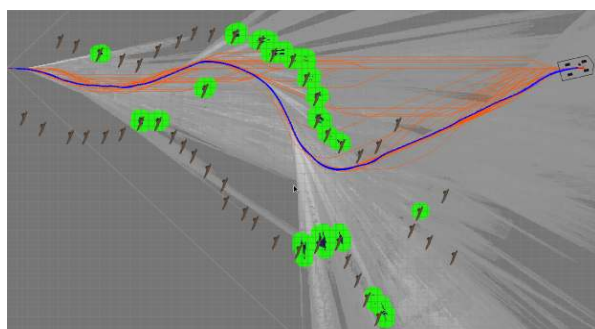
Tabela 6.4 – continuação da página anterior



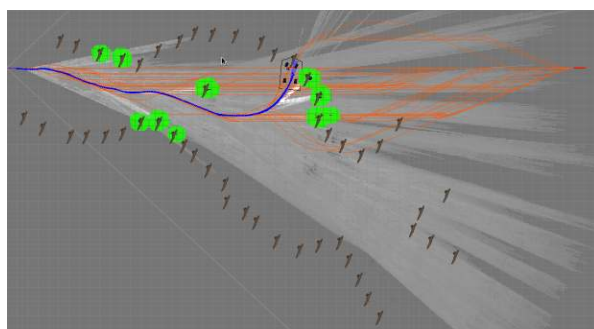
9



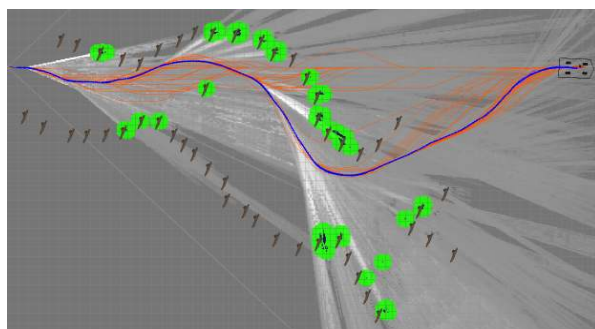
10



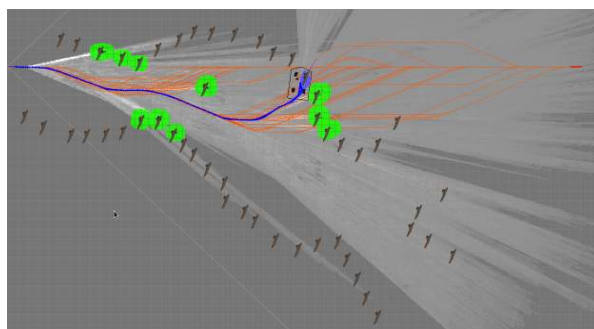
11



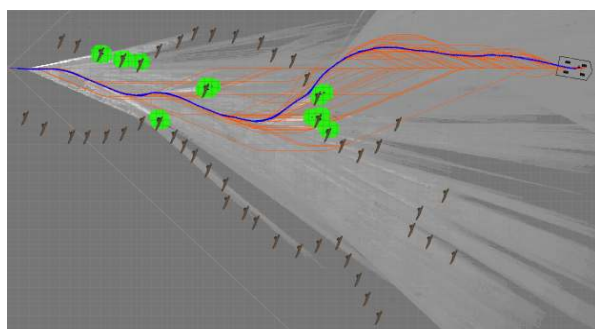
12



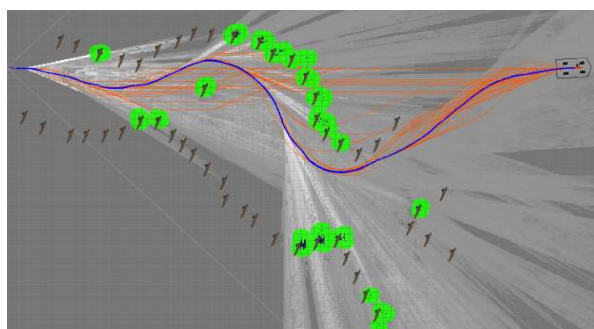
13



14



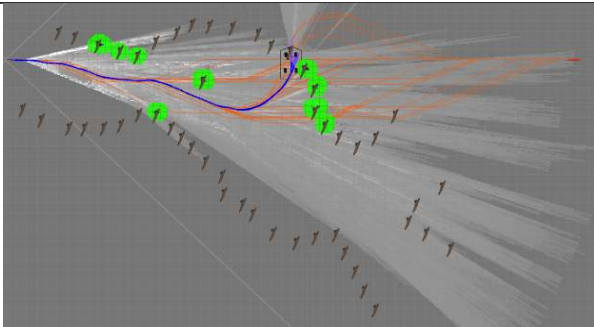
15



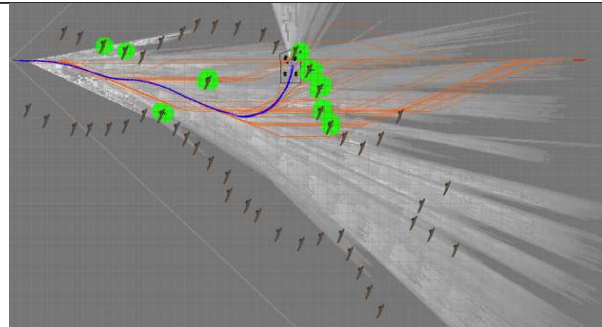
16

Continua na próxima página

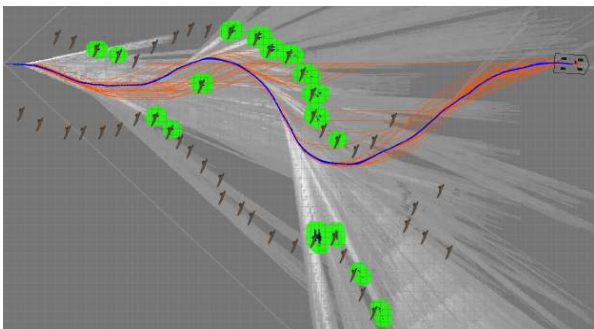
Tabela 6.4 – continuação da página anterior



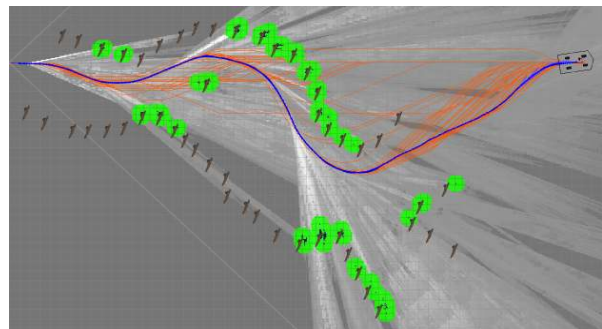
17



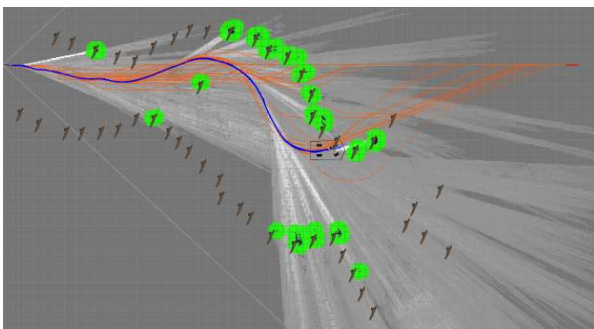
18



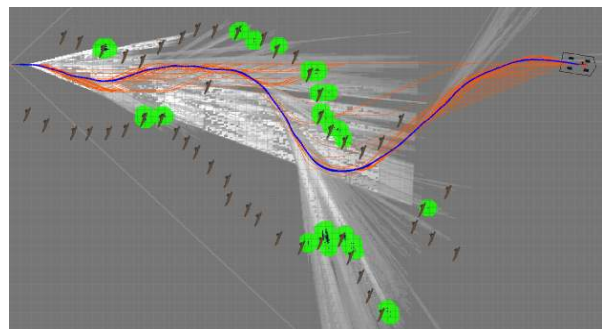
19



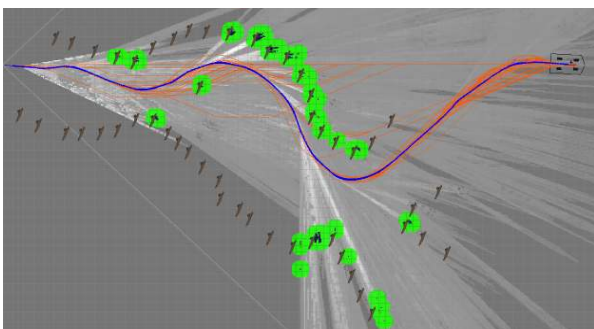
20



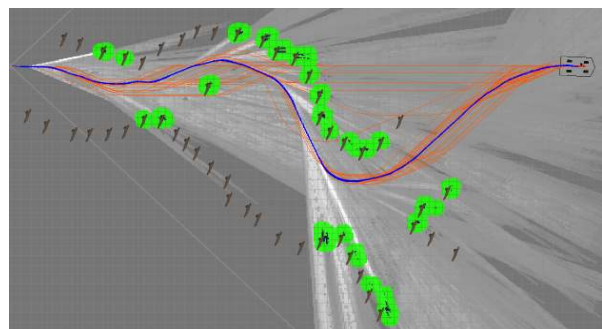
21



22



23



24



### 6.1.2 Segundo cenário de teste

O segundo experimento em simulação teve como principal propósito testar o sistema, sem alterações, em um cenário semiestruturado comumente encontrado em aplicações agrícolas como pomares (*Figura 6.5*).



Pomar de maçãs - *Harbes Farms*



Pomar de laranjas - *Harvey Fresh*

**Figura 6.5:** Exemplos de cenários de pomares

Neste tipo de cenário o veículo deve se manter entre corredores cercados pela vegetação que compõe o pomar. No final de cada corredor há uma área de retorno, devendo o veículo fazer a manobra de contorno e se posicionar em direção a um outro corredor, e assim sucessivamente até cobrir toda a área determinada. O cenário elaborado no simulador é apresentado na *Figura 6.6*.

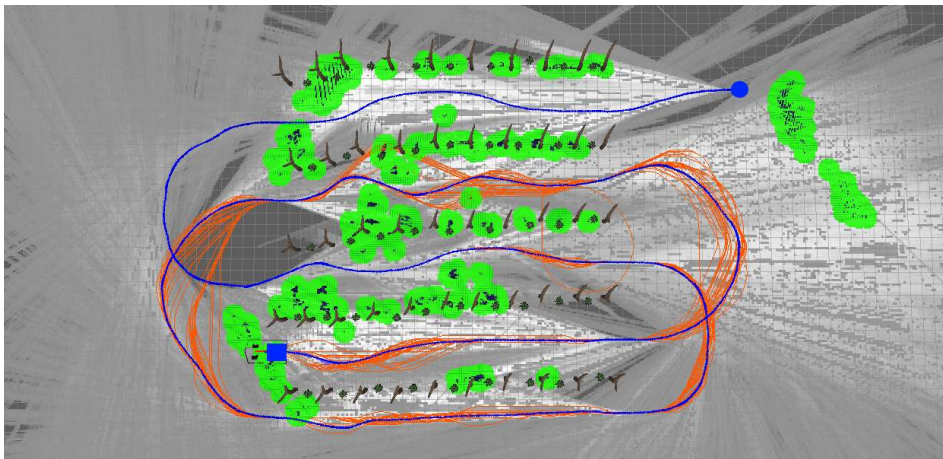


**Figura 6.6:** Imagens do cenário em simulação para o segundo experimento de teste

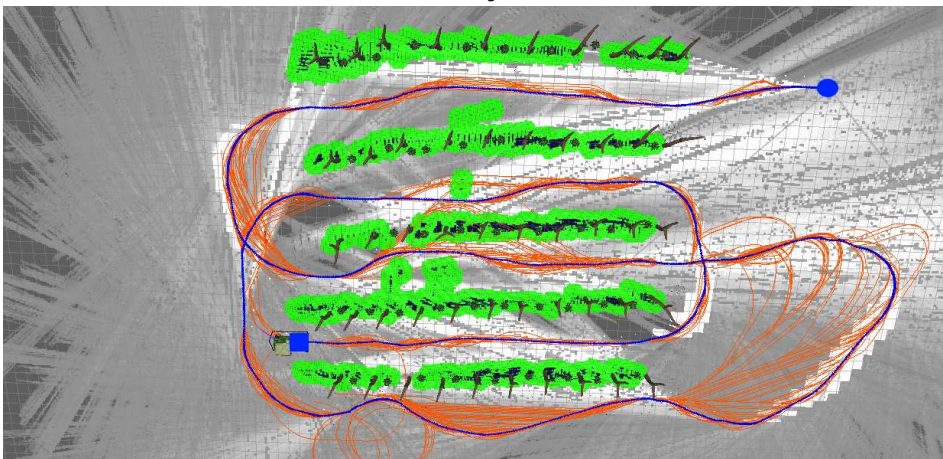
Foram executadas 2 simulações neste ambiente, tendo o veículo executado os trajetos com sucesso em todas as execuções. Os pontos de destino para o planejamento das trajetórias foram selecionados em locais determinados tal que o veículo executasse a entrada e saída de cada corredor, sendo posicionados no início e no final de cada corredor. A troca de corredor e o

espaçamento entre cada troca foi escolhido de acordo com a capacidade do veículo de executar a manobra de retorno devido a sua dinâmica de esterçamento.

Em alguns casos foram fornecidos pontos intermediários na manobra de retorno para que o veículo executasse uma trajetória mais fechada e condizente com o propósito da aplicação, em outros casos foi deixado o veículo navegar livremente até executar a manobra de retorno. Na *Figura 6.7* é possível visualizar o comportamento da navegação nas execuções (1 e 2). O traçado em azul representa o caminho percorrido pelo veículo, em verde, as regiões que delimitam o custo dos obstáculos detectados, em laranja, os planejamentos de trajetória ao longo da execução. Mesmo o sistema não sendo adaptado para tirar proveito do conhecimento prévio da estruturação básica do cenário o desempenho foi suficiente para completar a tarefa de forma autônoma sem colisão.



Execução 1



Execução 2

**Figura 6.7:** Visualização do comportamento da navegação no segundo experimento simulado. O início do percurso está marcado pelo círculo e o final pelo quadrado, em azul

## 6.2 Validação em Ambiente Real

A utilização do veículo real em um ambiente externo não estruturado teve como objetivo validar o sistema desenvolvido. O cenário real não foi replicado em simulação pois as características físicas não seriam semelhantes, principalmente o nível de detalhes. Desta forma, uma comparação direta entre o ambiente real e o ambiente simulado não seria válida fazendo uso dos mesmos critérios. No cenário real, o sistema de percepção foi ajustado para operar em 10 Hz, os demais subsistemas operaram idênticos ao caso simulado. O equipamento utilizado no veículo real foi um computador Inter Core i7 com 16 Gb de memória. A velocidade máxima do veículo foi ajustada para 3 m/s.

Foram utilizados dois ambientes para a validação, o primeiro ambiente consiste em uma área de estacionamento com terreno plano e árvores com um espaçamento regular. Como é possível ver na *Figura 6.8*, os troncos das árvores têm diferentes inclinações e galhos que precisam ser mapeados tridimensionalmente para evitar a colisão do veículo. O segundo ambiente consiste em uma área arborizada com gramado e terreno irregular contendo diferentes tipos de vegetação e estruturas que obstruem a passagem do veículo como pode ser visto na *Figura 6.10*. Ambos ambientes ficam situados no Campus I da Universidade de São Paulo em São Carlos, sendo as coordenadas de GPS  $(-22,006413; -47,895093)$  para o primeiro ambiente (*Figura 6.9*) e  $(-22,010340; -47,896668)$  para o segundo ambiente (*Figura 6.11*), em latitude e longitude.



**Figura 6.8:** Primeiro ambiente real de validação



**Figura 6.9:** Imagem de satélite do primeiro ambiente de validação - *Google Maps*

As imagens capturadas pela câmera no ambiente real apresentam uma maior qualidade em termos de textura e nível de detalhamento, que para o método de disparidade estéreo utilizado permite um melhor casamento dos pontos entre as imagens. Estas características permitem gerar nuvens de pontos mais densas e mais representativas da cena. Na *Figura 6.12* pode ser

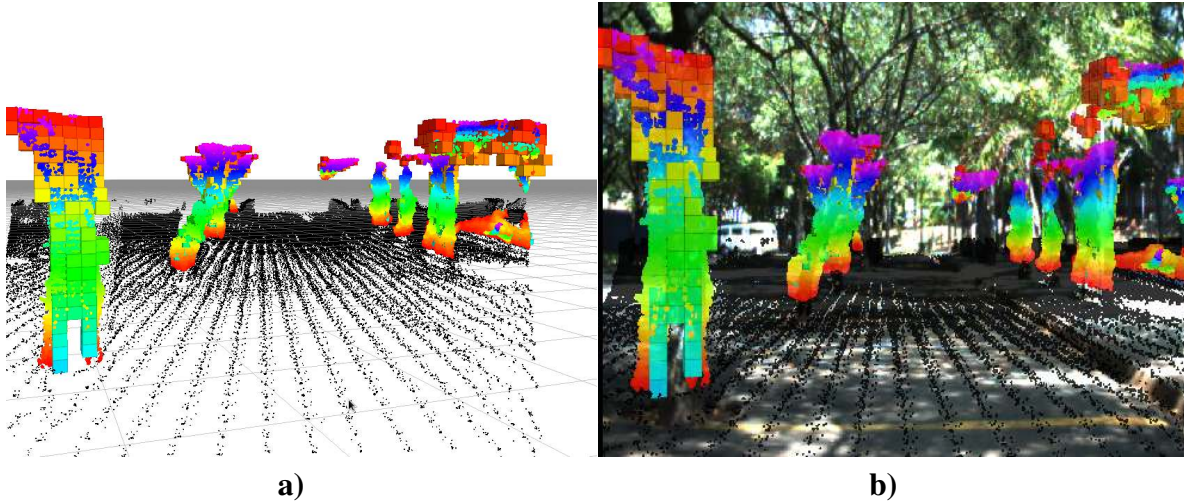


**Figura 6.10:** Segundo ambiente real de validação



**Figura 6.11:** Imagem de satélite do segundo ambiente de validação - *Google Maps*

vista a detecção do plano do chão (em preto) e a marcação dos obstáculos (em cores) obtidas do primeiro ambiente de validação. A qualidade da percepção e da ocupação espacial pode ser vista pelos contornos dos obstáculos marcados. Também, o sistema de localização permitiu marcar os obstáculos no mapa nas devidas posições em relação ao cenário real.

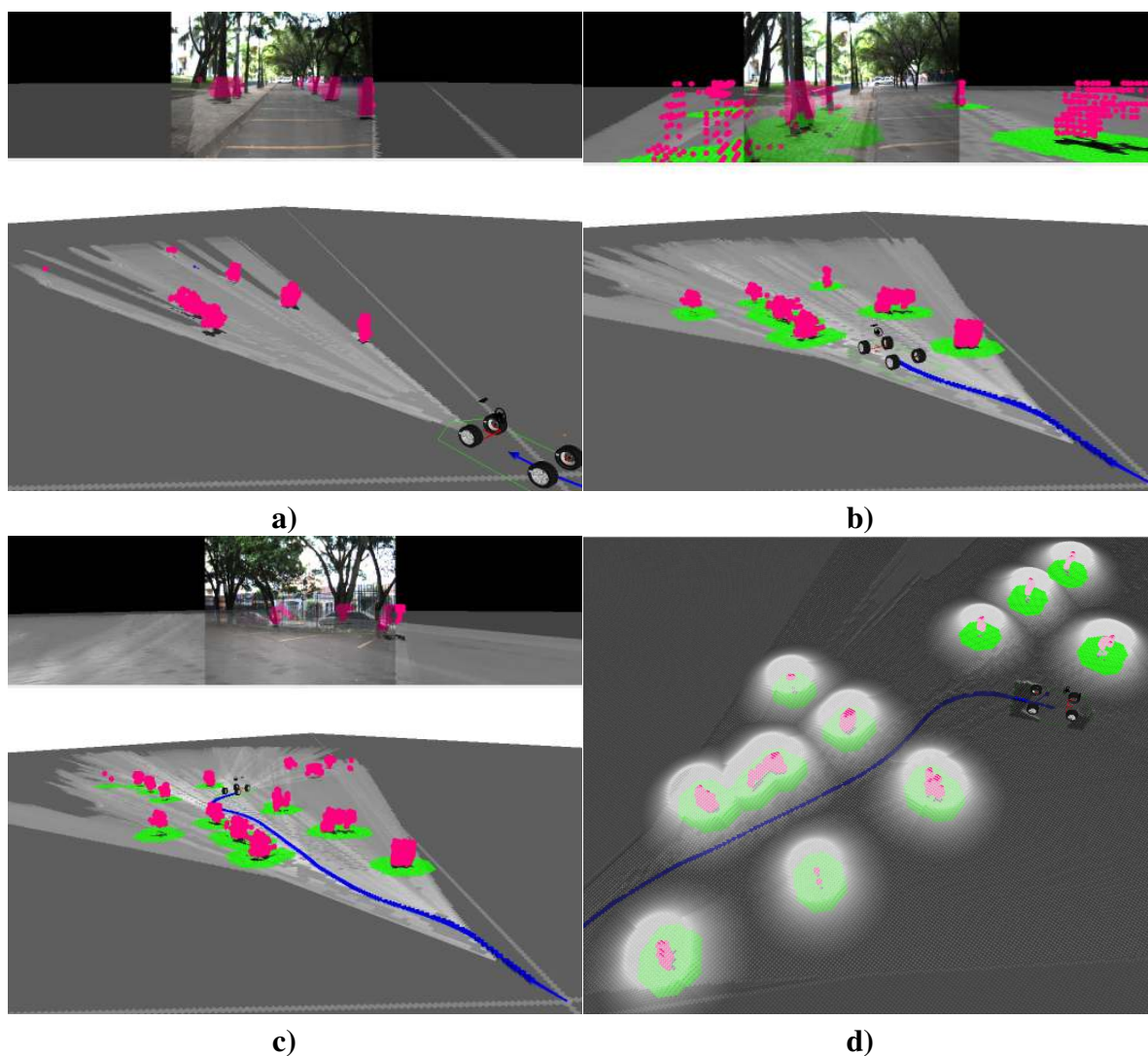


**Figura 6.12:** Detecção do plano do chão (em preto) e marcação dos obstáculos (em cores). Visualização (a) e projeção da visualização na cena (b)

A validação consistiu em executar algumas manobras de forma autônoma nos dois ambientes selecionados. As manobras compreendem a detecção de obstáculos e o planejamento de trajetória para a posição final desejada do veículo, levando em consideração o mapa de navegabilidade. A seguir são apresentados o comportamento do veículo em cada cenário, tendo sido executadas com toda a atuação controlada pelo sistema.

### 6.2.1 Manobra no primeiro ambiente

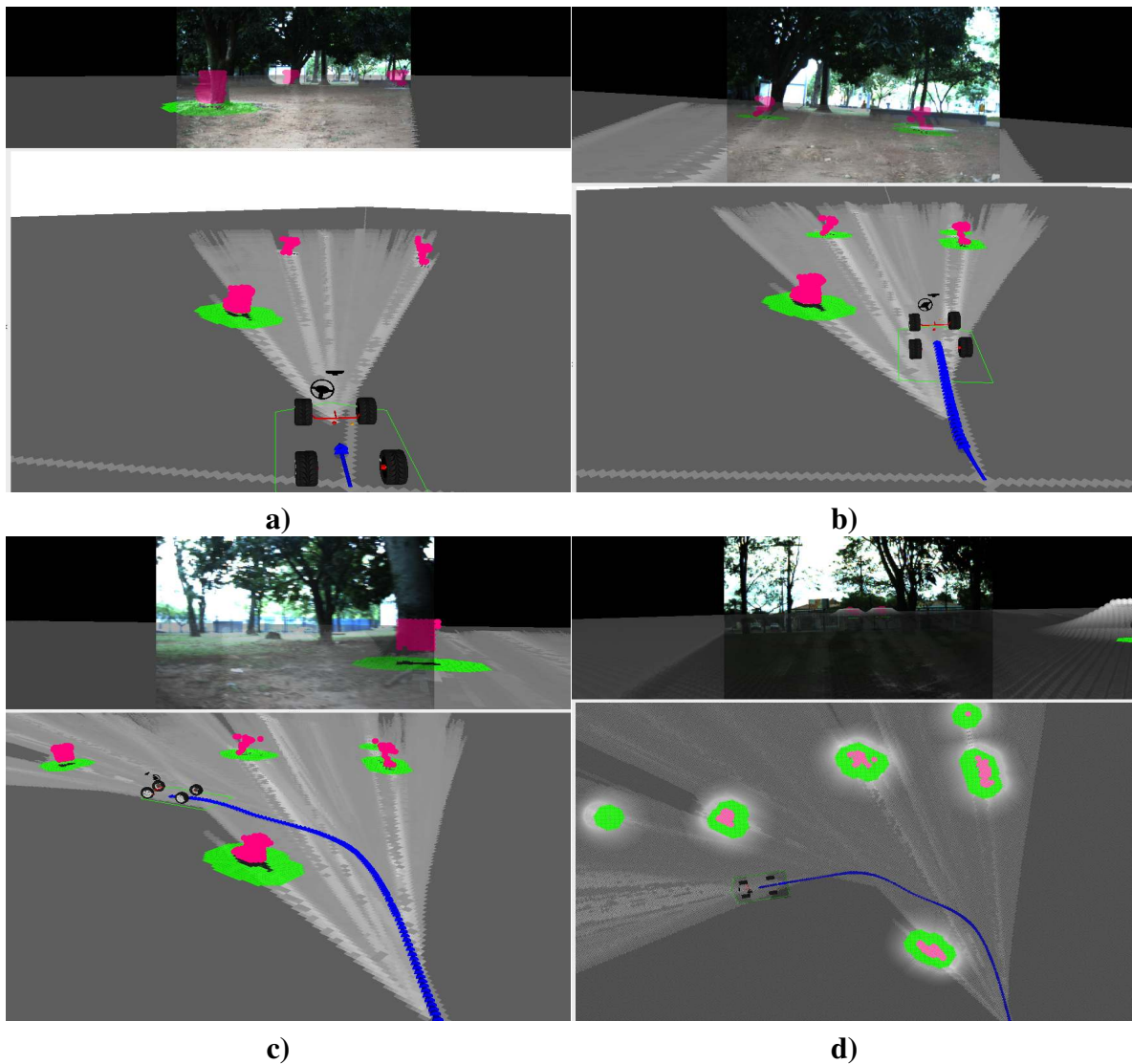
A *Figura 6.13* apresenta uma manobra executada no primeiro ambiente, sendo uma navegação entre obstáculos com um desvio à direita. Apesar de existir estruturas como a via e a calçada, o veículo não faz distinção destas zonas pois não impedem que o mesmo faça a manobra. Em **a)** é exibido o mapeamento inicial. Em **b)** e **c)** são exibidas a visualização da percepção do veículo e os mapeamentos efetuados durante o percurso. Em rosa, os obstáculos detectados. Em verde, os custos inflados. Em azul, o trajeto executado marcado pela odometria. É possível visualizar no plano do chão o mapa probabilístico de navegabilidade. Na imagem **d)** pode ser visto o mapa de custo gerado pelo mapa de navegabilidade e o caminho escolhido pelo planejador de trajetória e executado pelo veículo.



**Figura 6.13:** Visualização da manobra autônoma no primeiro ambiente de validação

### 6.2.2 Primeira manobra no segundo ambiente

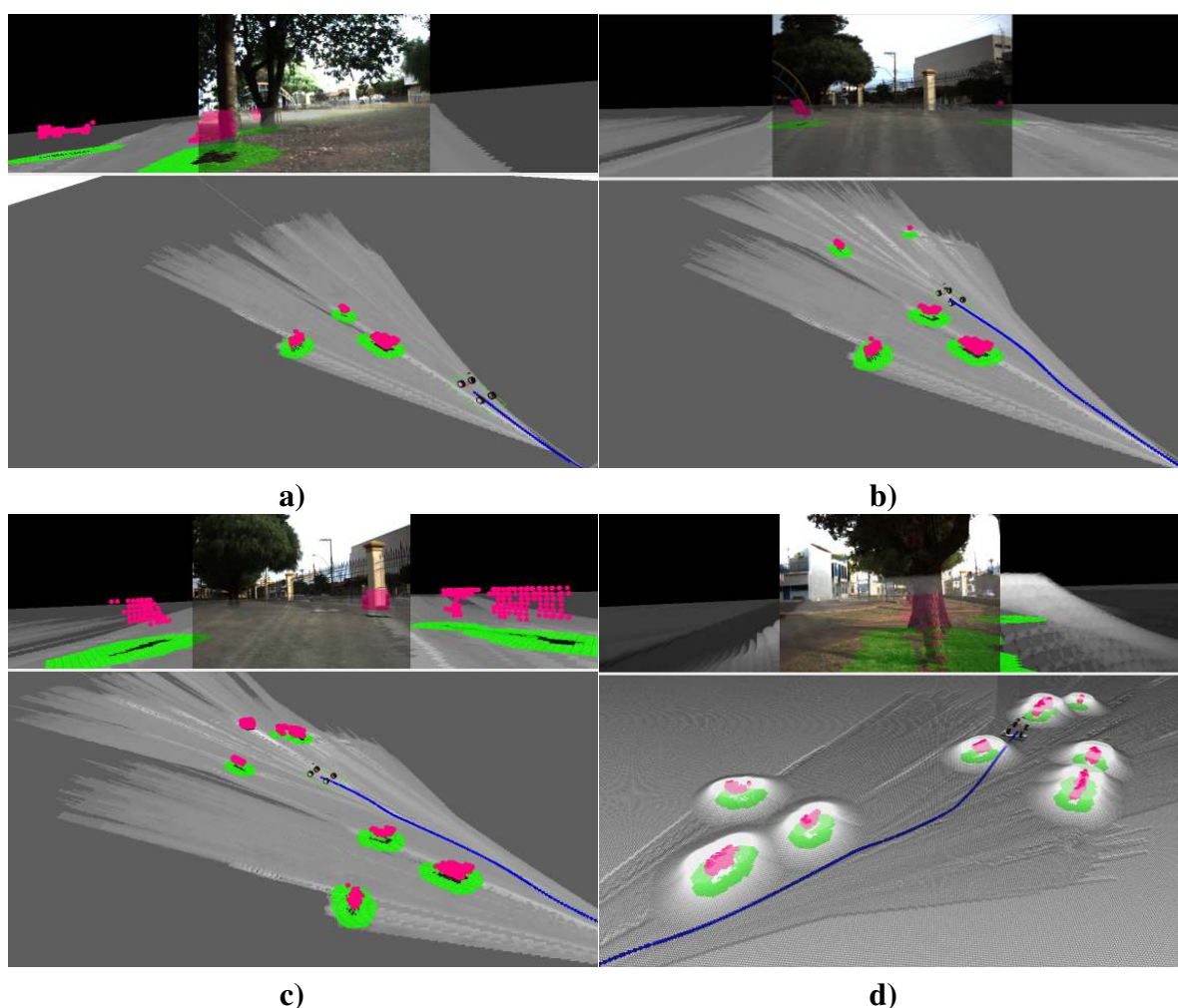
A *Figura 6.14* apresenta uma manobra executada no segundo ambiente, sendo uma navegação entre obstáculos com um desvio à esquerda em terreno irregular. Em **a)** é exibido o mapeamento inicial. Em **b)** e **c)** são exibidas a visualização da percepção do veículo e os mapeamentos efetuados durante o percurso. Também, é possível ver a inclinação do veículo em relação ao terreno irregular em **b)**. Em rosa, os obstáculos detectados. Em verde, os custos inflados. Em azul, o trajeto executado marcado pela odometria. É possível visualizar no plano do chão o mapa probabilístico de navegabilidade. Na imagem **d)** pode ser visto o mapa de custo gerado pelo mapa de navegabilidade e o caminho escolhido pelo planejador de trajetória e executado pelo veículo.



**Figura 6.14:** Visualização da primeira manobra autônoma no segundo ambiente de validação

### 6.2.3 Segunda manobra no segundo ambiente

A *Figura 6.15* apresenta outra manobra executada no segundo ambiente com terreno irregular e obstáculos com estruturas diversas. O veículo foi destinado a 43 metros em linha reta da posição inicial. No caminho haviam obstáculos que foram detectados pelo sistema de percepção e o planejamento executou os desvios necessários. Em **a)**, **b)** e **c)** são exibidas a visualização da percepção do veículo e os mapeamentos efetuados durante o percurso. É possível ver que alguns obstáculos não estavam no campo de visão ou estavam em oclusão inicialmente, sendo mapeados ao longo do percurso e tendo o planejador traçado os desvios necessários. Em rosa, os obstáculos detectados. Em verde, os custos inflados. Em azul, o trajeto executado marcado pela odometria. É possível visualizar no plano do chão o mapa probabilístico de navegabilidade. Na imagem **d)** pode ser visto o mapa de custo gerado pelo mapa de navegabilidade e o caminho escolhido pelo planejador de trajetória e executado pelo veículo.



**Figura 6.15:** Visualização da segunda manobra autônoma no segundo ambiente de validação

## 6.3 Vídeos

Foram disponibilizados vídeos acessíveis pela Internet com a visualização do sistema em execução demonstrando cada componente desenvolvido assim como o sistema completo em cenários de teste. A relação dos vídeos está listada na *Tabela 6.5*.

**Tabela 6.5:** Videos disponibilizados na Internet

Descrição	Endereço
Modelagem do veículo em simulação	<a href="http://youtu.be/deF_QIhFNtM">http://youtu.be/deF_QIhFNtM</a>
Teste de estabilidade do modelo simulado em terreno irregular	<a href="http://youtu.be/Hq-wrtJS17Y">http://youtu.be/Hq-wrtJS17Y</a>
Simulação de GPS	<a href="http://youtu.be/rUFdw6cYkH0">http://youtu.be/rUFdw6cYkH0</a>
Simulação de câmera estéreo	<a href="http://youtu.be/MrTNStg4kc0">http://youtu.be/MrTNStg4kc0</a>
Segmentação da nuvem de pontos em obstáculos	<a href="http://youtu.be/4fFCsbEqUdY">http://youtu.be/4fFCsbEqUdY</a>
Representação da ocupação espacial pelo OctoMap	<a href="http://youtu.be/N_vZDIXTsKo">http://youtu.be/N_vZDIXTsKo</a>
Controle Ackermann	<a href="http://youtu.be/JsZJ19NUZ3g">http://youtu.be/JsZJ19NUZ3g</a>
Planejamento de trajetória	<a href="http://youtu.be/dFRLzV6uVvo">http://youtu.be/dFRLzV6uVvo</a>
Desvio de obstáculos	<a href="http://youtu.be/3rNO7ehn6_4">http://youtu.be/3rNO7ehn6_4</a>
Sistema de navegação autônoma baseado em visão	<a href="http://youtu.be/z-ZBuiKYSRQ">http://youtu.be/z-ZBuiKYSRQ</a>
Execução do experimento no segundo cenário de simulação	<a href="http://youtu.be/NXVP8Ga8kFg">http://youtu.be/NXVP8Ga8kFg</a>
Visualização de manobra autônoma em ambiente real (primeiro ambiente)	<a href="http://youtu.be/3wjwDYqRl_8">http://youtu.be/3wjwDYqRl_8</a>
Visualização de manobra autônoma em ambiente real (segundo ambiente, primeira manobra)	<a href="http://youtu.be/QLhXOKPgrN0">http://youtu.be/QLhXOKPgrN0</a>
Visualização de manobra autônoma em ambiente real (segundo ambiente, segunda manobra)	<a href="http://youtu.be/Z-JMKwN4EqE">http://youtu.be/Z-JMKwN4EqE</a>



---

## Conclusão

---

Este trabalho proporcionou uma visão abrangente de um sistema de navegação autônomo para veículos terrestres, através da pesquisa, proposta e implementação de um sistema completo de navegação em ambientes não estruturados. Como foram abordadas todas as etapas necessárias para projetar esse tipo de sistema, sendo tratadas as questões referentes à modelagem do veículo, localização, mapeamento, controle, planejamento de trajetória, atuação e percepção, pôde-se experienciar os caminhos que estão sendo seguidos no âmbito da pesquisa e desenvolvimento para solucionar os diversos problemas relacionados a estes temas. A fidelidade dos modelos construídos para simulação em relação aos equipamentos utilizados permitiu a execução de testes de um modo seguro e próximos ao comportamento real. A simulação permite maior repetibilidade dos experimentos e a replicação dos mesmos por outros grupos de pesquisa.

Apesar de não ter sido efetuado um tratamento explícito para os obstáculos dinâmicos na cena, o replanejamento da trajetória sendo executado constantemente acaba contemplando a dinâmica da cena através da atualização dos custos e probabilidades de ocupação do espaço que ocorre durante a navegação do veículo. O planejamento acaba traçando novas rotas sempre atualizadas com a ocupação espacial mais recente do cenário, incorporando desta forma a dinâmica do ambiente. Isto permite que o veículo pare (fique bloqueado) caso um obstáculo esteja obstruindo a sua trajetória, assim como foi demonstrado nas simulações. Além disto, a utilização da OctoMap para representar probabilisticamente a ocupação espacial permitiu que os ruídos

apresentados pelo sistema de percepção e localização fossem incorporados a operação do sistema, sem um maior prejuízo na execução da tarefa a ele atribuída. Desta forma, o mapa de navegabilidade gerado permitiu o planejamento de trajetórias e a navegação autônoma do veículo em ambiente externo não estruturado.

Durante a validação foi possível observar que o sistema de visão se beneficiou do maior detalhamento das imagens no ambiente real, resultando em um melhor mapa de disparidade, o que era esperado. Já o sistema de localização teve desempenho inferior no equipamento real, o que também era esperado devido ao maior ruído existente nos equipamentos de medição inercial e no sinal de GPS. Também, o veículo real apresentou folga na barra da direção provocando erro no sistema de localização por *dead-reckoning*. Apesar das limitações dos equipamentos utilizados, onde o sistema de percepção possuía um campo de visão restrito e insuficiente para executar algumas manobras, assim como para realizar um mapeamento completo do ambiente em torno do veículo, o sistema de navegação se demonstrou capaz de conduzir o veículo autonomamente em cenários com terreno irregular e com obstáculos diversos.

O ambiente de simulação foi concebido com o propósito de testar o sistema em cenários com necessidades de manobras que verificassem as limitações do sistema, não apenas o caso ideal. Neste trabalho, foi abordado um sistema completo de navegação autônoma com todos os componentes necessários para a sua concepção. Os requisitos de sensores e até mesmo o veículo utilizado permitem a portabilidade deste projeto e a replicação dos experimentos. A partir desta base, é possível estudar cada parte do sistema mais profundamente, testando o comportamento em uma execução completa de uma navegação autônoma e validando em ambiente real. Além desta contribuição, se construiu objetivamente um sistema baseado em visão com uma configuração simplificada capaz de produzir uma percepção do ambiente suficiente para a navegação autônoma.

## 7.1 Principais Contribuições

As principais contribuições deste projeto são elencadas abaixo:

- i. Modelagem do veículo real CaRINA I e simulação deste veículo com as restrições de esterçamento do modelo cinemático de Ackermann, também, a modelagem de um sistema de suspensão no veículo para simulação em terrenos irregulares;
- ii. Integração e adaptação dos sensores simulados, correspondentes aos sensores do veículo real, incluindo a câmera estéreo, encoder, GPS, IMU e bússola;

- iii. Proposta de um modelo sensorial para marcação de ocupação espacial probabilística de pontos produzidos por câmera estéreo levando em consideração o erro inerente do cálculo de disparidade;
- iv. Descrição de um processo de filtragem de nuvem de pontos produzido por câmera estéreo a partir do processo de segmentação de blocos e remoção de ruídos provocados pelo processo de busca pela disparidade baseada na informação fotométrica das imagens;
- v. Aplicação e adaptação de um modelo probabilístico de ocupação espacial para gerar um mapa de navegabilidade convertido diretamente para mapa de custos;
- vi. Demonstração de um sistema autônomo em plataforma simulada e real, integrando os diversos componentes e módulos descritos acima, baseado em apenas uma câmera estéreo como sensor exteroceptivo capaz de executar a navegação em ambientes não estruturados.

## 7.2 Trabalhos Futuros

Uma possibilidade de extensão do trabalho é a aplicação da ideia de se utilizar o algoritmo T-RRT, ou outro algoritmo baseado em árvore de expansão, para gerar destinos intermediários em um percurso longo baseando-se em um mapa topográfico ou de elevação (DEM) que serviria de mapa custos, por exemplo. A validade desta aplicação foi testada durante o projeto porém não foi integrada como parte do sistema.

A integração de maior informação sensorial para aumentar o ângulo de visibilidade do sistema de percepção é uma necessidade para uma aplicação real. Uma possibilidade de trabalho é verificar o desempenho do sistema com outras combinações de sensores e o uso de mais de uma câmera estéreo. Apesar de já ter sido testado com sucesso a modelagem de manobras em marcha ré no planejamento de trajetórias, esta funcionalidade poderia ser integrada completamente ao sistema sendo melhor testada e validada.

O modelo sensorial desenvolvido pode ser estendido para escolher as probabilidades baseado nos parâmetros da câmera e demais parâmetros do sistema como a velocidade do veículo e frequência de captura das imagens, assim como outros requisitos como a distância mínima para um obstáculo ser marcado como obstáculo.



---

# Referências Bibliográficas

---

- ANGERMANN, M.; ROBERTSON, P. Footslam: Pedestrian simultaneous localization and mapping without exteroceptive sensors - hitchhiking on human perception and cognition. *Proceedings of the IEEE*, v. 100, n. Special Centennial Issue, p. 1840–1848, 2012.
- BADINO, H.; FRANKE, U.; PFEIFFER, D. The Stixel World - a compact medium level representation of the 3D-world. In: *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, 2009, p. 51–60.
- BENENSON, R.; MATHIAS, M.; TIMOFTE, R.; VAN GOOL, L. Fast stixels estimation for fast pedestrian detection. In: *ECCV, CVVT workshop*, 2012.
- BENENSON, R.; TIMOFTE, R.; VAN GOOL, L. Stixels estimation without depthmap computation. In: *ICCV, CVVT workshop*, 2011.
- BERGERMAN, M.; SINGH, S.; HAMNER, B. Results with autonomous vehicles operating in specialty crops. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, p. 1829–1835.
- BEVLY, D.; COBB, S. *Gnss for vehicle control*. GNSS technology and applications series. Artech House, 2010.
- BHATTACHARYA, S.; LIKHACHEV, M.; KUMAR, V. Topological constraints in search-based robot path planning. *Autonomous Robots*, v. 33, p. 273–290, 2012.
- BORENSTEIN, J.; KOREN, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *Robotics and Automation, IEEE*, v. 7, n. 3, p. 278–288, 1991.
- BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. Cambridge, MA: O’Reilly, 2008.
- BROCK, O.; KHATIB, O. High-speed navigation using the global dynamic window approach. In: *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999, p. 341–346 vol.1.

- CHOSSET, H.; LYNCH, K. M.; HUTCHINSON, S.; KANTOR, G.; BURGARD, W.; KAVRAKI, L. E.; THRUN, S. *Principles of robot motion: theory, algorithms, and implementation*. The MIT Press, 625 p., 2005.
- DUDEK, G.; JENKIN, M. *Computational principles of mobile robotics*. Cambridge University Press, 2000.
- FAUGERAS, O. *Three-dimensional computer vision: a geometric viewpoint*. Cambridge, MA, USA: MIT Press, 1993.
- FAUGERAS, O.; VIÉVILLE, T.; THERON, E.; VUILLEMIN, J.; HOTZ, B.; ZHANG, Z.; MOLL, L.; BERTIN, P.; MATHIEU, H.; FUA, P.; BERRY, G.; PROY, C. *Real-time correlation-based stereo : algorithm, implementations and applications*. Rapport de recherche RR-2013, INRIA, 1993.
- FERNANDES, L. C.; SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; SALES, D.; MENDES, C.; PRADO, M.; KLASER, R.; MAGALHÃES, A. C.; HATA, A.; PIGATTO, D.; BRANCO, K. C.; JR., V. G.; OSORIO, F. S.; WOLF, D. F. Carina intelligent robotic car: Architectural design and applications. *Journal of Systems Architecture*, v. 60, n. 4, p. 372 – 392, 2014.
- FERNANDES, L. C.; SOUZA, J. R.; SHINZATO, P. Y.; PESSIN, G.; MENDES, C. C. T.; OSORIO, F. S.; WOLF, D. F. Intelligent robotic car for autonomous navigation: Platform and system architecture. In: *II Brazilian Conference on Critical Embedded Systems - CBSEC*, 2012.
- FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, v. 24, n. 6, p. 381–395, 1981.
- FOX, D.; BURGARD, W.; THRUN, S. The dynamic window approach to collision avoidance. *Robotics Automation Magazine, IEEE*, v. 4, n. 1, p. 23–33, 1997.
- FUSIELLO, A.; TRUCCO, E.; VERRI, A. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, v. 12, n. 1, p. 16–22, 2000.
- HORNUNG, A.; WURM, K.; BENNEWITZ, M.; STACHNISS, C.; BURGARD, W. Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, v. 34, n. 3, p. 189–206, 2013.
- HOWARD, T. M.; KELLY, A. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, v. 26, n. 2, p. 141–166, 2007.
- JACKEL, L. D.; KROTKOV, E.; PERSCHBACHER, M.; PIPPINE, J.; SULLIVAN, C. The darpa lagr program: Goals, challenges, methodology, and phase i results. *Journal of Field Robotics*, v. 23, n. 11-12, p. 945–973, 2006.
- JAILLET, L.; CORTES, J.; SIMEON, T. Transition-based rrt for path planning in continuous cost spaces. In: *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, 2008, p. 2145–2150.

- KELLY, A.; AMIDI, O.; HERMAN, H.; PILARSKI, T.; STENTZ, A.; VALLIDIS, N.; WARNER, Y. Toward reliable off road autonomous vehicles operating in challenging environments. *The International Journal of Robotics Research*, v. 25, p. 5–6, 2006.
- KHALEGHI, B.; AHUJA, S.; WU, Q. A new miniaturized embedded stereo-vision system (MESVS-I). In: *Proceedings of the 2008 Canadian Conference on Computer and Robot Vision*, IEEE Computer Society, 2008, p. 26–33.
- KHOSHELHAM, K.; ELBERINK, S. O. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors : journal on the science and technology of sensors and biosensors*, v. 12, n. 2, p. 1437–1454, 2012.
- KOENIG, N.; HOWARD, A. Design and use paradigms for gazebo, an open-source multi-robot simulator. In: *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, 2004, p. 2149–2154 vol.3.
- KONOLIGE, K.; AGRAWAL, M.; BLAS, M. R.; BOLLES, R. C.; GERKEY, B.; SOLÀ, J.; SUNDARESAN, A. Mapping, navigation, and learning for off-road traversal. *Journal of Field Robotics*, v. 26, n. 1, p. 88–113, 2009.
- LATEGAHN, H.; GEIGER, A.; KITT, B. Visual SLAM for autonomous ground vehicles. *2011 IEEE International Conference on Robotics and Automation*, p. 1732–1737, 2011.
- LEE, K.; CHUNG, W.; YOO, K. Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy. *Mechatronics*, v. 20, n. 5, p. 582–595, 2010.
- LIKHACHEV, M.; FERGUSON, D.; GORDON, G.; STENTZ, A.; THRUN, S. Anytime search in dynamic graphs. *Artificial Intelligence*, v. 172, n. 14, p. 1613–1643, 2008.
- LUETTEL, T.; HIMMELSBACH, M.; WUENSCH, H.-J. Autonomous ground vehicles - concepts and a path to the future. *Proceedings of the IEEE*, v. 100, n. Special Centennial Issue, p. 1831–1839, 2012.
- MAGALHÃES, A. C.; PRADO, M.; JR, V. G.; WOLF, D.; OSÓRIO, F. S. Autonomous vehicle navigation in semi-structured urban environment. In: *8th Symposium on Intelligent Autonomous Vehicles (IAV)*, 2013.
- MARR, D. *Vision: A computational investigation into the human representation and processing of visual information*. New York, NY, USA: Henry Holt and Co., Inc., 1982.
- MAYBECK, P. The kalman filter: An introduction to concepts. In: COX, I.; WILFONG, G., eds. *Autonomous Robot Vehicles*, Springer New York, p. 194–204, 1990.
- MENDES, C.; WOLF, D. F. Desvio de Obstáculos Utilizando um Método Estéreo Semi-global. In: *VIII Encontro Nacional de Inteligência Artificial (ENIA)*, 2011, p. 12.

- MEYER, J.; SENDOBRY, A.; KOHLBRECHER, S.; KLINGAUF, U.; STRYK, O. Comprehensive simulation of quadrotor uavs using ros and gazebo. In: NODA, I.; ANDO, N.; BRUGALI, D.; KUFFNER, J., eds. *Simulation, Modeling, and Programming for Autonomous Robots*, v. 7628 de *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, p. 400–411, 2012.
- MUNDY, J. L.; ZISSERMAN, A. Repeated structures: Image correspondence constraints and 3D structure recovery. *Applications of invariance in computer vision*, p. 89–106, 1994.
- OZGUNER, U.; ACARMAN, T.; REDMILL, K. *Autonomous ground vehicles*. Artech House ITS series. Artech House, 2011.
- PEDRO, L.; DE PAULA CAURIN, G. Kinect evaluation for human body movement analysis. In: *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, 2012, p. 1856–1861.
- PESSIN, G. Evolução de estratégias e controle inteligente em sistemas multi-robóticos robustos. *Dissertação de Mestrado*, 2008.
- QIAN, N.; QIAN, D. N. Binocular disparity and the perception of depth. *Neuron*, v. 10, p. 359–368, 1997.
- QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. *ICRA workshop on open source software*, v. 3, n. 3.2, 2009.
- RABBANI, T.; VAN DEN HEUVEL, F. A.; VOSSELMANN, G. Segmentation of point clouds using smoothness constraint. In: *IEVM06*, 2006.
- ROTH, S. D. Ray casting for modeling solids. *Computer Graphics and Image Processing*, v. 18, n. 2, p. 109 – 144, 1982.
- RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.
- SCHARSTEIN, D.; SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, v. 47, n. 1-3, p. 7–42, 2002.
- SILVEIRA, C. L. B.; CAVALHEIRO, G. G. H.; JUNG, C. R.; JR., J. C. S. J.; MUSSE, S. R. An improved background subtraction algorithm and concurrent implementations. *Parallel Processing Letters*, v. 20, n. 1, p. 71–89, 2010.
- STENTZ, A. T. Optimal and efficient path planning for partially-known environments. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '94)*, 1994, p. 3310 – 3317.
- SUCAN, I.; MOLL, M.; KAVRAKI, L. The open motion planning library. *Robotics Automation Magazine, IEEE*, v. 19, n. 4, p. 72 –82, 2012.



- TALUKDER, A.; MANDUCHI, R.; RANKIN, A.; MATTHIES, L. Fast and reliable obstacle detection and segmentation for cross-country navigation. In: *Intelligent Vehicle Symposium, 2002. IEEE, 2002*, p. 610–618 vol.2.
- TAYLOR, K.; LAVALLE, S. I-bug: An intensity-based bug algorithm. In: *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, IEEE, 2009, p. 3981–3986.
- THRUN, S. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, v. 99, n. 1, p. 21–71, 1998.
- THRUN, S.; STEFFEN GUTMANN, J.; FOX, D.; BURGARD, W.; KUIPERS, B. J. Integrating topological and metric maps for mobile robot navigation: A statistical approach. In: *In Proceedings of the AAAI Fifteenth National Conference on Artificial Intelligence*, 1998.
- THRUN, S.; MONTEMERLO, M.; KOLLER, D.; WEGBREIT, B.; NIETO, J.; NEBOT, E. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research*, v. 2004, p. 1–48, 2004.
- WEBB, B. What does robotics offer animal behaviour? *Animal Behaviour*, v. 60, n. 5, p. 545–558, 2000.
- WENG, J.; COHEN, P.; HERNIOU, M. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, v. 14, n. 10, p. 965–980, 1992.
- WILLSON, R. G.; SHAFER, S. A. What is the Center of the Image? *Journal of the Optical Society of America A*, v. 11, n. 11, p. 2946–2955, 1994.
- WOLF, D. F.; DO VALLE SIMÕES, E.; OSÓRIO, F. S.; JUNIOR, O. Robótica móvel inteligente: Da simulação às aplicações no mundo real. In: *Mini-Curso: Jornada de Atualização em Informática (JAI), Congresso da SBC*, 2009.
- ZHANG, J.; KANTOR, G. A.; BERGERMAN, M.; SINGH, S. Monocular visual navigation of an autonomous vehicle in natural scene corridor-like environments. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.