Localização topológica e identificação de
obstáculos por meio de sensor laser 3D (LIDAR)
para aplicação em navegação de veículos
autônomos terrestres

Danilo Habermann

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-
USP

Data de Depósito:

Assinatura: _____

Danilo Habermann

Localização topológica e identificação de obstáculos por meio de sensor laser 3D (LIDAR) para aplicação em navegação de veículos autônomos terrestres

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

Área de Concentração: Ciências de
Computação e Matemática Computacional

Orientador: Prof. Dr. Fernando Santos Osório

USP – São Carlos
Outubro de 2016

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

H1141 Habermann, Danilo
Localização topológica e identificação de
obstáculos por meio de sensor laser 3D (LIDAR) para
aplicação em navegação de veículos autônomos terrestres
/ Danilo Habermann; orientador Fernando Santos
Osório. -- São Carlos, 2016.
94 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2016.

1. Localização. 2. Classificação de Obstáculos. 3.
Veículos Autônomos. 4. Sensor Laser (LIDAR). I.
Santos Osório, Fernando, orient. II. Título.

Danilo Habermann

**Topological localization and obstacles identification
using a 3D LIDAR in areas of autonomous ground
vehicles navigation**

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics.

FINAL VERSION

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Fernando Santos Osório

**USP – São Carlos
October 2016**

AGRADECIMENTOS

Agradeço a Deus pela vida e aos meus pais por todo amor, apoio e dedicação.

Agradeço a minha esposa e meu filho que souberam entender os meus momentos de ausência junto ao lar para que esse trabalho pudesse ser concluído.

Agradeço ao Departamento de Ciência e Tecnologia do Exército por ter me dado suporte pra que realizasse esse curso e ao Coronel Eduardo Carrilho da Cunha, meu tutor acadêmico militar, que acompanhou a evolução deste trabalho.

Agradeço ao Prof. Dr. Fabio Tozeto Ramos por ter supervisionado meu trabalho durante o os seis meses que estive no *Australian Centre for Fields Robotics* na Universidade de Sydney.

Agradeço de forma especial ao Prof. Dr. Fernando Santos Osório, meu orientador, que durante todo o meu curso me guiou de forma precisa para o desenvolvimento desta tese e por ter sido durante todo esse tempo uma pessoa extremamente atenciosa e prestativa em relação a todas as necessidades pessoais e acadêmicas surgidas ao longo dessa trajetória.

Agradeço, ainda, a todos os membros do Laboratório de Robótica Móvel do ICMC-USP, que me deram preciosas dicas e grande apoio técnico para que eu pudesse realizar os testes práticos deste trabalho.

RESUMO

O emprego de veículos terrestres autônomos tem se tornado cada vez mais comum nos últimos anos em aplicações civis e militares. Eles podem ser úteis para as pessoas com necessidades especiais e para reduzir os acidentes de trânsito e o número de baixas em combate.

Esta tese aborda o problema da classificação de obstáculos e da localização do veículo em relação a um mapa topológico, sem fazer uso de GPS e de mapas digitais detalhados. Um sensor laser 3D é usado para coletar dados do ambiente.

O sistema de classificação de obstáculos extrai as *features* da nuvem de pontos e usam-nas para alimentar um classificador que separa os dados em quatro classes: veículos, pessoas, construções, troncos de árvores e postes. Durante a extração de *features*, um método original para transformar uma nuvem 3D em um *grid* 2D é proposto, o que ajuda a reduzir o tempo de processamento.

As interseções de vias de áreas urbanas são detectadas e usadas como *landmarks* em um mapa topológico. O sistema consegue obter a localização do veículo, utilizando os pontos de referência, e identifica as mudanças de direção do veículo quando este passa pelos cruzamentos.

Os experimentos demonstraram que o sistema foi capaz de classificar corretamente os obstáculos e localizar-se sem o uso de sinais de GPS.

Palavras-chave: veículo autônomo, segmentação, classificação de obstáculos, identificação de interseção de vias, localização, LIDAR.

ABSTRACT

The employment of autonomous ground vehicles, both in civilian and military applications, has become increasingly common over the past few years. Those vehicles can be helpful for disabled people and also to reduce traffic accidents.

In this thesis, approaches to the problem of obstacles classification and the localization of the vehicle in relation to a topologic map are presented. GPS devices and previous digital maps are not employed. A 3D laser sensor is used to collect data from the environment.

The obstacle classification system extracts features from point clouds and uses them to feed a classifier which separates data into four classes: vehicle, people, building and light poles/ trees. During the feature extraction, an original method to transform 3D to 2D data is proposed, which helps to reduce the processing time.

Crossing roads are detected and used as landmarks in a topological map. The vehicle performs self-localization using the landmarks and identifying direction changes through the crossing roads.

Experiments demonstrated that system was able to correctly classify obstacles and to localize itself without using GPS signals.

Keywords: autonomous ground vehicle, point clouds segmentation, obstacle classification, crossroads identification, localization, LIDAR.

LISTA DE ILUSTRAÇÕES

Figura 1.1. Veículo CaRINA 1 – em destaque o sensor Velodyne	08
Figura 1.2. Veículo CaRINA 2 equipado com sensores.....	08
Figura 2.1: Principais sistemas de um veículo autônomo.....	14
Figura 2.2: Sensores utilizados pela equipe <i>Spirit of Berlin</i>	18
Figura 2.3: Representação da varredura vertical do sensor LUX IBEO.....	18
Figura 2.4 Sensores usados pela equipe vencedora do DARPA <i>Urban Challenge</i>	19
Figura 2.5: Sensor Velodyne HDL-64 S2	21
Figura 2.6: Sensor Velodyne HDL-32E	21
Figura 2.7: Sensor Velodyne VLP-16	22
Figura 2.8: Representação da varredura de 180° do sensor laser Sick LMS-291.....	22
Figura 3.1: Representação de uma rede neural artificial (Haykin, 1998).....	26
Figura 3.2: Hiperplanos separando dados de treinamento	27
Figura 3.3: Hiperplanos separando dados de treinamento (linear e não linear)	28
Figura 3.4: Exemplo de grafo de fatores com três variáveis	29
Figura 3.5: Exemplo de um HMM representado por modelo gráfico direcionado.....	30
Figura 3.6: Exemplo de um CRF representado por modelo gráfico não direcionado ..	30
Figura 4.1 – Divisão do grid polar em M seções de tamanhos iguais.....	36
Figura 4.2 – Resultado da segmentação do trabalho de Tongtong.....	37
Figura 4.3 – Faixas circulares $b_1, b_2, \dots, b_m$ de cada uma das m seções circulares, nas quais são distribuídos os pontos do Velodyne.	37
Figura 4.4 – Os pontos de menor elevação de $b_1, b_2, \dots, b_m$ de cada uma das m seções circulares são tomados para verificar o ângulo que formam entre si.	38
Figura 4.5: Resultado da segmentação de nuvem de pontos apresentado por Douillard et al, 2011.....	39
Figura 4.6: Resultado do método de Thrun (2006).....	40
Figura 4.7: Exemplo de uma <i>range image</i> criada a partir de pontos coletados por sensor laser.....	41
Figura 4.8: Sinais de laser virtuais 2D criados a partir de nuvem de pontos 3D com intuito de identificar uma interseção de vias (Zhu et al, 2012).....	43
Figura 4.9 Identificação de cruzamentos usando a transformada de Borgefors (1986). As células com maior altura estão mais distantes dos obstáculos	44
Figura 4.10. Criação de um mapa topométrico (unidades em metro) referente ao trabalho de Baldino et al (2011).....	44
Figura 4.11. Duas interseções de vias caracterizadas por um centro e seus ramos (Muller et al, 2011).....	45
Figura 5.1 Grid com n seções circulares e m “bins”.....	48

Figura 5.2: Etapas do método de classificação	49
Figura 5.3. Os pontos 3D são relacionados a um veículo e foram obtidos após o processo de segmentação baseado no método de Klassing (2008).....	50
Figura 5.4. Os pontos 2D são uma projeção nos eixos X e Z dos pontos da figura 5.3..	50
Figura 5.5. Representação de um grid 2D normalizado de (a) um carro, (b) um pedestre, (c) uma árvore e (d) uma construção.....	51
Figura 5.6. Resultados do experimento de validação cruzada para 16 tipos diferentes de topologia	51
Figura 5.7. Treinamento da ANN para 4 topologias diferentes. As curvas em azul representam o MSE do treinamento, e as curvas em vermelho o MSE da validação.....	52
Figura 5.8 Matriz de Confusão de 4 topologias diferentes de ANN.....	52
Figura 5.9. Método de classificação de objetos usando uma ANN 66-16-4.....	54
Figura 5.10 Visão geral do método de classificação de tipos de cruzamentos.....	56
Figura 5.11 Classificação de tipos de vias quanto à forma.....	57
Figura 5.12. Saída desejada de um detector de interseções. A mudança de direção do veículo não interfere na detecção da interseção.....	57
Figura 5.13 Imagens de locais onde foram coletados os dados.....	59
Figura 5.14 Exemplos de interseções de vias obtidas com sensor laser	60
Figura 5.15 Resultado do classificador AdaBoost.....	61
Figura 5.16 Resultado do método de identificação de interseções de vias.....	63
Figura 6.1 Mapa viário extraído do Google Maps. Os círculos representam as interseções de vias e os quadrados os pontos terminais. Ambos são informações importantes para a confecção do mapa topológico.....	66
Figura 6.2 Representação das diferentes origens e destinos para o cruzamento.....	66
Figura 6.3 - Descolamento angular entre duas nuvens de pontos 2D, tendo como referência o eixo A-B	67
Figura 6.4 Vista superior de uma nuvem de pontos 3D (unidade em metros).....	68
Figura 6.5 Imagem em preto e branco obtida da nuvem de pontos da figura 6.4.....	69
Figura 6.6 Pontos de corner referentes à figura 6.5.....	69
Figura 6.7 Trajeto realizado pelo veículo durante o primeiro teste do sistema de localização.....	71
Figura 6.8 Mudança de direção do veículo no teste 1.....	72

Figura 6.9 Trajeto realizado pelo veículo durante o segundo teste do sistema de localização.....	72
Figura 6.10 Mudança de direção do veículo no teste 2.....	73
Figura 6.11 Trajeto realizado pelo veículo durante o terceiro teste do sistema de localização.....	74
Figura 6.12 Mudança de direção do veículo no teste 3.....	74

LISTA DE TABELAS

Tabela 2.1: Características dos sensores utilizados no veículo autônomo que venceu a última competição DARPA <i>Urban Challenge</i>	20
Tabela 5.1: Desempenho de diferentes configurações de ANN	53
Tabela 5.2 Desempenho de classificadores obtidos com o Weka	53
Tabela 5.3 – Desempenho das melhores topologias para cada uma das situações: dados do conjunto de teste do <i>Curb Detector</i> , <i>Surface Detector</i> , <i>Curb Detector + Surface Detector</i>	56
Tabela 5.4: Conjunto de dados utilizados no método de detecção de interseções de vias.....	59
Tabela 5.5 Desempenho dos classificadores para os conjuntos de teste SC2 e K2.....	61
Tabela 5.6: Resultado do método de detecção de interseções.....	62

LISTA DE SIGLAS E ABREVIATURAS

- ANN – *Artificial Neural Networks*
- CaRINA- Carro Robótico Inteligente para Navegação Autônoma
- CAUGVT - *Committee on Army Unmanned Ground Vehicle Technology*
- CRF – *Conditional Random Fields*
- CRob – Centro de Robótica – USP São Carlos
- DARPA -*Defense Advanced Research Projects*
- ELROB - *European Land-Robot Trial*
- FSM – *Finite State Machine* (Máquina de Estados Finitos)
- GP-INSAC - *Gaussian Process Incremental Sample Consensus*
- GPS – *Global Positioning System*
- HMM – *Hidden Markov Models*
- ICMC – Instituto de Ciências Matemáticas e de Computação
- ICP - *Iterative Closest Point*
- IMU - *Inertial measurement unit*
- INCT-SEC - Instituto Nacional de Ciência e Tecnologia em Sistemas Embarcados Críticos
- K-NN - *k-nearest Neighbour*
- LIDAR - *Light Detection And Ranging*
- LRM- Laboratório de Robótica Móvel do ICMC-USP
- MDF - *Mission Data File*
- MLP - *Multi-Layer Perceptron*
- PROMETHEUS - *PROgramMme for a European Traffic of Highest Efficiency and Unprecedented Safety*
- RANSAC - *Random Sample Consensus*
- RBNN - *Radially bounded nearest neighbor graph*
- RNA - *Redes Neurais Artificiais*
- ROC - *Receiver operating characteristic*
- SVM – *Support Vector Machine*
- UKF – *Unscented Kalman Filter*
- USP – Universidade de São Paulo

SUMÁRIO

AGRADECIMENTOS	vii
RESUMO.....	ix
ABSTRACT	xi
LISTA DE ILUSTRAÇÕES.....	xiii
LISTA DE TABELAS.....	xvii
LISTA DE SIGLAS E ABREVIATURAS	xix
1 INTRODUÇÃO	1
1.1 MOTIVAÇÃO.....	2
1.1.1 APLICAÇÕES MILITARES	2
1.1.2 APLICAÇÕES CIVIS.....	3
1.2 CONTEXTUALIZAÇÃO	4
1.3 DEFINIÇÃO DO PROBLEMA E ABORDAGENS	7
1.4 OBJETIVOS.....	8
1.4.1 OBJETIVO PRINCIPAL	8
1.4.2 OBJETIVOS INTERMEDIÁRIOS.....	9
1.5 VISÃO GERAL DO SISTEMA E ORGANIZAÇÃO DO TEXTO	10
2 CONCEITOS FUNDAMENTAIS SOBRE VEÍCULOS AUTÔNOMOS	13
2.1 BREVE HISTÓRICO SOBRE VEÍCULOS AUTÔNOMOS.....	13
2.2 PRINCIPAIS SISTEMAS DE VEÍCULOS AUTÔNOMOS	14
2.3 NAVEGAÇÃO DE VEÍCULOS AUTÔNOMOS	16
2.4 LOCALIZAÇÃO DE VEÍCULOS AUTÔNOMOS	17
2.5 PRINCIPAIS SENSORES EMPREGADOS EM VEÍCULOS AUTÔNOMOS	18
2.6 SENSOR LIDAR 3D VELODYNE (SENSOR LASER 3D)	20
2.7 CONSIDERAÇÕES FINAIS	24
3 TÓPICOS DE APRENDIZADO DE MÁQUINAS E TÉCNICAS DE OTIMIZAÇÃO	25
3.1 APRENDIZADO DE MÁQUINA	25
3.2.1 APRENDIZADO SUPERVISIONADO.....	25
3.2.2 APRENDIZADO NÃO SUPERVISIONADO	32
3.3 ALGORITMOS DE SEGMENTAÇÃO DE NUVEM DE PONTOS	33
3.3.1 RANSAC.....	33
3.3.2 ITERATIVE CLOSEST POINT (ICP)	34
3.3.3 DETECÇÃO DE FEATURES EM IMAGENS	35
3.4 CONSIDERAÇÕES FINAIS.....	35

4	TRABALHOS RELACIONADOS	36
4.1	SEGMENTAÇÃO DE NUVEM DE PONTOS 3D	36
4.1.1	DETECÇÃO DOS PONTOS PERTENCENTES AO CHÃO	37
4.1.2	SEGMENTAÇÃO DOS PONTOS NÃO PERTENCENTES AO CHÃO	41
4.2	CLASSIFICAÇÃO DE OBJETOS EM NUVENS DE PONTOS 3D	42
4.3	IDENTIFICAÇÃO E CLASSIFICAÇÃO DE INTERSEÇÕES DE VIAS	43
4.4	LOCALIZAÇÃO DE VEÍCULOS AUTÔNOMOS	44
4.5	CONSIDERAÇÕES FINAIS	46
5	IDENTIFICAÇÃO DE ESTRUTURAS NAS PROXIMIDADES DO VEÍCULO	48
5.1	SEGMENTAÇÃO DA NUVEM DE PONTOS	48
5.2	CLASSIFICAÇÃO DE OBSTÁCULOS	49
5.3	DETECÇÃO E CLASSIFICAÇÃO DE INTERSEÇÕES DE VIAS	55
5.4	CONSIDERAÇÕES FINAIS DO CAPÍTULO	65
6	LOCALIZAÇÃO BASEADA NA DETECÇÃO DE INTERSEÇÃO DE VIAS 67	
6.1	CONFECÇÃO DO MAPA TOPOLÓGICO	67
6.2	MÉTODO PARA IDENTIFICAR MUDANÇA DE DIREÇÃO DO VEÍCULO	69
6.3	LOCALIZAÇÃO.....	72
7	DISCUSSÕES E CONCLUSÃO	78
7.1	CONTRIBUIÇÕES	78
7.2	PROBLEMAS E MELHORIAS DO SISTEMA DE IDENTIFICAÇÃO DE OBSTÁCULOS E ESTRUTURAS.....	78
7.3	PROBLEMAS E MELHORIAS DO SISTEMA DE LOCALIZAÇÃO	79
7.4	COMENTÁRIOS FINAIS	80
	REFERÊNCIAS BIBLIOGRÁFICAS	81

1 INTRODUÇÃO

Veículos autônomos terrestres podem ajudar a reduzir acidentes de trânsito e permitem uma maior mobilidade para as pessoas com necessidades especiais. Há, ainda, menos consumo de combustível e geração de poluição, graças a uma condução mais otimizada (Luettel et al, 2012).

O uso de veículos autônomos pode também ser muito útil na área de transportes de cargas, no plantio e colheita de grãos, no espargimento de defensivos agrícolas (Freitas et al, 2012). Com propósitos militares, esses veículos podem ajudar a diminuir o número de baixas, especialmente em missões de suprimento e reconhecimento.

Conforme Thrun et al (2006), os principais sistemas de um veículo autônomo são: sensoriamento, percepção, planejamento e controle do veículo, interface com o usuário, interface com o veículo e serviços globais. Os sistemas de interesse desse trabalho são os dois primeiros. Basicamente, o sistema de sensoriamento é o responsável por receber os dados do ambiente e fazer um tratamento inicial dessas informações. O sistema de percepção lida com a criação de mapas e a localização do veículo em relação a este.

Nesta tese o objetivo é identificar as principais estruturas ao redor do veículo e determinar sua localização, ou seja, determinar a posição do veículo em relação a um mapa topológico do ambiente.

Um sensor laser 3D (LIDAR) é empregado para coletar as informações do ambiente. Realiza-se, então, a segmentação das principais estruturas presentes na nuvem de pontos adquirida, os objetos e estruturas na área de alcance do sensor são classificados e identificam-se áreas livres para a navegação e características do ambiente (interseção de vias, postes, pedestres, carros, árvores).

A identificação de interseção de vias e o método desenvolvido para verificar qual direção o veículo toma ao passar pelos cruzamentos (frente, direita, esquerda) são partes fundamentais no sistema de localização apresentado neste trabalho.

Por meio de um mapa topológico, o veículo tem condições de seguir sua rota sem a necessidade de uma localização muito precisa e da criação prévia de um mapa global métrico muito detalhado. De posse de um mapa topológico global e do conhecimento sobre o posicionamento dos obstáculos, o veículo tem condições de realizar sua

navegação. Este sistema tem um grande diferencial pelo não uso do GPS e pelo uso de um único sensor para realizar todas essas tarefas.

Os algoritmos desenvolvidos foram avaliados em um ambiente tipicamente urbano, formado por ruas, avenidas, quadras e interseções. Não existe uma restrição de emprego desses algoritmos em estradas, em áreas rurais ou em minas subterrâneas. Contudo, é preciso realizar testes específicos para avaliar o desempenho do sistema nessas situações.

1.1 *MOTIVAÇÃO*

Veículos autônomos ou subsistemas de veículos autônomos têm uma vasta aplicação em operações militares ou em ambientes civis. A sua adoção permite criar aplicações buscando sempre o aumento da segurança, a facilidade de uso pelos usuários, a melhoria no desempenho do transporte de cargas e passageiros e a redução de custos e impactos ambientais.

Nesta tese, uma das importantes motivações foi a busca de uma abordagem capaz de não depender de sinais externos ao veículo, como o uso do GPS que necessita do sinal de satélites. Foi utilizado um único sensor, LIDAR 3D, instalado na parte superior do veículo. Esta motivação leva em consideração restrições impostas principalmente por aplicações militares, que não devem depender exclusivamente do sinal de GPS.

1.1.1 *APLICAÇÕES MILITARES*

Desde a Guerra do Golfo, o Exército Americano sentiu necessidade de transformar suas tropas, caracterizadas por veículos e armamentos pesados e de grande poder de fogo, para uma força mais leve, com maior capacidade de resposta e, ao mesmo tempo, mais letal e mais resistente. Em 2002, o comitê de estudos de veículos terrestres autônomos, ligado ao Departamento de Ciência Tecnologia do Exército Americano (CAUGVT, 2002), lançou um documento com conceitos técnicos e de emprego sobre sistemas de combate do futuro para o Exército (*Army's Future Combat System – FCS*), incluindo importantes questões sobre sistemas não tripulados, aéreos e terrestres.

De acordo com o comitê de estudos de veículos autônomos da Marinha americana (*Naval Operations*, 2005), veículos terrestres não tripulados têm um grande potencial de emprego em inúmeras missões militares, dando grande suporte às

operações de combate dos fuzileiros navais, bem como em apoio às operações logísticas.

A agência de pesquisas de defesa americana DARPA, desde 2002, estimula universidades, colégios e empresas, de dentro e de fora dos EUA, a desenvolver veículos autônomos. Um evento de grande importância para o desenvolvimento de veículos autônomos dentro do contexto da robótica móvel foi a competição organizada pelo DARPA (Buehler et al, 2009), na qual os participantes desenvolveram veículos que deveriam ser capazes de percorrer grandes distâncias. Em sua primeira edição, no ano de 2004, o prêmio para quem conseguisse percorrer o percurso em menor tempo era de US\$ 1.000.000,00. Foram inscritas 106 equipes, das quais 25 foram selecionadas para a competição final. A melhor equipe conseguiu percorrer apenas 12 km dos 240 propostos. Esses números caracterizam claramente a dificuldade do problema em questão. Em 2007, ocorreu o desafio urbano, em que participaram 86 equipes oriundas de oito países. Esses eventos reuniram as principais equipes de pesquisa sobre veículos autônomos em todo o mundo.

Diversos equipamentos militares podem também ser produzidos a partir de subsistemas de veículos autônomos. Um bom exemplo seria um sistema de armas automático capaz de realizar rastreamento de alvos, segmentação de imagens e classificação de objetos, principalmente em áreas de baixa visibilidade em decorrência de neblina, fumaça, poeira ou em ambientes sem nenhuma luminosidade.

A variedade de missões de caráter militar desempenhadas por veículos não tripulados é muito considerável, com perspectivas de expansão em um curto prazo, quer na diversidade quer no desempenho. Tal afirmação resulta da constatação do enorme potencial dos sistemas baseados em veículos autônomos para aumentar a capacidade operacional militar das forças, sobretudo no atual contexto de crescente assimetria das ameaças (PEREIRA, 2005).

O estudo de Amarante (2010) afirma que a alta letalidade do futuro campo de batalha pode inibir a participação de soldados, e a tecnologia pode oferecer “humanoides” robóticos para substituir os infantis. Esses humanoides poderiam operar com armamentos e veículos terrestres robóticos autônomos. A crescente complexidade das novas armas, no entanto, irá demandar operadores com consideráveis habilidades técnicas e o treinamento necessário tenderá a ser mais caro.

1.1.2 APLICAÇÕES CIVIS

Um dos principais objetivos da pesquisa realizada na área de veículos autônomos e inteligentes é diminuir o número de acidentes de trânsito. Outra motivação relevante para o desenvolvimento desse tipo de sistema é facilitar a mobilidade para pessoas com necessidades especiais.

Mais de 1,2 milhão de pessoas perdem suas vidas em decorrência de acidentes de trânsito anualmente em todo o planeta e outras 50 milhões ficam feridas nesses acidentes (Shinzato, 2015). Nos Estados Unidos, 60% dos acidentes de tráfegos fatais são decorrentes de falhas humanas, desatenção ou embriaguez ao volante (Nasaw, 2012).

Em muitos países do mundo, especialmente graças aos avanços da medicina nas últimas décadas, o número de idosos tem aumentado consideravelmente. É comum no meio desse grupo existir pessoas com reduzida capacidade de mobilidade e com baixa acuidade visual. Nesse contexto, os veículos autônomos podem ajudar na locomoção dessas pessoas, dando a elas mais independência, podendo inclusive ser um fator importante na elevação da autoestima.

Veículos autônomos têm também aplicações em outras áreas de grande interesse, como o transporte de carga e a automatização de máquinas agrícolas. A aplicação desse tipo de tecnologia nas estradas e no campo, seja no plantio, manutenção e colheita, ou seja no transporte de produtos industriais ou agrícolas, tem grande potencial para aumentar a produtividade no campo (Freitas et al, 2012). Além disso, veículos autônomos podem ser úteis em situações inóspitas ao ser humano como no combate a incêndios e no transporte de cargas em minas.

1.2 CONTEXTUALIZAÇÃO

Muitos grupos de pesquisa na área de veículos autônomos, dentre os quais destacamos o Google Car (Harris (2015) e Guizzo (2011)), Universidade de Stanford (Thrun et al, 2006), Universidade Carnegie Mellon (Ursom et al, 2008), têm utilizado diversos tipos de sensores para compreender toda riqueza de informação presente no ambiente (obstáculos móveis e fixos, sinalizações e área navegável) e para obter sua posição em relação a um mapa. Os principais tipos de sensores empregados são câmeras de vídeo, GPS, sensores inerciais, laser e radar. Cada um desses tipos de sensores possui vantagens, mas também limitações. No entanto, quando usados em conjunto, é possível extrair informações detalhadas da área em que se encontra o robô móvel a fim de possibilitar a sua navegação.

Buscando apresentar soluções menos dispendiosas ou propiciar a navegação em situações em que alguns dos sensores estejam danificados ou limitados por condições do ambiente (p. ex. a perda de sinal GPS próximo a altos edifícios ou a baixa luminosidade), algumas pesquisas têm utilizado um número reduzido de tipos de sensores. Nesses casos, o detalhamento dos mapas locais e a precisão da localização em relação a um mapa global são inferiores quando comparados com pesquisas que utilizam multi-sensores.

O GPS é um dos tipos de sensores mais utilizados em veículos não tripulados. Ele é muito preciso e útil para fazer a localização do veículo. Contudo, o seu sinal pode ser prejudicado quando se trafega em áreas densas em vegetação ou com muitos edifícios. Além disso, podem existir ações humanas intencionais no intuito de prejudicar a informação oriunda deste sistema.

De acordo com Adams (2000), desde a década de 70, os materiais militares dos EUA, incluindo o GPS, passaram predominantemente a ser desenvolvidos por empresas civis. Como algumas dessas não são americanas e muitas delas contam com colaboradores de várias partes do mundo, os riscos de sabotagem são elevados. Borio (2013) mostra que os receptores de GPS são extremamente vulneráveis a ação de *jammers*, que são aparelhos que emitem uma frequência eletromagnética com intuito de interromper a comunicação entre o receptor e as antenas de GPS. Os *jammers* podem ser facilmente adquiridos pela internet a um preço irrisório e têm sido muito utilizados por bandidos que roubam veículos monitorados.

Devido à vulnerabilidade do GPS, especialmente em ações militares, este trabalho passou a ser conduzido no sentido de desenvolver um sistema em apoio à navegação autônoma sem fazer uso do GPS, ou seja, sem depender de sinais oriundos de fontes externas (p.ex. sinais de satélite usados pelos receptores de GPS).

Basicamente, os trabalhos que não usam GPS, fazem uso de sensores inerciais, odometria e câmeras ou laser. Bayerl e Wuensche (2014) obtêm a localização do veículo em áreas rurais por meio de identificação de interseção de vias. Para isso eles fundem os dados de sensores laser, câmera estéreo e plataforma inercial. Uma grande vantagem desse trabalho é que não há necessidade de se criar um mapa digital prévio do ambiente a ser explorado. Contudo, ele usa um mapa georreferenciado, que contém muitas informações sobre o ambiente.

Badino et al (2011), utiliza apenas uma câmera de vídeo para fazer a localização do veículo. Marcos referenciais como cruzamentos, árvores e postes são identificados e

são comparados com um mapa topométrico, que armazena os tipos de marcos e as distâncias entre eles. A desvantagem desse método é que existe a necessidade de se criar um mapa topométrico a priori. Além disso, ao usar apenas câmera de vídeo, o sistema fica comprometido quando há mudanças de luminosidade no ambiente.

Neste trabalho de doutorado é apresentado um sistema de localização, assim como em Bayerl e Wuensche (2014), baseado na identificação de interseções de vias. Contudo, nesta tese utiliza-se apenas um único sensor: sensor laser 3D de múltiplos feixes. Mais especificamente, foi adotado o sensor do tipo Velodyne HDL-32 para validar a maioria dos algoritmos desenvolvidos nesta tese, os quais não necessitam de um mapa digital prévio do ambiente a ser explorado. Dessa forma, utiliza-se um mapa topológico (grafo representando nodos e suas conexões), contendo as interseções de vias e as vias associadas a essas, construído a partir de um mapa rodoviário (ou esboço de um mapa com limitada precisão na descrição do mesmo). Além da localização do veículo baseado em um mapa topológico, são identificadas as principais estruturas de um ambiente urbano rodoviário, tais como, guias, área navegável, veículos, pedestres, construções, postes e árvores. De posse dessas informações (localização global e identificação de objetos ao redor do veículo), o veículo pode realizar a navegação.

O Laboratório de Robótica Móvel (LRM) do ICMC/USP realiza estudos, implementações e análises de algoritmos inteligentes para o controle de robôs móveis autônomos. O laboratório conta com robôs e sensores para o desenvolvimento e teste dos algoritmos que se pretende desenvolver nesse trabalho. Dentre os robôs do laboratório, destacam-se os veículos CaRINA 1 (fig. 1.1) e CaRINA 2 (fig. 1.2) do projeto CaRINA¹ (Fernandes et al, 2014). Esses veículos são dotados de sistemas de percepção (LIDAR, GPS, bússola, radar, unidade inercial, câmeras) e de sistemas de atuação para o controle autônomo. Eles foram utilizados nas pesquisas e testes para validar os algoritmos desenvolvidos para a segmentação da nuvem de pontos, classificação de objetos e navegação autônoma.

¹ Projeto CaRINA (Carro Robótico Inteligente para Navegação Autônoma): <http://www.lrm.icmc.usp.br/>

Figura 1.1. Veículo CaRINA 1 – em destaque o sensor Velodyne

Figura 1.2. Veículo CaRINA 2 equipado com sensores: câmera LadyBug (1), sensor laser Velodyne (2), GPS/IMU Xsens MTI-G (3) e sensor laser Sick LMS-291 (4)

1.3 DEFINIÇÃO DO PROBLEMA E ABORDAGENS

Grande parte das pesquisas na área de veículos autônomos utilizam vários tipos de sensores, incluindo o GPS, que é um sensor vulnerável em certas situações. As

pesquisas que não utilizam GPS geralmente continuam empregando vários tipos de sensores. Há pesquisas em que se empregam somente câmeras de vídeo. Entretanto, o uso exclusivo deste tipo de sensor, ou mesmo em conjunto com outros em que atua como sensor principal, é ainda um grande desafio devido aos problemas relacionados com as variações de luminosidade.

Neste trabalho são definidas algumas restrições em relação ao sistema autônomo a ser tratado:

- (i) Não é considerado o sinal de GPS, uma vez que se deseja evitar a dependência de tal sinal;
- (ii) Não são adotados métodos que requeiram um mapa global detalhado do ambiente, usualmente obtidos por um detalhado mapeamento prévio;
- (iii) Não são utilizados métodos que requeiram uma localização muito precisa do veículo para que este possa se localizar e navegar;
- (iv) Busca-se uma solução baseada em um número reduzido de sensores, e se possível baseada apenas em um LIDAR de múltiplos feixes;
- (v) Busca-se o uso de mapas topológicos sem grande precisão, considerando a localização do veículo no mapa;
- (vi) Considera-se um ambiente composto por vias, tipicamente um ambiente estruturado composto por estradas, avenidas, rodovias e de características urbanas.

Nesta tese, utilizando-se apenas um sensor laser tridimensional para coletar os dados do ambiente, é apresentado um sistema de localização para ambientes urbanos que independe de GPS e de um mapa digital prévio com informações detalhadas sobre o ambiente. A localização será feita baseada no reconhecimento dos cruzamentos da via e identificação da direção percorrida pelo carro após passar por um cruzamento. Um mapa topológico é utilizado nesse sistema de localização. Além disso, são identificados os principais pontos de interesse e obstáculos do ambiente a fim de auxiliar na navegação.

1.4 OBJETIVOS

1.4.1 OBJETIVO PRINCIPAL

O objetivo principal deste trabalho é tratar os sinais obtidos de um sensor tipo *Lidar* de múltiplos feixes, identificando as principais estruturas e objetos ao redor do veículo, e desenvolver um sistema de localização baseado em uma mapa topológico, a fim de permitir a realização da navegação do veículo Carina 2 de modo autônomo em um ambiente urbano controlado.

1.4.2 OBJETIVOS INTERMEDIÁRIOS

Este projeto foi desenvolvido visando alcançar o objetivo principal descrito na subseção precedente, sendo composto pelos seguintes objetivos intermediários:

- (i) Aperfeiçoar e adaptar ferramentas para a aquisição, tratamento e análise dos dados de sensores do tipo LIDAR 3D (sensor laser do tipo Velodyne HDL32 e Velodyne HDL 64);
- (ii) Aperfeiçoar e adaptar técnicas para a segmentação dos elementos de uma cena 3D (nuvem de pontos), identificando, separando e marcando os elementos da cena, como por exemplo: piso (separando a parte da rua da parte da calçada), árvores, prédios, carros, pessoas e outros elementos presentes no ambiente;
- (iii) Propor, aperfeiçoar e adaptar ferramentas para o reconhecimento de elementos específicos de uma cena, como por exemplo: zona navegável da rua (espaço livre disponível para a navegação do veículo onde este pode passar sem problemas, e que define a rua), cruzamentos entre ruas, becos sem saída, e eventualmente rotatórias, curvas e outros tipos de configurações de uma via pública;
- (iv) Propor, aperfeiçoar e adaptar ferramentas para a classificação dos elementos da cena, identificando elementos estáticos (postes, prédios, árvores, placas de sinalização) e elementos dinâmicos (pessoas e veículos);
- (v) Propor e desenvolver técnicas de localização em ambientes urbanos (vias públicas), sem o uso de mapas detalhados do ambiente que tenham sido previamente fornecidos, sem o uso de informações precisas sobre a localização do veículo (prevê-se apenas o uso de uma localização aproximada baseada no reconhecimento de cruzamentos, sem uso de GPS). Deste modo, busca-se uma localização baseada exclusivamente em informações provenientes de nuvem de pontos 3D. A localização pode ser imprecisa, pois entende-se que a navegação possa ser realizada baseada

nas informações locais, fazendo com que o veículo se mantenha dentro da pista, e em um mapa global topológico, possibilitando que o veículo siga em direção ao seu destino.

Os objetivos descritos acima compõem as principais contribuições desta tese, que visa propor, desenvolver e avaliar um sistema de localização de veículos autônomos baseado em mapas topológicos, usando um sensor laser e sem depender de informações de GPS. Visa, ainda, detectar e identificar os obstáculos presentes no ambiente a fim de possibilitar a navegação de um veículo autônomo em um ambiente estruturado composto por vias e intersecções. Assim, esta tese contribui de forma original uma vez que define um sistema diferenciado e caracterizado pelos componentes e abordagens descritos acima.

1.5 *VISÃO GERAL DO SISTEMA E ORGANIZAÇÃO DO TEXTO*

A figura 1.3 apresenta a visão geral do sistema desenvolvido nesta tese. Os dados do ambiente são coletados por meio de um sensor laser 3D. O módulo de segmentação I separa os dados em dois grupos: pontos do chão e pontos de não-chão. A identificação da área navegável é obtida pelo processamento dos pontos pertencentes ao chão.

Os pontos de não chão são segmentados (Segmentação II) e em seguida cada segmento é classificado em carros, pedestres, construção, postes e troncos de árvores.

Os pontos de não-chão são ainda usados para realizar a detecção de intersecções de vias urbanas. O resultado dessa detecção em conjunto com os dados brutos de não-chão são usados no sistema de localização.

Para que o leitor tenha uma compreensão melhor do sistema desenvolvido, o texto foi dividido da seguinte forma:

A seção 2 apresenta alguns conceitos fundamentais sobre veículos autônomos, focando suas principais funcionalidades e os principais sensores utilizados. Especial atenção é dada ao sensor laser 3D, Velodyne 32-HDL.

Conceitos importantes de aprendizado de máquinas voltados para aplicações em veículos autônomos e algoritmos de otimização são apresentadas na seção 3. Na seção 4 são apresentados os principais trabalhos científicos relacionados com essa pesquisa.

As seções 5 e 6 trazem as principais contribuições desta tese. Na seção 5 são apresentados os métodos de identificação das principais estruturas presentes no ambiente de um veículo não tripulado, como pedestres, veículos, construções, árvores, postes, tipos de interseções e geometria da pista. Na seção 6, o destaque é o método de localização desenvolvido utilizando apenas um sensor laser e um mapa topológico.

A seção 7 traz uma discussão sobre os resultados obtidos, contribuições desta tese, trabalhos futuros e uma breve conclusão.

Figura 1.3: Visão geral do sistema

2 CONCEITOS FUNDAMENTAIS SOBRE VEÍCULOS AUTÔNOMOS

Este capítulo apresenta um breve histórico sobre veículos autônomos, suas principais funcionalidades e os principais tipos de sensores utilizados no auxílio de sua navegação.

Veículo autônomo é aquele que pode se locomover de um ponto a outro sem auxílio de controle humano local ou remoto. Os principais sistemas desse veículo são o Sensoriamento, Percepção, Planejamento e Controle, Interface com o Usuário e Interface com o Veículo.

2.1 BREVE HISTÓRICO SOBRE VEÍCULOS AUTÔNOMOS

Desde a década de 50 de forma bastante rudimentar e a partir dos anos 70 e 80 de modo mais abrangente, muitos estudos sobre veículos autônomos foram realizados. Contudo, os avanços ganharam força após a iniciativa europeia em investir nessa tecnologia, criando o Projeto PROMETHEUS - EUREKA *Prometheus Project*: "PROgramme for a European Traffic of Highest Efficiency and Unprecedented Safety" – (Williams, 1992) e com as competições americanas realizadas pelo DARPA.

O Projeto PROMETHEUS ocorreu entre os anos de 1987 e 1995. Foram investidos mais de 800 milhões de euros em tecnologias ligadas a veículos autônomos. Um dos resultados de destaque deste projeto foi alcançado em 1995, quando um veículo projetado por Ernst Dickmanns fez o percurso de ida e volta de Munique a Copenhague, alcançando uma velocidade máxima de 175 km/h. O veículo foi capaz de realizar ultrapassagens sem auxílio humano e conseguiu percorrer um trecho de 158 km sem qualquer intervenção humana. No entanto, o sistema não era completamente eficiente, pois foram necessárias, em alguns casos, ações do motorista no sentido de evitar colisões.

Muitas outras iniciativas isoladas continuaram na tentativa de desenvolver veículos autônomos, mas elas só começaram a ganhar grande visibilidade em todo o mundo quando a agência americana DARPA começou a estimular pesquisas nessa área. Desde 2002, o DARPA estimula universidades, colégios e empresas, de dentro e fora dos EUA, a desenvolver veículos autônomos. Em 2005, foi realizado o DARPA *Grand Challenge* em uma área desértica. Em 2007, ocorreu o DARPA *Urban Challenge* (Buehler, 2009), em que participaram 86 equipes oriundas de oito países. Esse evento congregou as principais equipes de pesquisa sobre veículos autônomos em todo o mundo.

Outro evento que estimula pesquisas na área de veículos não tripulados é o *European Land-Robot Trial* – ELROB – (Schneider et al, 2012). O ELROB não é uma competição como aquelas promovidas pelo DARPA. Trata-se de uma demonstração daquilo que as equipes de robótica da Europa são capazes de desenvolver. Os terrenos para a navegação são escolhidos para simular um emprego real para aplicações civis e militares. O primeiro *ELROB* foi realizado de 15 a 18 de maio de 2006 e foi organizado pelas Forças Armadas da Alemanha em uma área de treinamento para tropas de infantaria próximo à cidade de Hammelburg - Alemanha.

Além de instituições, existem muitas companhias privadas que investem em veículos autônomos. A empresa Google contratou pesquisadores renomados na área de veículos inteligentes para desenvolver um veículo autônomo, dentre eles, o professor e chefe da equipe vitoriosa nos desafios DARPA (1º lugar em 2005 e 2º lugar em 2007), Sebastian Thrun da Universidade de Stanford. Em agosto de 2012, a equipe da Google informou que eles atingiram uma marca de mais de 500 mil quilômetros rodados com seus veículos autônomos (Google Driverless Car, 2013). Informações mais detalhadas desse projeto são difíceis de serem obtidas, pois elas são mantidas em sigilo pela empresa. Por meio de alguns vídeos divulgados na internet que mostram o desempenho desses veículos, acredita-se que esse projeto seja o mais avançado no mundo atualmente. Várias montadoras de veículos, como a Mercedes e a Toyota, também estão trabalhando no desenvolvimento desse tipo de veículo, mas poucos detalhes técnicos são publicados. Em função disto, a maior fonte de informações sobre a pesquisa e desenvolvimento de veículos autônomos tem sido os eventos científicos (p.ex. IROS, ICRA, ITS, IV) e artigos acadêmicos publicados em periódicos.

2.2 PRINCIPAIS SISTEMAS DE VEÍCULOS AUTÔNOMOS

A construção de um veículo autônomo requer o domínio de diversas áreas do conhecimento, como visão computacional, mapeamento digital com uso de sensores laser, localização global com ou sem uso de um sistema de posicionamento global (GPS), entre muitas outras áreas.

O robô denominado Stanley, desenvolvido por pesquisadores da Universidade de Stanford, da Volkswagen (EUA), da Intel Research e da Mohr Davidow Ventures, foi o vencedor do DARPA *Grand Challenge*. Os principais sistemas deste veículo podem ser observados na figura 2.1.

Figura 2.1: Principais sistemas de um veículo autônomo (adaptado de THRUN et al, 2006.a)

O sistema de Sensoriamento possui diversos módulos de *software* responsáveis por receber os dados oriundos dos diversos sensores: laser, câmeras de vídeo, radar, GPS e *inertial measurement unit* (IMU).

O sistema de percepção estima a velocidade, a orientação e a localização do veículo em relação a um mapa global e identifica a área navegável e os obstáculos ao redor do veículo. Este sistema, ainda, detecta os limites de velocidade e a sinalização das estradas a fim de determinar uma velocidade de segurança para o veículo.

O sistema de Planejamento e Controle se encarrega de gerar sinais para o controle de direção, aceleração e frenagem. Além disso, este sistema identifica e atualiza a trajetória a ser realizada pelo veículo.

Por critério de segurança, caso seja necessário que o veículo pare imediatamente, o usuário pode apertar um botão de parada de emergência (*E-stop*). Com auxílio de uma interface *touch-screen* é possível inicializar o *software* do veículo autônomo. Essas duas funcionalidades são de responsabilidade do sistema de Interface com o Usuário.

O sistema de Interface com o Veículo contém todos os dispositivos mecânicos e eletrônicos para executar a frenagem, aceleração e o direcionamento das rodas. Ele, ainda, é responsável por fornecer a energia para todos os componentes do sistema.

O sistema de Serviços Globais fornece serviços básicos para todos os módulos de *software* dos outros sistemas, tais como sincronização e comunicação. Além disso, possui um servidor que armazena os dados de todos os parâmetros do veículo e os mantém atualizados.

2.3 NAVEGAÇÃO DE VEÍCULOS AUTÔNOMOS

Nesta tese não se pretende desenvolver um sistema de navegação. Contudo, alguns tópicos sobre esse assunto são necessários para se compreender melhor o sistema de localização, que é apresentado na seção 6.

A navegação de veículos autônomos é um dos problemas fundamentais da robótica móvel e pode ser entendida como a capacidade do veículo executar as ações planejadas de modo robusto, como por exemplo, realizar o deslocamento de uma posição-origem até uma posição-alvo, fazendo os devidos ajustes durante o deslocamento para evitar a colisão com obstáculos. A navegação robótica é a execução do plano de trajetórias e ações previamente definidas (WOLF et al, 2009).

Luettel et al (2012) divide a navegação em 3 principais tipos: navegação global, navegação reativa e navegação guiada.

- a) **Navegação Global:** o veículo é guiado por trajetórias geradas pelos algoritmos de planejamento de trajetórias em mapas métricos globais do ambiente. Na navegação global, o mapa separa o resto do sistema do sistema de percepção: para realizar a avaliação da situação e o planejamento de trajetória não é preciso saber quais sensores e quais algoritmos foram usados para criar o mapa. Somente as informações contidas no mapa são levadas em consideração nessas tarefas. Na navegação global o mapa precisa ser criado e sua consistência precisa ser mantida ao longo do tempo.
- b) **Navegação Reativa:** neste tipo de navegação não é necessário conhecer o mapa do ambiente e nem fazer qualquer tipo de planejamento de trajetória. Ela é muito utilizada quando se deseja, por exemplo, realizar tarefas como desviar de um obstáculo ou seguir um objeto a frente. As arquiteturas reativas são geralmente utilizadas como uma camada de nível inferior na navegação de robôs móveis, com a vantagem de responderem em tempo real, por mapearem diretamente a leitura dos sensores em ações do robô (Fracasso & Reali Costa, 2005).
- c) **Navegação Guiada:** este tipo de navegação pode ser entendido como um meio termo entre a navegação global e a reativa. O planejamento da trajetória não é feito baseado em um mapa preciso do ambiente e o seu movimento não é regido com a simplicidade de apenas se evitar uma colisão ou seguir um objeto a frente. Um bom exemplo de uma

navegação guiada é o caso de um veículo em área urbana, obtendo sua localização por meio de *landmarks* ou cruzamentos, mantendo-se dentro dos limites da pista e desviando dos obstáculos que se apresentam. O planejamento da trajetória nesse caso é realizado, mas ele não é baseado em mapas detalhados do ambiente. O planejamento pode ser baseado em um mapa topológico. Comportamentos do tipo “vire à direita no terceiro cruzamento” podem ser planejados se o sistema de percepção conseguir identificar as principais estruturas do ambiente para o reconhecimento das ruas e cruzamentos.

Muitos trabalhos sobre navegação de veículos autônomos estão focados na obtenção da localização precisa do veículo em relação a um mapa. Este mapa pode ser gerado previamente ou no mesmo instante em que o veículo se locomove. No geral, vários sensores são utilizados para a criação dos mapas e para realizar a localização. A técnica *Simultaneous Localization and Mapping* – SLAM (Thrun et al, 2006.b) é uma das mais conhecidas na área da robótica móvel e tem sido largamente empregada em virtude de construir um mapa do ambiente geometricamente consistente ao mesmo tempo em que o sistema realiza a localização. Entretanto, a técnica SLAM é bastante custosa computacionalmente e, também, possui restrições quanto ao seu emprego em ambientes dinâmicos ou sujeitos a muitas alterações em sua configuração.

2.4 LOCALIZAÇÃO DE VEÍCULOS AUTÔNOMOS

A localização de um robô móvel consiste em determinar a posição do robô em relação a um dado mapa do ambiente (Thrun et al, 2006.b). O robô recebe um mapa do ambiente e seu objetivo é determinar sua posição em relação a esse mapa usando seus sensores para perceber seus movimentos e o ambiente.

Normalmente são empregados três tipos de mapas: métrico, topológico e topométrico (Dudek & Jenkin, 2010). O primeiro normalmente é dividido em pequenas células e cada uma dessas possui uma referência métrica ou geográfica (latitude, longitude, altitude). O mapa topológico apresenta alguns elementos que podem ser bem caracterizados no ambiente (*landmarks*), como árvores, construções, guias e interseções de vias. Normalmente, esses elementos são representados no mapa por grafos, contendo vértices, os quais são conectados entre si por arestas buscando caracterizar a relação de vizinhança desses *landmarks*. O mapa topométrico é uma melhoria de um topológico.

Ele agrega a informação de distância entre os vértices, onde as arestas possuem uma informação métrica (grafo com arestas valoradas).

Thrun et al (2006) divide a localização de robôs móveis em três tipos :

- *Position tracking*: assume que a posição inicial do robô é conhecida.
- *Localização global*: a posição inicial do robô não é conhecida. O robô é inserido em algum lugar do ambiente e deve ser capaz de obter sua localização enquanto navega pelo ambiente.
- *Problema do robô sequestrado*: o robô é retirado da sua posição e colocado em outra. Este é um problema mais difícil do que o da posição global, pois o robô pode acreditar que tem sua posição conhecida após o “sequestro”.

Boa parte dos trabalhos na área de veículos autônomos tem considerado um mapa métrico e a localização global. Nessa abordagem, o GPS desempenha um papel fundamental, dado a grande precisão de sua resposta. Como esta tese objetiva usar um número reduzido de sensores, exceto o GPS, a localização global não será abordada. Tampouco serão utilizados mapas métricos. A localização será do tipo *position tracking* baseado em um mapa topológico. Assim, na revisão da literatura sobre localização, apresentada na subseção 4.5, apenas trabalhos que utilizam uma abordagem como essa é que serão considerados.

2.5 PRINCIPAIS SENSORES EMPREGADOS EM VEÍCULOS AUTÔNOMOS

Em virtude da grande versatilidade que deve possuir um veículo autônomo, é extremamente difícil conseguir utilizar um único sensor para detectar todas as variáveis do ambiente. Todos os projetos nessa área que foram analisados utilizam a integração de dados de mais de um sensor.

A equipe *Spirit of Berlin* da *Frei Universität Berlin* (Rojas, 2005) também participou do *DARPA Urban Challenge* em 2007. Eles utilizaram um sensor laser instalado no para-choque dianteiro do carro a fim de detectar os obstáculos que se encontrem à frente do veículo. Sobre o teto do automóvel, encontra-se uma antena de um sistema de posicionamento global (*Global Positioning System* – GPS), dois sensores *laser* fixados em uma base rotativa e uma câmera onidirecional (Figura 2.2).

Figura 2.2: Sensores utilizados pela equipe *Spirit of Berlin* (ROJAS, 2005).

O módulo GPS/IMU é um dispositivo que fornece as coordenadas em que se encontra o veículo, com auxílio do GPS e a velocidade e orientação, por meio de acelerômetros e giroscópios do módulo IMU.

O sensor laser de varredura (*Light Detection And Ranging* - LIDAR) utilizado pela equipe *Spirit of Berlin* é o modelo Lux da empresa IBEO. Este dispositivo tem um alcance de 200m, uma varredura de 85° no plano horizontal e de $3,2^\circ$ no plano vertical. Ele consegue varrer quatro faixas (planos), possibilitando obter a distância e altura do obstáculo (Figura 2.3). Em geral, este tipo de sensor é utilizado para detectar os obstáculos que se encontram a médias e longas distâncias, à frente do veículo autônomo. Com o uso do laser, não há prejuízo na coleta de informações do ambiente em caso de pouca ou nenhuma luminosidade.

Figura 2.3: Representação da varredura vertical do sensor LUX IBEO - utilizados pela equipe *Spirit of Berlin* (Rojas, 2005)

As câmeras de vídeo podem também ser utilizadas para detectar faixas da pista, veículos, pessoas e outros obstáculos. Elas são essenciais na identificação de sinais de

trânsitos. O uso de monovisão tem dificuldade em estimar com precisão a distância dos diversos obstáculos. Alguns trabalhos utilizam visão estéreo para aumentar a eficiência na estimação de distâncias. Contudo, isso aumenta muito o tempo de processamento de imagens. Em relação ao alcance, a câmera pode ser utilizada em várias faixas de distância, chegando usualmente até um máximo de 60 m, porém esta distância pode variar de acordo com a resolução e os parâmetros de ajuste das câmeras.

Em URMSON *et al.* (2008) podem ser observados os sensores utilizados no veículo “Boss”, vencedor do DARPA *Urban Challenge* 2007. Utiliza-se uma combinação de dezoito sensores para realizar a percepção do ambiente, garantindo a segurança pela redundância de diversos dispositivos. Os sensores ativos (laser e radar) são os mais empregados nesse trabalho em virtude da grande precisão. As características e as posições dos sensores empregados no veículo *Boss* podem ser acompanhadas pela Tabela 2.1 e pela Figura 2.4.

Figura 2.4 Sensores usados pela equipe vencedora do DARPA *Urban Challenge* (URSON *et al.*, 2008).

2.6 SENSOR LIDAR 3D VELODYNE (SENSOR LASER 3D)

Apesar do entendimento de que são necessários vários tipos de sensores para que o veículo autônomo possa extrair todas as informações importantes do ambiente, neste trabalho é utilizado apenas um único sensor laser 3D. O motivo disso é explorar ao máximo toda a capacidade desse tipo de sensor, bem como demonstrar que com apenas este único sensor é possível realizar a navegação do veículo em situações em que não se tem disponibilidade de usar outros sensores, seja por condições ambientais (prédios altos que limitam o sinal GPS ou condições de baixa luminosidade que prejudicam as imagens das câmeras), ou pelo mau funcionamento de um dos sensores ocasionado de forma acidental ou intencional.

Tabela 2.1: Características dos sensores utilizados no veículo autônomo que venceu a última competição DARPA *Urban Challenge*.

Tipo	Modelo/ Marca	Qtd	Alcance Máx.	Abertura Angular Máx.	Utilização
Laser	Sick LMS 291	06	80 m	180°	Deteccção de obstáculos frontais, traseiros e laterais
	Velodyne HDL-64	01	120 m	360° x 26°	Mapeamento tridimensional do ambiente
	Continental ISF 172	02	150 m	14°	Deteccção de obstáculos frontais
	IBEO Alasca XT	02	300 m	240° x 3,2°	Deteccção de obstáculos frontais
Vídeo	Point Grey Firefly (PGF)	02	-	45°	Deteccção de obstáculos frontais e identificação de sinalização
Radar	Continental ARS 300	05	200 m	60° x 3,2°	Deteccção de obstáculos frontais, traseiros e laterais
GPS/ IMU	Applanix Pos LV (APLV)	01	-	-	Detectar o posicionamento global e aceleração das rodas

Nesta pesquisa todos os dados do ambiente são coletados por um sensor laser 3D Velodyne HDL. Assim, é importante tecer alguns comentários sobre este dispositivo para que se possa ter uma ideia melhor de suas características.

Este sensor tem sido empregado por muitos trabalhos de pesquisa ligados a veículos autônomos. No DARPA *Challenge* de 2007, as duas primeiras equipes classificadas do concurso utilizavam esse tipo de sensor. O veículo autônomo que está sendo desenvolvido pela Google também utiliza esse mesmo tipo de sensor.

O sensor laser HDL-64 Velodyne foi desenvolvido especificamente para aplicações em veículos não tripulados. Ele consegue fazer uma varredura de 360° no plano horizontal e de 26,8° (+2° a -24,8°) na vertical, através da coleta de dados de 64 feixes de varredura a laser. É capaz de captar mais de 1,3 milhões de pontos do ambiente em um segundo, com um alcance de até 120 metros e erro máximo estimado menor que 2 cm. A Figura 2.5 apresenta o modelo do Velodyne HDL64 S2.

Figura 2.5: Sensor Velodyne HDL-64 S2.

O sensor laser Velodyne HDL-32E (figura 2.6) basicamente difere-se do HDL-64 por usar metade dos feixes de laser deste. Ele possui 32 feixes de laser cobrindo uma resolução angular vertical de $+10^\circ$ a -30° , além de fazer uma varredura angular horizontal de 360° . Ele é capaz de gerar uma nuvem de pontos de aproximadamente 700 mil pontos por segundo com um alcance de até 70 m, com uma precisão de 2 cm a uma frequência de 10 Hz.

Figura 2.6: Sensor Velodyne HDL-32E

Recentemente foi lançado no mercado uma versão com menos feixes de laser, trata-se do Velodyne VLP-16 (figura 2.7). Ele tem um alcance de 100 metros, gera 300 mil pontos por segundo, cobrindo uma resolução angular vertical de $+15^\circ$ a -15° , além de fazer uma varredura angular horizontal de 360° . Apesar de ser um sensor que gera informações menos detalhadas do ambiente quando comparado com os dois outros sensores, o seu grande diferencial é seu custo reduzido em relação aos modelos anteriores, cerca de 8 mil dólares. Uma outra grande vantagem deste sensor é que ele está inserido em um receptáculo todo fechado, ficando menos sujeito a problemas causados por excesso de poeira e umidade.

Figura 2.7: Sensor Velodyne VLP-16

Para facilitar a compreensão sobre o princípio básico de funcionamento do sensor Velodyne, é importante entender primeiro o funcionamento do sensor laser Sick LMS. Este sensor, já há tempo muito conhecido na robótica móvel, opera por meio da medição do tempo de ida e volta de pulsos de laser. Um feixe de laser pulsado é emitido por um elemento emissor. Quando este sinal encontra um objeto dentro do seu raio de ação ele é refletido e registrado pelo módulo receptor do sensor. O tempo entre a transmissão e a recepção do impulso é diretamente proporcional à distância entre o Sick LMS e o objeto. Com auxílio de um espelho rotativo integrado, o laser é emitido para várias direções, mapeando uma área de até 180° em apenas um nível de altura (Figura 2.8).

O sensor Velodyne tem o mesmo princípio de funcionamento. Contudo, em vez de usar um único conjunto de emissor e receptor, ele utiliza vários. O número desses conjuntos varia com o modelo do sensor. Por exemplo, o modelo HDL-64 possui 64 conjuntos e o modelo VPL-16 possui 16 conjuntos de emissores e receptores.

Esses conjuntos de emissores e receptores são instalados em uma haste vertical e cada um deles é posicionado com um diferente ângulo em relação ao plano horizontal. Como a haste gira 360° , consegue-se obter pontos 3D do ambiente em todas as direções.

Figura 2.8: Representação da varredura de 180° do sensor laser Sick LMS-291 (manual do sensor Sick LMS).

2.7 *CONSIDERAÇÕES FINAIS*

Boa parte das equipes que desenvolvem veículos autônomos utilizam vários tipos de sensores. Os mais empregados são as câmeras de vídeo, os radares e o GPS, a IMU e os sensores laser. Cada um desses sensores possuem seus pontos positivos e negativos.

Para a realização da localização, quase a totalidade dos trabalhos usam o GPS como um sensor fundamental. Como já apresentado, em algumas situações, os sinais do GPS podem ser corrompidos ou estar indisponíveis. Nesse sentido, esta tese apresenta um sistema de localização que não utiliza sinais de GPS. Apenas um sensor laser 3D é utilizado para todas as atividades, desde a identificação de obstáculos até a localização do veículo.

3 TÓPICOS DE APRENDIZADO DE MÁQUINAS E TÉCNICAS DE OTIMIZAÇÃO

Como apresentado no capítulo 2, veículos autônomos são sistemas muito complexos. Eles precisam interpretar as variáveis do seu ambiente, identificando os obstáculos ao seu redor e a área de navegabilidade a fim de realizar seu deslocamento de forma correta e segura.

Neste trabalho, todos os dados do ambiente são oriundos de uma nuvem de pontos 3D gerada por um sensor laser (Velodyne HDL-32E e Velodyne HDL-64). É preciso extrair desses dados informações relevantes para a navegação do veículo. Em um cenário urbano, a identificação dos elementos mais comuns como pedestres, veículos, construções, guias, ruas, calçadas e cruzamentos são de extrema importância. Para tal, o uso de técnicas de aprendizado de máquina é importante para que o veículo autônomo possa reconhecer essa variedade de objetos. Esta área do conhecimento tem sido exaustivamente empregada nos últimos anos para resolver inúmeros problemas que vão desde fraudes em operações com cartões de crédito até a navegação de veículos autônomos (Mitchell, 1997).

3.1 APRENDIZADO DE MÁQUINA

Murphy (2012) define aprendizado de máquina como o conjunto de métodos que pode automaticamente detectar padrões em dados e em seguida usar padrões não analisados previamente para fazer previsões sobre os dados futuros ou para realizar outros tipos de decisões considerando a incerteza do processo. O aprendizado de máquina agrega várias áreas do conhecimento, incluindo estatística, inteligência artificial, biologia, ciência cognitiva, engenharia da computação e teoria de controle.

A maioria dos autores divide as técnicas de aprendizado de máquina em três tipos: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (Murphy, 2012). Com exceção deste último tipo que não será abordado especificamente neste trabalho, alguns tópicos sobre os demais são apresentados a seguir.

3.2.1 APRENDIZADO SUPERVISIONADO

O objetivo do aprendizado supervisionado é induzir conceitos a partir de exemplos que estão pré-classificados, ou seja, exemplos que estão rotulados com uma

classe conhecida, também chamados de conjunto de treinamento. Se as classes possuírem valores discretos, o problema é categorizado como classificação. Caso elas possuam valores contínuos, o problema é denominado de regressão.

a) Redes Neurais Artificiais

As Redes Neurais Artificiais (RNA), que são modelos clássicos de aprendizado supervisionado, são modelos matemáticos inspirados na observação da complexa rede de neurônios do cérebro humano. Essas redes podem ser empregadas com muito êxito para aprendizado de exemplos compostos de funções de valores reais ou discretos (Mitchell, 1997).

As mais destacadas características da RNA são a capacidade de se adaptar ou aprender por meio de exemplos (e.g., padrões, amostras), a generalização do conhecimento adquirido, a atuação como mapeadores de entrada e saída (aprendizado supervisionado), além da tolerância às falhas (Haykin, 1998).

Uma das RNA mais conhecidas é o *Multi-Layer Perceptron* (MLP). Este consiste em um conjunto de neurônios, dispostos em camadas organizadas em sequência. Normalmente, uma rede neural MLP possui 3 camadas (Figura 3.1). A camada de entrada, que recebe os dados a serem inseridos na rede, a camada escondida, a qual realiza o cálculo dos pesos (conexões) com os dados de entrada, e a camada de saída, que fornece os resultados (Haykin, 1998).

Cada neurônio de uma camada é usualmente conectado com todos os neurônios das camadas anteriores e posteriores, com um determinado peso. Para que os dados de entrada sejam corretamente relacionados às saídas esperadas, é necessário estimar corretamente os pesos que conectam os neurônios da rede. Esta estimativa é realizada em uma etapa de aprendizado, que consiste em usar exemplos pré-classificados e um algoritmo de treinamento.

O algoritmo de treinamento das RNA MLP mais usado nesta tarefa é o *back-propagation* que, por ser supervisionado, utiliza pares de entrada e saída por meio de um mecanismo de correção de erros, ajustando os pesos da rede. O treinamento ocorre em duas fases: *forward* e *backward*. A fase *forward* é utilizada para definir a saída da rede para um dado padrão de entrada. Já a fase *backward* utiliza a saída desejada e a saída fornecida pela rede neural para estimar o erro de resposta e assim atualizar os pesos de suas conexões (Braga et al., 2007).

Figura 3.1: Representação de uma rede neural artificial (Haykin, 1998).

b) AdaBoost

O Adaboost, criado por Freund e Schapire (1999), é um método de aprendizado de máquina baseado na ideia de criar uma regra de predição de elevada precisão por meio da combinação de várias regras de classificação de baixa precisão, chamados de classificadores fracos. O nome AdaBoost deriva de *Adaptive Boosting*. Ele é adaptável no sentido de que as classificações subsequentes feitas são ajustadas a favor das instâncias classificadas negativamente por classificações anteriores.

Este algoritmo usa uma associação de classificadores denominados fracos. Cada um destes é na verdade um classificador simples que utiliza apenas uma reta para separar o conjunto de dados em dois grupos. O algoritmo 1 apresenta os passos para se implementar um classificador AdaBoost. A resposta final é uma associação de todos os classificadores fracos.

Algoritmo 1: AdaBoost (Freund e Schapire, 1999)

-
- 1 Dados: $(x_1, y_1), \dots, (x_m, y_m)$, onde $x_i \in \mathcal{H}$, $y_i \in \{-1, +1\}$
 - 2 Inicialize: $D_1(i) = 1/m$, para $(i) = 1, 2, \dots, m$.
 - 3 **Para** $t = 1, \dots, T$
 - 4 Treine um classificador fraco usando a distribuição D_t
 - 5 Obtenha uma hipótese fraca $h_t: \mathcal{H} \rightarrow \{-1, +1\}$
 - 6 Selecione h_t com um baixo erro, $\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
 - 7 Escolha $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right)$
 - 8 Atualize, para $i = 1, \dots, m$, $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

Onde Z_t é um fator de normalização

9 Fim

10 Atualize a hipótese final $H(x) = \text{sign}(\sum_{t=1}^T \alpha_t h_t(x))$

c) Support Vector Machine

Conforme Schölkopf e Smola (2001), *Support Vector Machine* (SVM) é um método de aprendizado de máquina supervisionado utilizado para classificação. As entradas são representadas por pontos no espaço. O algoritmo tenta encontrar o melhor hiperplano capaz de separar o conjunto de dados de treinamento em duas classes. O melhor hiperplano é escolhido de forma que ele maximize as margens de ambas as classes. A margem é a distância entre o hiperplano e os elementos mais próximos a este hiperplano. Na figura 3.2, há três hiperplanos, H1, H2 e H3. O H2 e H3 conseguem separar o conjunto de treinamento, já o H1 não. Comparando H2 e H3, nota-se que a margem de H3 (W3 na fig. 3.2) é maior do que H2.

Na figura 3.3(a), a linha em vermelho representa um hiperplano linear, a margem entre as classes é indicada pelas setas e pela linha tracejada. Na letra (b) da figura 3.3, apresenta-se o caso não linear. Neste caso uma função kernel deve ser usada para criar um espaço onde as classes são separáveis por um hiperplano. Uma vez feito isso, o método torna-se muito similar ao caso linear.

Figura 3.2: Hiperplanos separando dados de treinamento. O hiperplano H3 é o que possui a maior margem.

Figura 3.3: Hiperplanos separando dados de treinamento. Em (a) tem-se o caso linear, e em (b) o caso não linear.

d) Modelos Gráficos Probabilísticos

Para um melhor entendimento de HMM e de *Conditional Random Fields* (CRF), apresentados nas subseções subsequentes, faz-se necessário uma breve introdução sobre modelos gráficos probabilísticos (*Probabilistic Graphical Models*).

Os modelos gráficos são poderosas ferramentas para a representação e inferência em distribuições probabilísticas multivariáveis. Eles têm sido muito usados em diversas áreas como visão computacional, estatísticas Bayesianas e processamento de linguagem (Sutton & McCallum, 2011).

Os modelos gráficos probabilísticos são uma representação de uma distribuição de probabilidade, na qual cada vértice do grafo é uma representação de cada variável aleatória e as arestas entre os vértices representam as relação entre essas variáveis. A ausência de uma aresta entre duas variáveis representa uma independência condicional entre essas variáveis. Independência condicional significa que duas variáveis aleatórias a e b são independentes dada uma terceira variável c , se elas são independentes em sua distribuição de probabilidade condicional. Formalmente temos: $p(a, b|c) = p(a|b)p(b|c)$.

A independência condicional é um importante conceito que pode ser usado para decompor distribuições complexas de probabilidade em um produto de fatores, com cada um desses consistindo de um subconjunto de variáveis aleatórias. Este conceito torna os complexos cálculos mais eficientes, como por exemplo, para realizar o aprendizado e a inferência. A fatoração de uma distribuição de probabilidade é escrita como o produto de seus fatores $\psi_s(\vec{v}_s)$, sendo $\vec{v}_s$ o subconjunto das respectivas variáveis aleatórias que constituem tal fator, ou seja, $p(\vec{v}) = \prod_s \psi_s(\vec{v}_s)$.

A figura 3.4 mostra um exemplo de um grafo de fatores com três variáveis aleatórias. Os círculos desta figura representam as variáveis e os quadrados representam os fatores. A distribuição de probabilidade conjunta pode ser escrita como $p(y_1, y_2, y_3) = \psi_1(y_1, y_2)\psi_2(y_2, y_3)\psi_3(y_1, y_3)$. Os fatores devem ser funções de variáveis aleatórias que geram números reais. Eles não necessariamente precisam representar probabilidades. Neste caso, para se encontrar a probabilidade é preciso multiplicar o produto dos fatores por uma constante ou função normalizadora.

Figura 3.4: Exemplo de grafo de fatores com três variáveis (Sutton & McCallum, 2011).

e) Hidden Markov Models (HMM)

Koller e Fridman (2009) definem Hidden Markov Models (HMM) como um método estatístico que modela uma sequência de observações, assumindo que estas são um produto de uma sequência de estados escondidos, os quais não podem ser observados diretamente. O HMM pode ser representado por um grafo direcionado em que as arestas apresentam uma direção (figura 3.5), $X = \{x_t\}_{t=1}^T$ são as observações e $Y = \{y_t\}_{t=1}^T$ são uma sequência de estados.

Para modelar a distribuição de probabilidade conjunta $p(Y, X)$ de forma tratável, o HMM faz duas considerações sobre independência. Primeiramente, ele considera que cada estado dependa de seu imediato predecessor, isto é, cada estado y_t é independente de todos seus antecessores $y_1, y_2, \dots, y_{t-2}$, dado que seu prévio estado é y_{t-1} . Segundo, o HMM assume que cada observação x_t depende somente do corrente estado y_t .

Assim, podemos encontrar a distribuição conjunta de probabilidade do HMM conforme definido na equação 3.1, em que Z é uma constante de normalização e λ_k é um peso associado a uma função característica f_k . O produto $\lambda_k f_k$ compõe a matriz A , representada na (figura 3.5), e está fortemente relacionado com o fator ψ apresentado anteriormente. O

algoritmo de aprendizado do HMM vai calcular os pesos e a constante de normalização. Este algoritmo é detalhadamente apresentado e explicado em Sutton & McCallum(2011).

$$p(x, y) = \frac{1}{Z} \exp(\sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t)) \quad (3.1)$$

Figura 3.5: Exemplo de um HMM representado por modelo gráfico direcionado

f) Conditional Random Fields (CRF)

O CRF é um modelo gráfico probabilístico não direcionado. Diferente do HMM, ele modela diretamente a probabilidade posterior dos estados escondidos dadas as observações. Assim, ele é capaz de capturar complexas relações entre as observações e os estados escondidos (Koller e Friedman, 2009).

O CRF pode ser visto como uma generalização do HMM, em que as transições de probabilidade, em vez de terem um valor fixo, são uma função das observações. A dependência de cada estado é dada por um conjunto de funções características. O modelo é treinado para aprender estas funções e as distribuições condicionais entre os estados do conjunto de treinamento.

Figura 3.6: Exemplo de um CRF representado por modelo gráfico não direcionado

O CRF faz considerações sobre a independência dos estados, mas não das observações. Por isso ele é um método mais robusto do que o HMM. O CRF pode ser obtido conforme apresentado na equação 3.2. A principal diferença em relação à

equação 3.1 é que neste caso encontra-se uma probabilidade condicional e não uma probabilidade conjunta, e tem-se uma função de normalização $Z(x)$ em função das observações e não mais uma constante.

$$p(y/x) = \frac{1}{Z(x)} \exp(\sum_{k=1}^K \lambda_k f_k(y_t, y_{t-1}, x_t)) \quad (3.2)$$

3.2.2 APRENDIZADO NÃO SUPERVISIONADO

Os métodos de aprendizagem não supervisionada assumem que os rótulos de seus elementos não são conhecidos. A questão é determinar como os dados estão organizados, ou seja, separar os dados em classes ou agrupamentos, baseado na similaridade dos padrões, onde não é fornecido previamente um rótulo associando cada padrão a uma determinada classe.

O objetivo é encontrar padrões nos dados de entrada e organizá-los em categorias. Se dois elementos têm padrões semelhantes, ambos deverão ser associados a uma mesma classe dentro da aprendizagem não supervisionada. Se algum valor de entrada apresenta um padrão que não se assemelha ao de nenhuma classe, o algoritmo cria uma nova classe para abrigar esse valor de entrada.

O aprendizado não supervisionado é sem dúvida muito usual na aprendizagem humana e animal. Ele é também mais empregado do que a aprendizagem supervisionada, uma vez que não necessita de um especialista humano para rotular manualmente os dados (Murphy, 2012).

As técnicas de agrupamento de dados consistem em um vasto campo de pesquisa nas áreas de mineração de dados, segmentação de imagem e reconhecimento de objetos, entre outras. De maneira geral, os métodos de agrupamento de dados podem ser divididos em hierárquicos, baseados em partições, baseados em densidade e baseados em distribuição (Ferreira, 2012). Para a segmentação de dados oriundos de sensor laser o método de agrupamento baseado em partições é o mais apropriado (Klassing et al, 2008).

Dentre os algoritmos baseado em partições o, *k-Means* é um dos mais conhecidos. Inicialmente, o algoritmo posiciona de forma aleatória K centroides que representam os centros de cada grupo. Cada ponto x_i é associado ao centroide C_l mais próximo e o conjunto de pontos associados a cada centroide passa a representar um grupo. Então, cada centroide é movido em direção à média de cada conjunto e as

associações de pontos aos centroides são refeitas. Esse processo de mover os centroides é interrompido quando todos os pontos não mudam suas associações ou apenas quando parte deles muda. O algoritmo 2 apresenta o *k-means* (Ferreira, 2012).

Algoritmo 2: k-means

- 1 Posicionar aleatoriamente K centroides
 - 2 **repita**
 - 3 Associe cada ponto ao centróide mais próximo;
 - 4 Recalcular a posição dos K centroides
 - 5 **até** Até que os grupos não mudem;
-

3.3 OTIMIZAÇÃO: ALGORITMOS COMUNS EM SEGMENTAÇÃO DE NUVEM DE PONTOS 3D

3.3.1 RANSAC

O Random Sample Consensus (RANSAC) é usualmente empregado para detectar o plano que serve de suporte ao veículo (identifica a região plana de uma rua). Proposto por Fischeler e Bolles (1981), o RANSAC é um paradigma para se enquadrar um determinado modelo matemático em um conjunto de dados. Tal paradigma vem sendo utilizado com frequência nas áreas da robótica móvel. Sua simplicidade, desempenho e robustez para lidar com dados experimentais são alguns motivos que levaram à sua escolha (Mendes, 2012).

O paradigma localiza de forma iterativa a melhor combinação entre os dados e um dos modelos disponíveis, estimando melhores valores para os parâmetros livres do modelo. Uma diferença para as técnicas convencionais é que ao invés de utilizar a maior quantidade de dados possível para se obter a solução inicial, o RANSAC começa com um pequeno conjunto de dados e o expande com dados consistentes quando possível (Fischeler e Bolles, 1981).

O algoritmo começa selecionando alguns dados aleatoriamente que satisfaçam os parâmetros livres do modelo e então checa, usando uma margem de erro predefinida, $\mathcal{E}$, a quantidade de dados que se encaixa neste modelo. O processo se repete N vezes ou até encontrar os parâmetros do modelo que se enquadre em τ por cento da amostra.

Os principais passos do RANSAC são resumidos no Algoritmo 3.

Algoritmo 3: RANSAC

1. Selecionar aleatoriamente o número mínimo de dados que satisfaçam o modelo
 2. Testar na amostra os dados que se enquadram no modelo com tolerância $\mathcal{E}$
 3. Se τ por cento da amostra se enquadrou no modelo ou realizaram-se N iterações, estime o modelo com todos os dados enquadrados e termine.
 4. Caso contrário, repita do passo 1 ao 3.
-

3.3.2 ITERATIVE CLOSEST POINT (ICP)

O *Iterative Closest Point* (ICP) é um algoritmo muito empregado para fazer o alinhamento de nuvens de pontos (ou frames), ou seja, minimizar a diferença entre elas. O ICP é apresentado em detalhes no Algoritmo 4. O grande ponto negativo deste algoritmo é o seu elevado custo computacional.

De acordo com Segal et al (2009) o ICP pode ser resumido em dois passos:

- (1) Calcular a correspondência entre dois *scans*;
- (2) Calcular a transformação que minimiza a distância entre os pontos correspondentes.

Algoritmo 4: ICP

Entrada: - Duas nuvens de pontos: $A=\{a_i\}$, $B=\{b_i\}$

- Uma transformação inicial: T_0

- d_{max}

Saída: A transformação correta: T

$T \leftarrow T_0$

Enquanto não convergir

para $i \leftarrow 1$ **to** N

$m_i \leftarrow \text{EncontrePontoMaisProximoEmA}(T.b_i);$

se $\|m_i - T.b_i\| \leq d_{max}$ **então**

$w_i \leftarrow 1;$

senão

$w_i \leftarrow 0;$

fim

fim

$T \leftarrow \text{argmin}_T \{\sum_i w_i \|T.b_i - m_i\|^2\};$

fim

3.3.3 DETECÇÃO DE FEATURES EM IMAGENS

Nesta tese, são utilizados apenas sensores laser. Contudo, o método desenvolvido para identificar a mudança de direção do veículo, apresentado na subsecção 6.2, cria uma imagem digital 2D a partir de uma nuvem de pontos 3D. Além disso, ele extrai algumas *features* dessa imagem para servirem como dados de entrada do algoritmo ICP apresentado na subsecção 3.3.2. Ao usar as *features* da imagem em vez da nuvem de pontos 3D criada pelo sensor laser, reduz-se de maneira bastante acentuada o número de entradas do algoritmo, otimizando o processamento. Assim, faz-se necessária uma breve fundamentação sobre extração de *features*.

O principal propósito da extração de *features* é minimizar o conjunto de dados original, obtendo importantes características que podem ser usadas para reconhecer padrões, segmentar e classificar imagens digitais.

A extração de *features* em imagens digitais pode ser realizada usando diversos tipos de modelos matemáticos, técnicas de processamento de imagens e ferramentas de inteligência computacional.

3.4 CONSIDERAÇÕES FINAIS

Neste trabalho são utilizados redes neurais artificiais, AdaBoost e SVM para os sistemas de classificação, ou para a comparação entre eles. O CRF e o HMM são utilizados para melhorar a saída do sistema de identificação de interseções de vias.

Em uma das fases do sistema de localização desta tese, é necessário transformar uma nuvem de pontos 3D em uma imagem 2D. Dessa imagem são extraídas os pontos de corner (um tipo de *feature* usada em imagens digitais), os quais são encontrados nas junções de regiões da imagem com diferente brilho (Zou et al, 2008).

Os demais conceitos apresentados nesta seção são úteis para a melhor compreensão dos trabalhos relacionados.

4 TRABALHOS RELACIONADOS

A fim de realizar a percepção para aplicação em veículos autônomos é necessário realizar uma segmentação (separação) e classificação dos elementos presentes na cena, ou seja, identificar elementos que aparecem na nuvem de pontos para que o veículo possa caracterizar as áreas livres para navegação, bem como conhecer o posicionamento de todos os obstáculos que possam atrapalhar o seu percurso e as principais estruturas do ambiente, tais como guias e interseções de vias. Os elementos mais comuns de interesse da classificação nesse tipo de aplicação são pedestres, veículos, construções, árvores, postes, entre outros.

4.1 SEGMENTAÇÃO DE NUVEM DE PONTOS 3D

Muitos trabalhos iniciais, utilizando sensores laser para obtenção de mapa ou extração de informações úteis da nuvem de pontos, foram realizados em 2D pelo motivo da maioria dos sensores serem 2D (sensores de feixe único) e, também, para diminuir os gastos com processamento. Contudo, trabalhar apenas em duas dimensões deixa o trabalho muito pobre em informações confiáveis do ambiente.

Com o surgimento de sensores laser, que fazem uma leitura tridimensional do ambiente (sensores com múltiplos feixes), como os sensores Velodyne, Ibeo e Riegl, muitas pesquisas na área de veículos autônomos começaram a utilizar os dados 3D desses novos dispositivos para realizar a percepção de uma maneira muito mais rica em detalhes. Contudo, o volume de dados a ser processado a cada frame aumentou significativamente. Por exemplo, utilizando o sensor laser 2D Sick LMS 291 é possível obter a cada varredura do sensor cerca de 720 pontos 2D. Já com uso do Velodyne HDL-32, a cada varredura são obtidos cerca de 75 mil pontos 3D.

Em virtude da grande quantidade de pontos 3D de cada varredura, também chamada de nuvem de pontos, aumentou-se muito o custo computacional. Como veículos autônomos precisam trabalhar em tempo real, algumas técnicas começaram a ser utilizadas para aumentar a velocidade desse processamento, dentre as quais destacamos a segmentação.

A segmentação subdivide uma imagem em suas partes ou objetos constituintes. O nível até o qual essa subdivisão deve ser realizada depende do problema a ser resolvido (Gonzalez e Woods, 2000). O objetivo da segmentação é simplificar a representação de uma imagem em algo que seja mais compreensível e fácil para se analisar.

A segmentação de nuvem de pontos usualmente segue os mesmos princípios e objetivos da segmentação convencional de imagens. Para aplicações em veículos autônomos, a maioria dos trabalhos divide a segmentação de nuvem de pontos em duas fases. A primeira preocupa-se em extrair a superfície (chão) e a segunda em segmentar os objetos que estão acima da superfície (Douillard et al (2012), Himmelsbach et al (2010)).

4.1.1 DETECÇÃO DOS PONTOS PERTENCENTES AO CHÃO

Em ambientes de tráfego urbano, muitos obstáculos à navegação do veículo autônomo estão apoiados no chão, como veículos, pedestres, placas de sinalização, árvores, postes de energia elétrica, placas de sinalização, construções, entre outros. Assim, o trabalho de segmentação com intuito de futuramente identificar os obstáculos ao redor do veículo é simplificado quando se consegue identificar e extrair os pontos da nuvem que pertencem ao chão.

Alguns trabalhos como (Zhu et al, 2010) e (Samples & James, 2010) consideram que o chão é sempre um plano perfeito e excluem os pontos do chão apenas analisando a altura dos pontos. Essa simplificação não permite que esses métodos possam ser aplicados na maioria dos terrenos em que veículos comuns, onde se pode ter rampas, por exemplo. Com intuito de identificar o chão, Tongtong et al (2011) apresenta um método que funciona em tempo real. Os pontos oriundos de um sensor Velodyne HDL-64E S2 são distribuídos em um *grid* circular de 50 metros de raio, dividido em várias seções circulares, conforme pode ser visto na figura 4.1. Então, ele utiliza os algoritmos *Gaussian process regression* (GP regression) e *Incremental Sample Consensus* (INSAC) para encontrar um modelo geométrico que mais se aproxima ao chão em cada uma das seções circulares. Ao encontrar o modelo, ele considera que todos os pontos da nuvem, que possuem uma distância menor que um determinado limite, pertencem ao chão. Os resultados deste método podem ser vistos na figura 4.2.

Figura 4.1: Divisão do grid polar em M seções de tamanhos iguais. O retângulo ao meio representa a posição do veículo em que está instalado o sensor (extraído de Tongtong et al, 2011)

Figura 4.2: Resultado da segmentação do trabalho de Tongtong et al (2011). Os pontos pertencentes ao chão são coloridos de amarelo. Os pontos vermelhos são os obstáculos e os pontos em verde são os objetos que estão pendurados (acima do solo), como galhos de árvores.

O trabalho de Himmelsbach et al (2010), assim como o método de Tongtong et al, 2011, divide os dados oriundos de uma nuvem de pontos gerada por um Velodyne HDL-64 LIDAR e os distribui em um *grid* circular com n seções circulares ($S_1, S_2, \dots, S_n$) e com uma resolução angular de $\Delta\alpha$. Cada uma dessas seções é dividida em m faixas circulares, $b_1, b_2, \dots, b_m$, como na figura 4.3. Cada ponto 3D, $p_i = (x_i, y_i, z_i)$, é transformado em 2D, $(p'_i = \sqrt{x_i^2 + y_i^2}, z_i)$. Em seguida, toma-se o ponto de menor elevação dentro de cada faixa circular. O objetivo é encontrar segmentos de reta formados por esses pontos de menor elevação em cada seção circular (figura 4.4). Os segmentos de retas cujo coeficiente angular é menor que um determinado limite são considerados como pertencentes ao chão. O último passo é encontrar para cada ponto da seção circular o segmento de reta que está mais próximo a ele. Se a distância entre o ponto analisado e a reta pertencente ao chão mais próxima a ele for inferior a um determinado limite, esse ponto é considerado como pertencente ao chão.

Figura 4.3: Faixas circulares $b_1, b_2, \dots, b_m$ de cada uma das m seções circulares, nas quais são distribuídos os pontos do Velodyne (Himmelsbach et al, 2010).

Figura 4.4: Os pontos de menor elevação de $b_1, b_2, \dots, b_m$ de cada uma das m seções circulares são tomados para verificar o ângulo que formam entre si (Himmelsbach et al, 2010) .

O trabalho de Douillard et al (2011) apresenta um conjunto de métodos de segmentação para vários tipos de nuvens de pontos. Entre os diversos tipos, destaca-se a nuvem de pontos oriunda do sensor Velodyne, que é chamada de dados esparsos. Os pontos da nuvem de pontos não são distribuídos em um *grid* circular, mas sim em um *voxel grid*. O *voxel*, que é o elemento básico de um *voxel grid*, é uma forma resumida de *volumetric pixel* e tem a forma de um cubo. Para diminuir o custo computacional, deixa-se de analisar cada ponto da nuvem e passa-se a considerar apenas os *voxels* ocupados. Para segmentar o chão ele utiliza o *Gaussian Process Incremental Sample Consensus* (GP-INSAC).

O GP-INSAC, que é uma melhoria apresentada por Douillard et al (2011), é um método probabilístico usado para estimar os pontos pertencentes ao chão. Ele parte do pressuposto que a maioria dos pontos da nuvem pertence ao chão (*inliers*) e não a obstáculos ou outros objetos (*outliers*). O GP-INSAC pode ser utilizado para quaisquer nuvens de pontos que tenham em comum a mesma referência cartesiana. Ele oferece uma flexibilidade de processar quaisquer fontes de dados 3D, incluindo fontes múltiplas de dados 3D que tenham sido fundidas em *frame* comum. O método é composto de 4 passos: compressão dos dados, cálculo da semente (escolha de possíveis pontos pertencentes ao chão), emprego do INSAC e descompressão dos dados. O resultado da segmentação pode ser verificado na figura 4.5.

Figura 4.5: Resultado da segmentação de nuvem de pontos apresentado em Douillard et al (2011).

Sobre um veículo são montados cinco sensores laser 2D Sick LMS em ângulos diferentes em relação ao plano horizontal. Essa disposição desses sensores 2D possibilita a geração de uma nuvem de pontos 3D, (X_k^i, Y_k^i, Z_k^i) , em que o índice k representa o frame de aquisição e i representa a posição do ponto dentro da nuvem. Os pontos dessa nuvem são divididos em um *grid* 2D. Cada célula do *grid* armazena o valor da diferença entre o ponto de maior elevação e o de menor elevação, $|Z_k^i - Z_m^j|$.

A detecção de obstáculos em nuvens de pontos de laser pode ser formulada como um problema de classificação, atribuindo a cada célula do *grid* 2D três possíveis valores: ocupado, livre e desconhecido. É considerado ocupado por um obstáculo se $|Z_k^i - Z_m^j|$ é inferior a um determinado valor δ . É considerado livre de obstáculos se existe pelo menos um ponto no *grid* e $|Z_k^i - Z_m^j|$ é maior ou igual a δ . Se nenhuma leitura cai na célula, a célula é considerada desconhecida. Todas as células livres são consideradas uma área navegável para o veículo. Para minimizar o erro, (Thrun, 2006) utiliza um modelo de Markov de primeira ordem. A figura 4.6 apresenta o resultado do método em que as células em branco são as áreas livres, as em cor vermelha representam os obstáculos e as células em cinza representam a área desconhecida.

Figura 4.6: Resultado do método de Thrun (2006). Área em branco representa a área livre, em vermelho os obstáculos e em cinza áreas desconhecidas.

4.1.2 SEGMENTAÇÃO DOS PONTOS NÃO PERTENCENTES AO CHÃO

Os pontos oriundos de um sensor laser não trazem informações mais ricas e precisas sobre os elementos do ambiente, basicamente só contam com o valor da distância em relação ao sensor e intensidade de sinal. Contudo, a intensidade do sinal laser é mais comumente utilizada em trabalhos na área de sensoriamento remoto (Zhou, 2013 e Hui, 2008). O uso da intensidade do sinal laser na área de veículo autônomo ainda é um assunto muito incipiente (Tagolu, 2012). Assim, a maioria dos algoritmos só pode contar com a informação de distância entre os pontos para realizar a segmentação.

Existem, entretanto, alguns trabalhos que, no intuito de usarem algoritmos de segmentação utilizados no processamento de imagem digital, transformam a nuvem de pontos 3D em uma *range image*. Zhu et al (2010) cria uma *range image* a partir dos pontos coletados por sensores laser. Cada pixel é representado pela conversão da distância a um valor de $[0, 255]$. Um exemplo de uma *range image* pode ser observada na figura 4.7. Assume-se que o chão pode ser representado por um plano paralelo à base do veículo e extraem-se todos os pontos do chão apenas levando em consideração a altura dos mesmos. Usando o mesmo critério da altura, os pontos do céu são extraídos também antes de realizar a segmentação. Após excluir os pontos do chão e do céu, utiliza-se o algoritmo apresentado em Falzenswalb e Huttenlocher (2004) para segmentar a imagem.

Figura 4.7: Exemplo de uma *range image* criada a partir de pontos coletados por sensor laser (Zhu et al, 2010).

O trabalho de Klassing (2008) apresenta um método muito eficiente para fazer o agrupamento de dados 3D oriundos de um sensor laser. O algoritmo denominado *radially bounded nearest neighbor graph* (RBNN) pode resumidamente ser descrito pelos seguintes passos:

- (1) Percorrer todos os pontos da nuvem;
- (2) Se o ponto atual já foi assinalado como pertencente a um grupo, vá para o próximo ponto;
- (3) Para o ponto que está sendo analisado
 - Encontre todos os seus vizinhos com distância inferior a r
 - Se qualquer um dos vizinhos pertence a um grupo, faça o ponto pertencer a este grupo, então faça todos os vizinhos sem um grupo pertencerem a este grupo
 - Se o ponto atual já pertence a um grupo e existem vizinhos pertencentes a outros grupos, então agrupe todos esses grupos.

4.2 CLASSIFICAÇÃO DE OBJETOS EM NUVENS DE PONTOS 3D

A identificação dos obstáculos presentes no ambiente tem grande importância para os sistemas de percepção e de planejamento e controle de veículos autônomos. Quando se identifica o obstáculo é possível utilizar ferramentas adequadas para fazer o rastreamento. Por exemplo, veículos possuem um comportamento mais previsível, já pedestres podem se movimentar em qualquer direção e têm um comportamento imprevisível.

A maioria dos trabalhos que usa sensor laser para coletar informações do ambiente realiza a identificação de objetos do ambiente por meio da classificação.

Russel e Norvig (1995) definem classificação como um problema de determinar se um determinado objeto pertence a uma pré-determinada categoria.

Douillard et al (2010) realiza uma segmentação da nuvem de pontos e em seguida classifica os segmentos em quatro classes distintas: carros, postes, árvores e pessoas. Para fazer a classificação ele compara os pontos 3D dos objetos segmentados obtidos com modelos conhecidos. O algoritmo ICP é utilizado para alinhar os segmentos com os modelos e uma métrica é utilizada para calcular o erro. O segmento é classificado como pertencente à classe do modelo que gera o menor erro. O método possui um índice de acerto de 92%.

Zhu et al (2010) gera uma *range image* a partir da nuvem de pontos do sensor Velodyne e realiza a segmentação conforme explicado na seção 4.1.2. Os objetos segmentados são classificados em seis classes distintas: construção, vegetação, carros, árvores, pedestres e bicicletas. Um vetor de características é obtido para cada nuvem de pontos dos objetos segmentados. Algumas dessas características calculadas são o tamanho horizontal da nuvem, a altura máxima e a variância das normais. Para cada ponto da nuvem, os k vizinhos mais próximos destes pontos são identificados e, a partir deles, encontra-se um vetor normal ao plano formado por esses pontos. Para realizar a classificação, utiliza-se *support vector machine* (SVM), tendo como entrada o vetor característico. O método possui um índice de acerto de 89.56%.

4.3 IDENTIFICAÇÃO E CLASSIFICAÇÃO DE INTERSEÇÕES DE VIAS

As interseções de vias são *landmarks* muito comuns nas áreas urbanas e são muito confiáveis, pois são menos sujeitas a variações quando comparadas a construções, árvores e postes. Além disso, a identificação das interseções de vias é muito útil em missões de navegação, pois nas proximidades das interseções há grande presença de veículos e pedestres e, portanto, um elevado risco de acidentes.

No trabalho de Zhu et al (2012), o SVM é utilizado para identificar as interseções de vias. Ele recebe os dados do sensor Velodyne e gera 360 sinais virtuais de laser 2D (figura 4.8). Esses sinais virtuais são gerados a partir de um ponto imaginário à frente do veículo. Os sinais partem do ponto virtual e se estendem até encontrar um obstáculo. Esses sinais virtuais são as entradas de um classificador baseado em SVM e são classificados em cruzamentos e não cruzamentos. Os exemplos

classificados passam por outro classificador SVM para serem classificados em cruzamentos do tipo “T” ou cruzamentos do tipo “cruz”.

Fig. 4.8: Sinais de laser virtuais 2D criados a partir de nuvem de pontos 3D com intuito de identificar uma interseção de vias (Zhu et al, 2012).

Mueller et al (2011) utiliza um laser 3D para criar um mapa de ocupação com células de 0,25 m x 0,25 m. Ele calcula a diferença entre a altura máxima e mínima dos pontos pertencentes a cada uma dessas células. Se o valor encontrado for superior a um determinado limite, considera-se que a célula representa um obstáculo vertical. A transformada de distância apresentada por Borgefors (1986) é empregada para valorizar as células que estão mais distantes de obstáculos. Normalmente, nas áreas urbanas existem obstáculos ao longo dos dois lados das laterais das vias (guias, postes, construções, árvores). Os pontos que estão no meio das vias são os mais distantes dos obstáculos. A figura 4.9 traz um exemplo de uma interseção de vias e de uma reta utilizando a transformada de Borgefors. Pode-se notar que a interseção de vias é caracterizada pela presença de um pico na figura 4.9(a). O método conseguiu identificar corretamente 80% das interseções.

4.4 LOCALIZAÇÃO DE VEÍCULOS AUTÔNOMOS

O trabalho de Baldino et al (2011) descreve um método de localização baseado em informações visuais. É utilizado um mapa topométrico, o qual é construído previamente com ajuda de um veículo base equipado com câmeras de vídeo, IMU e GPS pelo terreno pelo qual se deseja realizar a posterior localização. O mapa é dividido em várias células, espaçadas entre si por uma distância fixa (figura 4.10). Cada uma dessas células armazena *features* extraídas de imagem.

Fig. 4.9: Identificação de cruzamentos usando a transformada de Borgefors (1986). As células com maior altura estão mais distantes dos obstáculos. Em (a) pode-se notar que os pontos mais elevados (cor vermelha) na parte central formam um pico, representando a presença de um cruzamento. Em (b) é apresentado o exemplo de uma via reta, em que há uma linha que passa pelos pontos mais elevados.

Ao passar com o veículo pela área mapeada, localiza-se o mesmo por meio das *features* extraídas da imagem produzida por uma única câmera e da distância percorrida calculada baseada na velocidade do veículo. Um filtro de Bayes é utilizado para determinar a posição mais provável do veículo em relação ao mapa topométrico.

Fig. 4.10: Criação de um mapa topométrico (unidades em metro) referente ao trabalho de Baldino et al (2011). Em (a) é apresentado o mapa completo. Em (b) é mostrada uma visão ampliada da área sinalizada pela seta em (a); cada um dos quadrados representa um vértice do mapa e nesses locais são armazenadas as *features* extraídas da imagem.

O sistema de localização de Mueller et al (2011) faz o reconhecimento de interseções de vias, conforme explicado na seção 4.3, baseado em um mapa obtido a partir de dados georreferenciados (Geo Information System). Cada interseção é caracterizada pela posição do seu centro e pelos seus ramos (vértice e arestas na figura 4.11). A posição do centro do cruzamento, a quantidade de arestas e o ângulo que essas fazem umas com as outras são levados em conta para localizar uma interseção de via no mapa.

Um filtro de partículas é utilizado para melhorar o resultado da localização. No início, partículas são espalhadas pelo mapa. Conforme a detecção das interseções de vias e a medida de deslocamento do veículo (medidos pela odometria e IMU) vão sendo processadas, as partículas vão sendo atualizadas. Esse método foi testado em uma zona rural em um trecho de 4,8 km e foi capaz de identificar oito dos dez cruzamentos existentes nesse trecho.

Fig. 4.11: Duas interseções de vias caracterizadas por um centro e seus ramos (Muller et al, 2011)

O sistema de localização apresentado nesta tese, diferentemente de Baldino et al (2011), não necessita de uma mapa digital prévio do ambiente. Além disso, não sofre problemas com a mudança da luminosidade, já que utiliza um sensor laser e não uma câmera de vídeo. Em relação ao trabalho de Mueller et al, esta tese utiliza apenas um sensor. Não é preciso usar uma IMU para estimar a mudança de direção do veículo. Outra vantagem é que o mapa não precisa ser digital. O mapa topológico pode ser gerado a partir de um mapa viário.

4.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais estudos relacionados com o trabalho proposto nesta tese. Eles descrevem os módulos e subsistemas que integram a solução completa proposta na tese (vide capítulo 5 para maiores detalhes da proposta e metodologia).

A segmentação da nuvem de pontos permite separar o chão dos demais elementos, bem como identificar os elementos presentes no ambiente (estáticos ou dinâmicos), obtidos pelo agrupamento e segmentação de determinadas regiões da nuvem de pontos 3D, e posteriormente classificados de acordo com suas características.

A identificação da interseção de vias é fundamental na localização do veículo baseada em mapas topológicos. Não é necessário o uso de mapas digitais previamente construídos e bem detalhados do ambiente.

5 IDENTIFICAÇÃO DE ESTRUTURAS NAS PROXIMIDADES DO VEÍCULO

Este capítulo apresenta os métodos desenvolvidos e os resultados obtidos para a identificação de obstáculos e a classificação dos tipos de interseções e vias.

5.1 SEGMENTAÇÃO DA NUVEM DE PONTOS

Antes de realizar a identificação das estruturas presentes no ambiente por meio de um sensor laser, é comum realizar algumas etapas básicas. Inicialmente, divide-se a nuvem de pontos em dois conjuntos: dos pontos pertencentes ao chão e dos pontos não pertencentes ao chão. Em seguida, faz-se a segmentação dos pontos não pertencentes ao chão.

Nesta tese, a segmentação dos pontos não pertencentes ao chão foi realizada usando o método de Klassing (2008), apresentado na subseção 4.1.2. Para a identificação dos pontos pertencentes ao solo, um método próprio foi desenvolvido, conforme apresentado a seguir.

Seja $P_i = (x_i, y_i, z_i)$ a coordenada retangular dos pontos oriundos de uma nuvem de pontos 3D. Esses pontos são distribuídos em um *grid* semicircular ($S_1, \dots, S_n$), em que $n = \frac{2\pi}{\Delta\alpha}$ e $\Delta\alpha$ é a resolução angular (figura 5.1). Cada *grid* semicircular é dividido em m bins. O número de bins depende da característica do sensor utilizado. Neste trabalho o valor de m é 32, que é o número de anéis da nuvem de pontos oriunda do sensor Velodyne 32-HDL. Seja P'_{jk} o conjunto de pontos pertencentes a $S_j = (j = 1, \dots, n)$ e $b_k = (k = 1, \dots, m)$. Seja P''_{jk} o conjunto dos pontos de menor elevação de P'_{jk} . Assim, o conjunto P''_{jk} é o conjunto formado pelo ponto de menor elevação de cada anel da nuvem de pontos (b_k) contida em S_j . De forma similar a Himmelsbach et al (2010), os pontos 3D são convertidos em 2D ($xy_i = \left(\sqrt{x_i^2 + y_i^2}, z_i \right)$).

É preciso verificar se os pontos de P''_{jk} pertencem ao chão ou não. Inicialmente, toma-se o primeiro ponto do conjunto P''_{jk} e verifica-se se sua elevação é inferior a um determinado limiar (neste trabalho o valor usado foi de 15 cm). Caso positivo, este ponto é considerado pertencente ao chão. Caso contrário, esse ponto é marcado como não pertencente ao chão e o ponto seguinte é analisado. Ao identificar o primeiro ponto pertencente ao chão, inicia-se uma análise do ângulo de elevação (β), conforme a equação 5.1, em que k representa o índice do primeiro ponto marcado como pertencente

ao chão. Se β é inferior a um determinado valor, o $(k + 1)$ -ésimo ponto é rotulado como pertencente ao chão.

Figura 5.1: Grid com n seções circulares e m “bins”

$$\beta = \left| \frac{z_{b_{k+1}} - z_{b_k}}{xy_{b_{k+1}} - xy_{b_k}} \right| \quad (5.1)$$

Este algoritmo para detecção dos pontos do chão foi testado e validado com dados reais e em tempo real, usando o veículo CaRINA 2. Os resultados desta técnica de detecção dos pontos pertencentes ao chão foram apresentados em Hata et al (2013).

5.2 CLASSIFICAÇÃO DE OBSTÁCULOS

É muito importante reconhecer os tipos de obstáculos existentes ao redor do veículo. Isto pode ajudar o sistema de planejamento e controle a utilizar algoritmos específicos para cada tipo de objeto. Por exemplo, um veículo autônomo deve ter especial atenção com pedestres e pode usar algoritmos menos complexos para rastrear obstáculos estáticos.

Basicamente, os métodos de reconhecimento extraem *features* (características) do conjunto de dados e, então, usam ferramentas de *machine learning* para treinar e validar os dados.

A figura 5.2 apresenta uma visão geral do método de classificação de obstáculos desenvolvido neste trabalho. Os dados são coletados usando um sensor laser 3D. Em seguida os pontos do chão são extraídos, conforme apresentado na subseção 5.1. Todos os pontos não pertencentes ao chão são inicialmente considerados como obstáculos. Estes são segmentados, usando o método de Klasing (2008), o qual leva em

consideração o agrupamento dos pontos mais próximos entre si, sem definir antecipadamente o número de segmentos existente no conjunto de dados analisado. Cada segmento é pré-processado, isto é, são extraídas as *features* do conjunto de pontos, e por último utiliza-se uma ferramenta de *machine learning* para classificar os dados em 4 classes distintas: carros, pessoas, construções e árvores ou postes.

Fig. 5.2: Etapas do método de classificação (Habermann et al, 2016b)

Neste trabalho foi desenvolvido um novo método para extração de *features* com a intenção de reduzir o custo computacional, visando aplicações em tempo real.

Após a extração do chão e a segmentação dos demais pontos da nuvem de pontos (método de Klasing (2008)), encontra-se o centro de massa, a altura, a largura e a profundidade de cada um dos segmentos. Com intuito de diminuir o número de *features* para a futura classificação, a largura ou a profundidade de cada segmento serão desprezadas. A ideia é ter uma projeção 2D dos pontos 3D pertencentes a cada grupo encontrado após a segmentação. Considera-se sempre a altura de cada ponto. O outro valor numérico a ser considerado de cada segmento de pontos é o maior valor entre largura ou profundidade.

A figura 5.3 mostra os pontos 3D oriundos de um dos grupos encontrados pelo processo de segmentação. Encontram-se a largura W e a profundidade D . Como D é maior que W , a projeção dos pontos 3D é realizada no plano formado pelos eixos Z e X . Os pontos pertencentes ao eixo Y são desprezados neste caso. A altura h sempre é mantida.

Fig. 5.3. Os pontos 3D são relacionados a um veículo e foram obtidos após o processo de segmentação baseado no método de Klassing (2008).

Os pontos 2D encontrados de cada segmento são distribuídos em um *grid* de 64 células (0,5 m x 0,5 m) e, em seguida, as células são normalizadas. Os valores normalizados de cada célula alimentam as entradas de um classificador. A figura 5.4 apresenta os pontos obtidos do segmento da figura 5.3 transformados em 2D. A figura 5.5 apresenta um *grid* 2D normalizado com um exemplo de cada uma das classes: carros, pedestres, árvores ou postes e construções.

Fig.5.4 Os pontos 2D são uma projeção nos eixos X e Z dos pontos da figura 5.3. Como a profundidade do veículo é maior que dimensão da largura do veículo ($D > W$), os pontos do eixo Y foram desprezados.

Fig.5.5 Representação de um *grid* 2D normalizado de (a) um carro, (b) um pedestre, (c) uma árvore e (d) uma construção.

Como classificador, adotou-se uma rede neural artificial MLP contendo 66 entradas (64 células do *grid*, altura e $\max(\text{largura}, \text{comprimento})$), 4 saídas (carros, pessoas, construção e postes ou tronco de árvores) para classificar os objetos.

Para definir o número de neurônios da camada escondida, foram realizados experimentos de validação cruzada em 16 diferentes tipos de topologias. Em cada topologia foi acrescentado de 2 o número de elementos escondidos, iniciando em 2 e finalizando em 32. O experimento foi repetido 30 vezes, considerando um conjunto de dados de 2365 exemplos. As topologias que geraram os menores valores de *mean squared errors* (MSE) foram selecionadas.

A figura 5.6 apresenta o resultado do *cross-validation* (95% de intervalo de confiança). De acordo com os resultados, verifica-se que as configurações 66-14-4, 66-16-4, 66-18-4 e 66-20-4 tiveram o menor MSE. Por essa razão, elas foram escolhidas para o treinamento. Foi utilizado o algoritmo *resilient back-propagation* limitado a 5000 ciclos. A base de dados de 2365 exemplos foi dividida em três partes: 70% para treino, 20% para validação e 10% para testes. A figura 5.7 apresenta o MSE para cada um desses três conjuntos.

Fig.5.6 Resultados do experimento de validação cruzada para 16 tipos diferentes de topologia

Fig.5.7 Treinamento da ANN para 4 topologias diferentes. As curvas em azul representam o MSE do treinamento, e as curvas em vermelho o MSE da validação.

Utilizou-se o conjunto de dados de teste, ou seja, de exemplos não vistos para avaliar o desempenho das quatro topologias. De acordo com as matrizes de confusão de cada uma das topologias apresentadas na figura 5.8, pode-se verificar que a capacidade dos modelos classificarem corretamente carros e pessoas é levemente superior a de classificar construções, postes e árvores.

Fig.5.8 Matriz de Confusão de 4 topologias diferentes de ANN. As letras A, B, C e D representam as classes dos carros, pessoas, postes ou árvores e construções, respectivamente.

Pela análise da tabela 5.1, onde são destacados em negrito os melhores resultados para cada um dos índices analisados, pode-se verificar que a topologia 66-16-4 apresenta um desempenho melhor do que as outras três topologias analisadas.

A rede neural artificial 66-16-4 foi comparada com outros classificadores comuns na área de *machine learning*. Para fazer isso foi utilizado um framework denominado Weka (Holmes & Donkin, 1994) , que é uma coleção de algoritmos de machine learning.

A tabela 5.2 apresenta os valores de *correctness*, *F1-score*, e ROC AUC de quatro métodos: SVM com kernel sigmoidal, SVM com kernel RBF, 2-NN e rede bayesiana. Esses valores foram obtidos com o Weka, considerando o mesmo conjunto de dados utilizado para avaliar o desempenho da ANN. Pelas análises dos índices de desempenho apresentados nas tabelas 5.1 e 5.2, pode-se verificar que a ANN com 16 neurônios na camada escondida teve um resultado superior que o SVM sigmoidal, SVM RBF e rede bayesiana. O 2-NN teve um valor de ROC AUC superior, contudo a ANN foi superior nos índices de acerto e F1 score.

Tabela 5.1 Desempenho de diferentes configurações de ANN

	66-14-4	66-16-4	66-18-4	66-20-4
Training MSE	0.02175	0.02659	0.02417	0.02900
Validation MSE	0.27984	0.21897	0.31030	0.24125
Right	86.47%	88.16%	84.57%	87.10%
Wrong	13.53%	11.84%	15.43%	12.90%
Accuracy	0.75786	0.80275	0.75473	0.78831
F1 Score	0.82341	0.87328	0.83386	0.86222
ROC AUC	0.88317	0.90860	0.87320	0.89287

Tabela 5.2 Desempenho de classificadores obtidos com o Weka

	SVM(Sigmoid)	SVM (RBF)	2-NN	Bayesian Net
Rigth	64.33%	77.28%	86.83%	79.83%
Wrong	35.67%	22.72%	13.17%	20.17%
F1 Score	0.625	0.773	0.867	0.802
ROC AUC	0.750	0.840	0.951	0.945

A figura 5.9 apresenta exemplos da classificação obtida pela ANN 66-16-4 feita com dados reais e em tempo real.

Em relação aos métodos que usam apenas um sensor laser para realizar a classificação de obstáculos, o método apresentado nesta tese obteve resultados similares em relação a Zhu et al (2010), Zhan et al (2011) e Wang et al (2012). Contudo, é difícil afirmar categoricamente que um método é melhor ou pior, pois todos usaram um conjunto de dados diferente para realizar as análises. A vantagem deste método é o baixo custo computacional. A redução dos segmentos 3D em um plano 2D possibilitou o emprego em tempo real.

Este método de classificação apresentado acima foi testado e validado com dados reais, usando os veículo CaRINA 2. Os resultados deste trabalho foram Habermann et al (2013c) e Habermann et al (2016b).

Fig.5.9 Método de classificação de objetos usando uma ANN 66-16-4. Os pontos em azul representam construção; em verde, postes ou árvores; em vermelho, carros e em amarelo representam pessoas.

5.3 DETECÇÃO E CLASSIFICAÇÃO DE INTERSEÇÕES DE VIAS

A detecção de interseções de vias e identificação da forma da via são tarefas muito importantes na área de veículos autônomos. Além de serem fundamentais para elevar o nível de segurança, essas tarefas podem ser úteis em um sistema de localização baseado em *landmarks*.

Interseções são áreas de elevada concentração de veículos e pedestres e por isso há um elevado índice de acidentes em suas proximidades. Quase metade de todos os acidentes automobilísticos ocorridos no estado de *New South Wales* (Austrália) em 2012 ocorreram próximos a um cruzamento.

A identificação da forma da via, por exemplo, saber se o veículo está trafegando em uma curva, reta ou está em algum tipo específico de cruzamento, pode ajudar em determinar uma velocidade mais adequada e também pode servir como referência em um sistema de localização.

Por questões didáticas, será exposto inicialmente o método de identificação da forma da via e em seguida, então, é exposto o método robusto para detecção de interseções de vias.

5.3.1 Classificação das Formas de Vias

A figura 5.10 apresenta a visão global do método de classificação da forma da via desenvolvido nesta tese. Utiliza-se um sensor laser 3D para coletar os dados do ambiente. Em seguida, o módulo de detecção da via (*Road Detector*) recebe os pontos oriundos dos métodos *Curb Detector* (Hata et al, 2013) e *Surface Detector* (extração de pontos do chão - subseção 5.1.1) e os distribui em um grid de 30x30 células, com resolução de 2 m². O *Curb Detector* é um método que consegue identificar as guias nas bordas das ruas e rodovias.

Os valores de cada célula do *grid* são normalizados a fim de alimentar uma ANN MLP com 900 entradas em oito classes distintas: reta, curva à esquerda, curva à direita, via lateral à esquerda, via lateral à direita, bifurcação em Y, interseção em T e cruzamento de vias.

Para determinar a melhor topologia da ANN, foi realizado o *10-fold cross-validation* em dezesseis topologias diferentes, que foram escolhidas baseadas em experiências prévias. O experimento foi repetido 30 vezes, usando 4893 exemplos. Esses exemplos foram coletados na cidade de São Carlos usando o Velodyne-32 HDL. As topologias que obtiveram os menores valores de MSE foram 900-5-8, 900-10-8, 900-15-8 e 900-20-8.

Para o treinamento da rede, as quatro topologias com melhor desempenho foram treinadas separadamente. O conjunto de dados com 4893 exemplos foi dividido aleatoriamente em três conjuntos: treinamento (70%), validação (30%) e teste (10%). Para o treinamento da rede foi utilizado o algoritmo *resilient back-propagation* e o número de iterações foi limitado em 5000. As topologias foram treinadas em 3 situações diferentes. A primeira situação considerou apenas os pontos oriundos do *Curb Detector*. A segunda usou apenas os dados do *Surface Detector* e a terceira utilizou os dados conjuntos do *Curb Detector* e também os do *Surface Detector*. As topologias com

melhor desempenho para cada uma dessas situações foram selecionadas. A Tabela 5.3 apresenta os resultados.

Tabela 5.3 – Desempenho das melhores topologias para cada uma das situações: dados do conjunto de teste do *Curb Detector*, *Surface Detector*, *Curb Detector* + *Surface Detector*.

	900-20-8 (curb)	900-10-8 (surf)	900-10-8 (curb+surf)
Right	82.43%	80.77%	94.47%
Wrong	13.08%	14.33%	4.49%
Unknown	4.49%	4.90%	1.04%
Accuracy	0.7578	0.75615	0.91799
F1 score	0.8234	0.84019	0.95394

Pela análise da Tabela 5.3, a rede ANN 900-10-8, que leva em conta os dados do *Curb Detector* e *Surface Detector*, obteve o melhor desempenho. Isso mostra que a combinação dos métodos *Curb Detector* e *Surface Detector* traz um grande ganho ao método de classificação. A figura 5.11 apresenta alguns resultados obtidos com esse método.

Fig.5.10 Visão geral do método de classificação de tipos de cruzamentos

Este método de classificação da forma das vias foi testado e validado com dados reais em tempo real, usando os veículo CaRINA 2. Os resultados foram apresentados em (Hata, Habermann, Osorio & Wolf, 2014)

Fig.5.11 Classificação de tipos de vias quanto à forma. Nestas imagens os pontos em vermelho foram detectados pelo Surf Detector e os pontos em amarelo pelo Curb Detector. Esses pontos em conjuntos são tratados e alimentam um ANN a fim de classificar em reta, curva à esquerda, curva à direita, cruzamento de vias, via lateral à esquerda, via lateral à direita, bifurcação em Y e interseção em T.

5.3.2 Identificação de Interseção de Vias

Nesta tese, um método robusto para detecção de interseções de vias é de fundamental importância para o sistema de localização baseado em um mapa topológico que usa apenas um sensor laser para coletar dados do ambiente.

Para este sistema de localização, apresentado na seção 6, é muito importante reconhecer o início e fim de cada interseção (figura 5.12). Pois neste intervalo, os dados obtidos pelo sensor são analisados para verificar se houve ou não mudança na direção do veículo e assim determinar qual destino tomado.

Fig.5.12 Saída desejada de um detector de interseções. A mudança de direção do veículo não interfere na detecção da interseção.

Os métodos de extração dos pontos do chão, de segmentação, de classificação de obstáculos e de classificação das formas da via foram desenvolvidos baseados apenas na análise de um único *frame*. Já para a detecção de interseção de vias, foi considerada uma sequência de *frames*. Para isso duas ferramentas são muito úteis para análise: *Hidden Markov Models* (HMM) e *Conditional Random Fields* (CRF). Um resumo dessas duas teorias é apresentado na subseção 3.2.1.

O detector de interseções de vias pode ser dividido em três etapas: extração de *features* de uma nuvem de pontos 3D, uso de um classificador convencional para classificar cada frame em interseção e não-interseção e, por fim, uso de um método estatístico para melhorar a resposta, analisando uma sequência de saídas do classificador.

a) Extração de features

Como existe a necessidade de se detectar a interseção mesmo nos casos em que o carro mude de direção, é importante usar um conjunto de *features* que seja invariante à rotação do veículo.

Neste trabalho foi utilizado o conjunto com 41 *features*, cujas equações são apresentadas no Apêndice A do trabalho de Granström et al (2011). Basicamente, a extração dessas *features* são baseadas no histograma de distribuição dos pontos, volume coberto pela nuvem de pontos, curvatura, centroide, diferença angular média e *Range Kurtosis*.

O apêndice I desta tese apresenta o programa feito em Matlab para extrair as *features* de uma nuvem de pontos 3D, baseado no trabalho de Granström et al (2011).

b) Escolha do melhor classificador convencional

Essas *features*, oriundas de um único frame, alimentam um classificador, o qual rotula as saídas em interseções e não interseções. Três classificadores muito comuns em trabalhos dessa natureza foram pré-selecionados: SVM, AdaBoost e ANN. Para selecionar o classificador de melhor desempenho, foram considerados quatro conjuntos de dados, conforme apresentado na Tabela 4. É importante frisar que estes quatro conjuntos foram extraídos de quatro regiões distintas.

Os dois primeiros conjuntos de dados, denominados SC1 e SC2, foram coletados com uso do sensor Velodyne HDL-32 na cidade de São Carlos. Para obtenção de SC1 foram percorridos cerca de cinco quilômetros, entre ruas e avenidas, passando por 57

interseções de vários tipos. SC2 tem cerca de um quilômetro e 12 interseções de vias. Os dados de São Carlos foram adquiridos em áreas bem diversificadas, contendo exemplos com elevado ou reduzido número de obstáculos verticais (figura 5.13).

Os conjuntos denominados K1 e K2 fazem parte de dados públicos extraídos do repositório KITTI² (Universidade de Karlsruhe, Alemanha). Estes conjuntos de dados foram obtidos com o uso do sensor laser Velodyne HDL-64. K1 conta com 61 interseções de vias e K2 possui 19. Esses dados fazem parte do conjunto *odometry benchmark*. K1 inclui as pastas de 00 a 07 desse conjunto e K2 contém a pasta 08. Exemplos de nuvem de pontos dos conjuntos SC1 e K1 podem ser vistos na figura 5.14.

Tabela 5.4: Conjunto de dados utilizados no método de detecção de interseções de vias

	Conjunto de Dados			
	SC1	SC2	K1	K2
Interseções	57	12	61	19
Frames	4150	892	5046	1834
Frames de interseções	2794	645	3323	1280
Frames de não interseções	1356	247	1723	554
Sensor	Velodyne 32		Velodyne 64	

² <http://www.cvlibs.net/datasets/kitti/>

(a)

(b)

Fig.5.13 Imagens de locais onde foram coletados os dados. Em (a) tem-se uma interseção com pouca informação vertical, em (b) há vários obstáculos estáticos.

Fig.5.14 Exemplos de interseções de vias obtidas com sensor laser 3D. As interseções em (a) e (b) são do conjunto de dados K1, obtidas com o Velodyne HDL-64 em Karlsruhe- Alemanha. As interseções (c) e

(d) são do conjunto de dados SC1 e foram obtidas com o sensor Velodyne-32 HDL na cidade de São Carlos.

Para determinar o melhor modelo de cada classificador, foram considerados os conjuntos SC1 e K1. A fim de encontrar a melhor topologia da ANN, utilizou-se apenas uma camada escondida com o número de neurônios variando de 5 a 40, usando o *10-fold crossvalidation*. A melhor topologia encontrada para a ANN foi de 41-5-2 (41 entradas, 5 neurônios na camada escondida e 2 saídas).

Para a análise do SVM, foram analisados dois *kernels* diferentes, polinomial e linear. Ambos os *kernels* aprenderam muito bem o conjunto de treinamento. Contudo, o *kernel* linear foi capaz de generalizar melhor.

Diferentes configurações do AdaBoost foram treinadas, modificando o número de classificadores fracos. Para os dados de São Carlos o uso de 50 classificadores fracos trouxe melhor desempenho, já para os dados de Karlsruhe, 100 classificadores fracos trouxeram um resultado mais significativo.

A tabela 5 apresenta um resumo do desempenho de cada um dos classificadores, considerando os conjuntos de teste SC2 e K2. Pode-se verificar que para ambos os conjuntos de teste, o classificador AdaBoost teve o melhor desempenho.

O classificador AdaBoost obteve uma acurácia (*accuracy*) de 0,87 e 0,9027 para os conjuntos de teste SC2 e K2, respectivamente. O resultado numérico pode ser considerado bom. Contudo, o uso desse classificador não gera uma saída adequada para o problema de detectar o início e fim de uma interseção de vias, conforme explicado no parágrafo a seguir.

Tabela 5.5 Desempenho dos classificadores para os conjuntos de teste SC2 e K2

Dados	Classificador	Acuracy	Precision	Recall
SC2	ANN	0.7088	0.8700	0.8101
	AdaBoost	0.8700	0.6478	0.8466
	SVM	0.7220	0.7854	0.4987
K2	ANN	0.8049	0.6606	0.6867
	AdaBoost	0.9027	0.8736	0.8190
	SVM	0.3626	0.8267	0.3009

A figura 5.15 apresenta o resultado da classificação feita pelo AdaBoost, considerando duas interseções seguidas do conjunto de testes SC2. Os pontos de maior

amplitude em cada uma das curvas representam as interseção de vias. Pode-se verificar que o AdaBoost gera seis bordas de subida e seis de descida (cada uma destas bordas representando a identificação "intermitente" de uma interseção). Se fosse considerada cada borda de subida e descida como o início e fim de cada cruzamento, este trecho com apenas duas interseções, seria erroneamente contabilizado com seis interseções de vias. Em decorrência desse problema, é importante agregar um método estatístico (considerando o aspecto da sequência temporal de dados) à saída desse classificador para melhorar a resposta final.

Fig.5.15 Resultado do classificador AdaBoost referente a duas interseções de vias. Em azul, a saída do AdaBoost de cada laser frame. Em vermelho, o *ground truth*, sendo que os valores de maior amplitude representam a interseção.

c) Escolha do melhor método estatístico

O classificador AdaBoost foi alimentado com os dados de treinamento para gerar duas sequências (uma com os de SC1 e outra com os de K1), cada uma dessas composta de interseções e não interseções.

Para encontrar os melhores parâmetros do HMM foi usada a função *hmmestimate* do Matlab. Para verificar o desempenho, utilizou-se o modelo encontrado pela função *hmmestimate* e a função *hmmdecode* para os dados de teste SC2 e K2.

O CRF foi treinado e testado de forma similar ao HMM. Contudo, foi empregada a biblioteca de algoritmos de Schmidt & Swersky (2008), considerando-se uma sequência de 20 frames.

A tabela 5.6 apresenta os resultados obtidos com o HMM e CRF. Os valores de *accuracy* e *precision* obtidos com o HMM e CRF melhoraram o desempenho para os dois conjuntos de testes, SC2 e K2, comparando apenas com o uso do Adaboost (classificador com melhor desempenho em relação à ANN e SVM). Para os dados de

SC2, o CRF teve um resultado levemente superior ao HMM. Contudo, para os dados de K2 foi o HMM que superou o CRF.

Tabela 5.6: Resultado do método de detecção de interseções

Dados	Classificador	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>
SC2	AdaBoost	0.8700	0.6478	0.8466
	AdaBoost+HMM	0.8789	0.6680	0.8639
	AdaBoost+CRF	0.8789	0.8145	0.7287
K2	AdaBoost	0.9027	0.8736	0.8190
	AdaBoost+HMM	0.9104	0.8889	0.8285
	AdaBoost+CRF	0.9044	0.8393	0.8484

Na figura 5.16 as curvas em vermelho representam o *ground truth*, observa-se que tanto o HMM e o CRF reduziram o ruído do sinal de saída do classificador AdaBoost.

Nesta tese, é importante encontrar um sistema que consiga caracterizar o início e fim de um cruzamento, pois isso é fundamental para se obter a mudança de direção do veículo ao passar por uma interseção de via (apresentado na seção 6). O início e fim de um cruzamento pode ser obtido por um pico de subida e outro de descida (figura 5.15). Para que tenha um bom desempenho, pulsos espúrios (ruído) devem ser evitados.

Nesse sentido, o CRF apresentou um melhor desempenho em relação ao HMM. Na figura 5.16(a), o CRF deixou de identificar apenas uma das interseções e não gerou nenhum falso positivo. Já o HMM, gerou um falso positivo. Pode-se visualizar na figura 5.16(b) que o CRF deixou de reconhecer duas interseções e gerou dois falsos positivos. O HMM deixou de reconhecer apenas uma, mas gerou quatro falsos positivos.

Figura 5.16 Resultado do método de identificação de interseções de vias.

5.4 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Na subseção 5.2 foi apresentado um método para classificar os obstáculos em quatro classes: veículos, pedestres, postes ou tronco de árvores e construção. Em relação aos trabalhos que tentam fazer a classificação baseados apenas em único frame do laser, e não utilizando uma sequência de frames, o método apresentado nesta tese obteve um ótimo desempenho. Já para a classificação de interseções de vias, considera-se uma sequência de frames, algo que não foi abordado por outros pesquisadores.

O método de detecção de interseção de cruzamentos apresentado na subseção 5.3 teve um desempenho bastante satisfatório. Porém, é importante alcançar uma eficácia muito superior para poder ser empregado em situações reais. Analisando as interseções em que o sistema teve dificuldade em detectar, observou-se que se tratava de um ambiente com pouca informação vertical. Basicamente, contava-se apenas com a presença das guias para caracterizar uma interseção, conforme a figura 5.13(a). A fim de melhorar a identificação de vias, pode-se utilizar a saída do *Curb Detector* (subseção 5.2), que é um algoritmo que procura identificar as guias das ruas, para melhorar este

método de detecção de interseções. Uma outra possibilidade é utilizar um odômetro para estimar a distância percorrida pelo veículo e comparar com um mapa topométrico, contendo as interseções e as distâncias entre elas.

6 LOCALIZAÇÃO BASEADA NA DETECÇÃO DE INTERSECÇÃO DE VIAS

Diferente da grande maioria das pesquisas nessa área, esta tese não pretende apresentar um sistema que seja capaz de localizar um veículo relativo a um mapa métrico digital. O objetivo é apresentar um método alternativo sem uso de GPS, obtendo uma localização global aproximada, mas que seja eficaz para que o veículo possa navegar em uma área urbana.

O sistema de localização é baseado no reconhecimento de trechos de vias entre as interseções. De uma maneira geral, nas cidades brasileiras esses trechos, que são limitados por duas interseções de vias vizinhas, possuem na média um comprimento de 100 metros. Assim, o sistema de localização proposto terá um erro inerente médio estimado de 100 metros. É possível saber com precisão o trecho (localização do veículo entre duas específicas interseções) onde o veículo está, mas não a posição do mesmo em relação ao trecho.

Apesar do elevado erro deste método comparado a sistemas que usam GPS, é possível realizar de modo adequado a navegação do veículo. Para que isso seja possível, o sistema de navegação deve ser capaz de planejar e seguir uma determinada rota e ser capaz de se locomover de forma segura. Para isso basta ter o conhecimento da posição de todos os obstáculos e da área navegável (conforme apresentado na subseção 5.1) produzido em tempo real pelo próprio veículo (percepção local), e um mapa topológico similar ao descrito na subseção 6.1.

6.1 CONFECÇÃO DO MAPA TOPOLÓGICO

Esta subseção apresenta um mapa topológico que pode ser construído a partir de um mapa viário urbano com limitada precisão.

O mapa viário deve conter todos os cruzamentos da área em que o carro pode percorrer. Todos os cruzamentos devem ser identificados. Deve-se, ainda, identificar os pontos terminais, ou seja, pontos que identificam o término do mapa.

Para exemplificar a confecção do mapa, utiliza-se a figura 6.1, a qual apresenta um mapa viário nas proximidades do ICMC-USP. Os círculos identificados de C1 a C9 representam todos os cruzamentos pertencentes à área em que o veículo poderá navegar. Os quadrados nomeados de F1 a F8 representam os pontos terminais do mapa. A partir desses pontos o veículo não deveria navegar, pois não contará mais com um mapa de referência.

6.2 MÉTODO PARA IDENTIFICAR MUDANÇA DE DIREÇÃO DO VEÍCULO

Neste trabalho, as interseções de vias foram escolhidas como importantes pontos de referência (*landmarks*) para propiciar a localização do veículo. Para se conhecer o trajeto por onde está se navegando, é muito importante identificar qual é a direção tomada pelo veículo ao passar pelas interseções de vias do caminho.

A figura 6.3 apresenta duas nuvens de pontos 2D, oriundas de nuvens 3D em que a altura dos pontos foi desprezada. O veículo inicialmente seguia na direção AB da figura 6.3(a) e faz uma conversão à direita, buscando o ponto C. Visualmente, é possível perceber nitidamente o deslocamento entre as duas nuvens de pontos das figuras 6.3(a) e 6.3(b). O método desenvolvido para identificar a mudança de trajetória do veículo ao passar por uma interseção é baseado no cálculo do deslocamento angular entre as nuvens de pontos consecutivas, considerando os dados de início ao término de cada interseção.

Figura 6.3- Deslocamento angular entre duas nuvens de pontos 2D, tendo como referência o eixo A-B. Em (a), o veículo que trafegava no sentido A-B começa uma conversão à direita, com intuito de alcançar o ponto C. Em (b) é apresentada uma nuvem de pontos com a conversão à direita quase finalizada.

Para calcular o deslocamento angular entre as nuvens de pontos, são analisadas as nuvens que são coletadas entre o início e término de cruzamento, realizando oito passos apresentados a seguir (os quais são detalhados nos parágrafos seguintes):

- i. Extrair os pontos pertencentes ao chão da nuvem de pontos (subseção 5.1).
- ii. Transformar a nuvem de pontos 3D em uma nuvem 2D (desprezar a altura dos pontos).
- iii. Transformar a nuvem de pontos 2D em uma imagem preto e branco (P&B).
- iv. Extrair os pontos de *corner* da imagem.

- v. Usar o algoritmo ICP (subseção 3.3.2) com os pontos de *corner* de duas imagens subsequentes para a obtenção da matriz de translação e rotação.
- vi. Multiplicar T_y (componente transversal da matriz de translação gerada pelo ICP) por R_y (ângulo de rotação em relação a um eixo perpendicular ao plano em que o veículo trafega, obtido da matriz de rotação do ICP) e desprezar os valores do produto entre T_y e R_y superiores a 200 (verificou-se empiricamente que valores superiores a 200 representavam ruídos).
- vii. Calcular o valor de $\sum_{i=a}^b T_y^i R_y^i$, em que $[a, b]$ representa o conjunto das nuvens de pontos dentro do intervalo de início e término de cruzamento.
- viii. A direção do veículo ao término do cruzamento é determinada, levando-se em conta o seguinte:
 - Se $\sum_{i=a}^b T_y^i R_y^i$ for maior do que 150, significa que o veículo virou à esquerda (o valor 150 foi obtido empiricamente)
 - Se $\sum_{i=a}^b T_y^i R_y^i$ for menor do que -150, significa que o veículo virou à direita
 - Se $\sum_{i=a}^b T_y^i R_y^i$ for menor do que 150 e maior do que -150, significa que o veículo continuou em frente após passar pela interseção.

Para transformar uma nuvem de pontos 2D em uma imagem P&B, os pontos da nuvem são divididos em células de dimensão 0,2 x 0,2 m. Cada célula será relacionada a um pixel da imagem. As células ocupadas recebem o valor 0 (preto) e as células não ocupadas o valor 1 (branco). A figura 6.4 representa a vista superior de uma nuvem de pontos 3D. A imagem obtida desta nuvem, após a exclusão dos pontos pertencentes ao chão, pode ser observada na figura 6.5.

Figura 6.4 Vista superior de uma nuvem de pontos 3D (unidade em metros).

Figura 6.5 Imagem em preto e branco obtida da nuvem de pontos da figura 6.4

Após transformar uma nuvem em uma imagem digital P&B, são extraídos os pontos de *corner*, que são um tipo de *feature* extraído de uma imagem. Os pontos de *corner* são encontrados nas junções de regiões da imagem com diferente brilho (Zou et al, 2008). Para extrair os pontos de *corner*, foi utilizada uma função do Matlab denominada ***corner***, fixando o número máximo de pontos a 200. Este valor é o *default* da função *corner* e funcionou muito bem para os experimentos desta tese.

O principal objetivo de encontrar os *corners* da imagem é realizar uma redução de dimensionalidade a fim de empregar o algoritmo ICP. A nuvem de pontos 3D original tem em média 70 mil pontos. Após a transformação da nuvem em imagem e da extração de pontos de *corner* desta imagem, é encontrado um conjunto de no máximo 200 pontos, que podem ser considerados um extrato da geometria original da nuvem de pontos. A figura 6.6 apresenta os pontos de *corner* encontrados na figura 6.5.

Figura 6.6 Pontos de corner referentes à figura 6.5.

O algoritmo ICP, apresentado na subseção 3.3.2, é muito conhecido na área de imagens e mapeamento digitais. Ele é utilizado para fazer o alinhamento entre duas nuvens de pontos. O algoritmo retorna a matriz de rotação e de translação dos pontos. O custo computacional deste algoritmo é extremamente elevado, por isso ele não é muito comum em aplicações na área de veículos autônomos que possuam um requisito de processamento de dados em tempo real, ou, restrições impostas pela capacidade de processamento de dados do sistema embarcado.

Nesta tese, para reduzir o custo computacional, em vez de se utilizar os pontos de um *frame* do sensor laser (cerca de 70 mil pontos 3D), os pontos de *corner* é que são utilizados como a entrada do algoritmo ICP. A saída deste algoritmo gera duas matrizes, $T = [T_x \ T_y \ T_z]$ e $R = [R_x \ R_y \ R_z]$, que são as matrizes de translação e rotação. Os valores T_y e R_y são os mais relevantes para este método, pois eles estão relacionados com a mudança de direção do veículo. Verificou-se durante os testes que o produto entre T_y e R_y gerava uma resposta bem distinta quando o veículo mudava de direção e os demais valores de T e R pouco contribuíam nesta tarefa.

No passo *vii*, após realizar testes em três diferentes percursos, verificou-se que o valor numérico 150, resultante do somatório $\sum_{i=a}^b T_y^i R_y^i$, seria o mais adequado para fazer a correta distinção entre seguir em frente, virar à esquerda ou virar à direita.

Este método reduziu significativamente o custo computacional. O ICP considerando duas nuvens de 70.000 pontos leva horas para ser processado, já utilizando no máximo 200 pontos, leva cerca de 0,1 segundo.

6.3 LOCALIZAÇÃO

A localização de robôs móveis nada mais é do que determinar a posição de um robô, neste caso um veículo, em relação a um dado mapa.

O sistema de localização proposto nesta tese é do tipo *position tracking* (subseção 4.5), ou seja, o robô conhece sua posição inicial, e utiliza um mapa topológico de simples construção, o qual tem as interseções de vias como *landmarks* (subseção 6.1).

Considerando que o veículo sabe sua posição original (última interseção de via pela qual passou) e seu próximo destino (interseção para qual está indo), toda vez que

passar por uma interseção, o sistema determinará se houve mudança de direção, conforme o método apresentado na subseção 6.2.

Como exemplo, tomando por base o mapa da figura 6.1, supondo que a interseção C9 foi a última interseção de via pela qual o veículo passou e que o veículo virou à direita após passar por C1, o sistema atualizará sua posição em relação ao mapa, mudando de C9-C1 para C1-C8 (vide figuras 6.1 e 6.2). O sistema de localização sempre manterá atualizada a atual posição do veículo, apresentada por duas interseções, a interseção pela qual passou e a interseção para onde se locomove.

Foram realizados três testes para verificar o desempenho do sistema de localização. Os dois primeiros testes foram feitos nas proximidades do ICMC-USP. O terceiro teste foi feito em uma região de São Carlos onde havia uma rotatória, que é um tipo de interseção mais complexa.

O veículo Carina 2 e o sensor laser Velodyne 32-HDL foram utilizados para coletar os dados dos testes. Os algoritmos de identificação de interseção e de localização foram processados em laboratório, utilizando os programas ROS e Matlab.

Para os testes 1 e 2, a região mostrada na figura 6.1 foi mapeada conforme explicado na subseção 6.1. Nessa mesma região foram feitos dois deslocamentos distintos (testes 1 e 2).

No primeiro teste o veículo percorreu uma distância de cerca de 850 metros, passando por sete interseções de vias. O trajeto realizado pelo veículo pode ser visto na figura 6.7.

Figura 6.7 – Trajeto realizado pelo veículo durante o primeiro teste do sistema de localização.

O sistema acertou todas as mudanças de direção feitas pelo veículo e conseguiu localizar o veículo de forma precisa em relação ao mapa topológico. A figura 6.8 apresenta a curva gerada pela multiplicação entre R_y e T_y . Se o somatório $\sum_{i=a}^b T_y^i R_y^i$, em que $[a, b]$ representa o conjunto das nuvens de pontos dentro do intervalo de início e término de cruzamento, for superior a 150, significa que houve uma conversão à esquerda. Se $\sum_{i=a}^b T_y^i R_y^i$ for inferior a -150, representa que houve uma conversão à direita e se $\sum_{i=a}^b T_y^i R_y^i$ for inferior a 150 e superior a -150, significa que o veículo prosseguiu em frente após passar por um cruzamento. O valor 150 foi obtido empiricamente.

Figura 6.8 Mudança de direção do veículo no teste 1. A curva em preto é o produto entre R_y e T_y . Os valores positivos da curva em azul representam os locais de interseções. O eixo x representa os frames extraídos do sensor e o eixo y não tem dimensão. As letras F, D e E representam seguir em frente, conversão à direita e conversão à esquerda, respectivamente.

No segundo teste, o veículo trafegou pela mesma área, porém seguiu por uma trajetória distinta, passando por 8 interseções percorrendo cerca de 1 km, conforme apresentado na figura 6.9. Como no teste anterior, o sistema acertou todas as mudanças de direção feitas pelo veículo e conseguiu localizar o veículo em relação ao mapa topológico. A figura 6.10 apresenta a curva gerada pela multiplicação entre R_y e T_y .

Figura 6.12 Mudança de direção do veículo no teste 3

7 DISCUSSÕES E CONCLUSÃO

Esta tese apresentou um sistema de identificação de obstáculos e de localização baseado em um mapa topológico. Para tal usou os dados coletados por um sensor laser 3D. O método apresentado não utiliza GPS e não requer nenhum mapa digital prévio do terreno. O mapa topológico pode ser feito a partir de um mapa viário comum, ou até mesmo a partir de um esboço do terreno.

Tendo a capacidade de se localizar e identificar os obstáculos, o veículo tem condições de navegar, pois pode fazer o planejamento de rotas por meio de um mapa topológico e se localizar em relação a este e desviar de todos os obstáculos ao seu redor. Uma das limitações deste sistema é a incapacidade de entender a sinalização de trânsito. Contudo, em certas aplicações militares, isso pode não ser necessário. Além disso, o uso de uma câmera de vídeo poderia resolver essa limitação.

7.1 CONTRIBUIÇÕES

O grande diferencial deste trabalho é a capacidade de determinar a localização do veículo sem o uso de GPS, fazendo uso de apenas um único sensor (seção 6).

As demais contribuições são o método de detecção dos pontos pertencentes ao chão (subseção 5.1), um novo método para extração de *features* com a finalidade de classificar obstáculos (redução de uma nuvem de pontos 3D em um *grid* 2D, subseção 5.2) e um método para identificar o início e o fim de uma interseção de vias (subseção 5.3.2).

7.2 PROBLEMAS E MELHORIAS DO SISTEMA DE IDENTIFICAÇÃO DE OBSTÁCULOS E ESTRUTURAS

A detecção dos pontos pertencentes ao solo é fundamental para o desenvolvimento deste trabalho, já que o mesmo é um precursor dos algoritmos de identificação de obstáculos e localização. Esta tese desenvolveu um método para a detecção dos pontos pertencentes ao solo, o qual funcionou de forma eficiente mesmo quando testado em tempo real nos casos dos algoritmos de identificação de obstáculos e de identificação da geometria da pista. Na literatura, há outros métodos desenvolvidos para o mesmo fim. Não foi realizado um teste específico para comparar o método desenvolvido por esta tese com os demais. Um trabalho futuro poderia verificar isso.

O método de classificação de obstáculos, apresentado na subseção 5.2 foi realizado baseado em apenas um *frame*. A classificação pode ser aprimorada ao se analisar um conjunto de *frames* e não apenas um único.

Na subseção 5.3.2 foi apresentado um método para detecção de interseção de vias. Inicialmente, extraem-se as *features* e utiliza-se o classificador AdaBoost para fazer uma classificação primária. Este classificador foi escolhido, pois teve um melhor desempenho em relação ao SVM e ao ANN. Numa segunda fase, foi comparado o HMM e CRF. Este último foi o selecionado porque melhorou o desempenho e diminuiu o ruído, gerando um pico subida e descida para caracterizar o início e fim de uma interseção. Não se encontrou na literatura nenhum trabalho similar que objetivasse esse tipo de saída para identificar o início e fim da interseção de vias. Para validar este método, foram empregados dois conjuntos de dados distintos, um coletado em São Carlos e outro em Karlsruhe na Alemanha. Este último é um repositório de acesso livre ao público.

Este método teve um desempenho abaixo do desejado especialmente em terrenos onde havia poucos obstáculos verticais. Nesse caso, uma sugestão de melhoria seria inserir mais um conjunto de *features*, como por exemplo, a saída do *Curb Detector* (subseção 5.3.1), que é um método que consegue identificar as guias das vias urbanas.

7.3 PROBLEMAS E MELHORIAS DO SISTEMA DE LOCALIZAÇÃO

O sistema de localização desta tese, apresentado na seção 6, pode ser muito útil, especialmente, em situações em que não se pode ou não se deve contar com o sinal de GPS e em que se dispõe de um único sensor, o qual é capaz de gerar uma nuvem de pontos.

O sistema de localização obteve pleno êxito durante a realização dos três testes. Ele foi capaz de identificar todos os cruzamentos e de encontrar sua posição em relação ao mapa topológico durante todo o percurso. Contudo, esta forma de localização tem um ponto fraco, ela é totalmente dependente da correta identificação das interseções de vias.

Além do já citado na subseção 7.1, o uso de um odômetro simples para medir os deslocamentos do veículo poderia trazer uma grande melhoria na identificação de interseção de vias. O conhecimento prévio das distâncias entre as interseções, o que pode ser extraído de um mapa viário, e o conhecimento da distância percorrida pelo

veículo por meio de um odômetro podem ajudar na melhor caracterização das interseções.

A odometria em conjunto com o algoritmo de classificação das formas de vias pode trazer uma elevada melhoria no sistema de localização. O CRF poderia ser utilizado para agregar essas informações para caracterizar melhor as diferenças entre os cruzamentos.

7.4 COMENTÁRIOS FINAIS

Durante o desenvolvimento dessa pesquisa, verificou-se que a grande maioria dos trabalhos tem concentrado seus esforços no uso de uma localização de grande precisão, utilizando-se para isso um GPS com erro na casa dos centímetros e um mapa digital métrico, o qual agrega muitas informações sobre o terreno a ser percorrido, como a posição de obstáculos estáticos e interseções de vias e informação sobre a velocidade permitida na pista.

Identificou-se uma lacuna nessa abordagem no caso em que não se pode utilizar o GPS ou não se dispõe de um mapa digital prévio. Nesse sentido, foi desenvolvido um sistema de localização que não depende dessas variáveis. Provou-se que é possível realizar e foram propostos novos métodos para fazer uma localização confiável por meio de *landmarks*. No caso desta tese, as interseções de vias foram os *landmarks* escolhidos para servir como base no sistema de localização.

Mostrou-se que o sensor laser 3D tem uma grande versatilidade, podendo ser empregado para identificar obstáculos, estruturas e identificar a mudança de direção do veículo. O preço desse tipo de sensor é ainda bastante elevado. Contudo, num período de menos de 10 anos, a empresa que desenvolveu o sensor Velodyne 64-HDL apresentou um novo modelo com o preço dez vezes menor. Sabendo que grandes empresas que desenvolvem veículos não tripulados estão usando este tipo de sensor, não podemos desconsiderar o fato desses sensores se tornarem mais baratos em um futuro breve.

Por fim, esta tese possibilitou a proposta, a implementação e o teste de um sistema de localização, o qual permite a navegação de um veículo autônomo em vias urbanas, sem o uso de um GPS, sendo relevante em operações militares, onde o sinal de GPS pode não estar disponível.

REFERÊNCIAS BIBLIOGRÁFICAS

- ADAMS, Thomas, US Army (2000). “10 GPS Vulnerabilities”. <http://www.c4i.org/gps-adams.html>, acessado pela última vez em 11 de novembro de 2015.
- AMARANTE, J.C.A., “A Batalha Automatizada: Um Sonho Exequível?”, Caderno de Estudos Estratégicos da Escola Superior de Guerra, Junho de 2010.
- ATREYA, A.; Cattle, B.; Collins, B. (2007), “DARPA Urban Challenge - Princeton University - Technical Paper”, Princeton University.
- BADINO, H., Huber, D.; Kanade, T. (2011). “Visual Topometric Localization”. IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 2011.
- BAYERL, Sebastian F.X; Wuensche, Hans-Joachim (2014). “Detection and Tracking of Rural Crossroads Combining Vision and LiDAR Measurements”. 17th International Conference on Intelligent Transportation Systems (ITSC).
- BORIO, D.; O’Driscoll, C.; Fortuny, J. (2013). “Jammer impacto n Galileo and GPS receivers”. International Conference on Localization and GNSS, 2013, pages 1-6.
- BORGEFORS, G. (1986). “Distance transformations in digital images,” in Computer Vision, Graphics and Image Processing, vol. 34, 1986, pp. 344–371.
- BRAGA, A. P.; Carvalho, A. P. L. F.; Ludermir, T. B. Redes neurais artificiais – teoria e aplicações. 2a ed. LTC, 2007.
- BUEHLER, M.; Iagnemma, K.; Singh, S. The DARPA Urban Challenge autonomous vehicles in city traffic. [S.l.]: Berlin Springer, 2009.
- CAUGVT - Technology Development for Army Unmanned Ground Vehicles, (2002), Committee on Army Unmanned Ground Vehicle Technology, Board on Army Science and Technology Division on Engineering and Physical Sciences; The National Academies Press, Washington, D.C.
- DARPA, “DARPA Urban Challenge”, site visitado pela última vez em janeiro de 2014, In <http://archive.darpa.mil/grandchallenge/index.asp>
- DARPA, “Urban Challenge Technical Evaluation Criteria”. DARPA Urban Challenge Documentation. Arlington, VA. March, 2007.
(http://www.darpa.mil/grandchallenge/docs/Technical_Evaluation_Criteria_031607.pdf).
- DARPA, “Urban Challenge Technical Evaluation Criteria”. DARPA Urban Challenge Documentation.
(http://www.darpa.mil/grandchallenge/docs/Technical_Evaluation_Criteria_031607.pdf). (2007).

- DENATRAN : Anuário Estatístico do Departamento Nacional de Transito DENATRAN - RENAEST 2006, www.denatran.gov.br - Portal RENAEST.
- DOUILLARD, J.; Morton, P.; Underwood, Bailey, T. (2012), “Scan Segments Matching for Pairwise 3D Alignment”. ICRA 2012.
- DOUILLARD, J.; Underwood, V; Kuntz, N.; Vlaskine, V; Quadros, S; Singh, S. (2010), “A Pipeline for the Segmentation and Classification of 3D Point Clouds”.
- DOUILLARD, J.; Underwood, V; Kuntz, N.; Vlaskine, V; Quadros, S; Morton, P.; Frenkel, A. (2011), “On the Segmentation of 3D Lidar Point Clouds”, 2011 IEEE International Conference on Robotics Automation.
- DUDEK, Gregory; Jenkin, Michael. (2010). “Computational Principles of Mobile Robotics”. Cambridge University Press, Second Edition, 2010.
- FELZENSZWALB, P.F.; Huttenlocher, D.P.(2004). Efficient belief propagation for early vision Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, 2004, Volume: 1, Pages: I-261 - I-268 Vol.1, DOI: 10.1109/CVPR.2004.1315041.
- FERNANDES, Leandro; Jefferson R. Souza, Gustavo Pessin, Patrick Y. Shinzato, Daniel Sales, Caio Mendes, Marcos Prado, Rafael Klaser, André Chaves Magalhães, Alberto Hata, Daniel Pigatto, Kalinka Castelo Branco, Valdir Grassi, Fernando S. Osorio, Denis F. Wolf .(2014). “CaRINA Intelligent Robotic Car: Architectural design and applications”, Journal of Systems Architecture, Abril de 2014.
- FERREIRA, Leonardo Nascimento. “Técnicas de agrupamento de dados baseada em redes complexas para o posicionamento de cluster heads em redes de sensores sem fio”. Dissertação de mestrado do ICMC-USP, 2012.
- FISCHLER, M. A.; BOLLES, R. C. (1981). “Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. Commun. ACM, v. 24, p. 381-395, 1981.
- FRACASSO, Paulo Thiago; Reali Costa, Anna Helena. (2005). “Navegação Reativa De Robôs Móveis Autônomos Utilizando Lógica Nebulosa Com Regras Ponderadas”. VII SBAI/ II IEEE LARS, 2005.
- FREUND, Y., Schapire, R.E. (1999). “A Short Introduction to Boosting”. Journal of Japanese Society for Artificial Intelligence.
- FREITAS, G., Bradley Hamner, Marcel Bergerman and Sanjiv Singh (2012). “A Practical Obstacle Detection System for Autonomous Orchard Vehicles”. IEEE/RSJ International Conference on Intelligent Robots and Systems.

- GONZALEZ, R.C., Woods, R.E. “Processamento de Imagem digitais”. Editora Edgard Blücher, 1ª Edição, 2000.
- GOOGLE Driverless Car (http://en.wikipedia.org/wiki/Google_driverless_car, acessada em 22 de julho de 2013)
- GRANSTRÖM, K.; Schön, T. B.; Nieto, J. I. e Ramos, F. (2011). “Learning to close loops from range data.” I. J. Robotic Res., vol. 30, no. 14, pp. 1728–1754, 2011.
- GUIZZO, Erico (2011). “How Google’s Self-Driving Car Works”. Publicado na revista IEEE Spectrum em 18 de outubro de 2011.
- HABERMANN, D., Bragança, R., Wolf, D., Osório, F.S. (2013a) “Detecção e Classificação de Objetos com uso de Sensor Laser para Aplicações em Veículos Autônomos Terrestres”. Simpósio de Aplicações Operacionais em Áreas de Defesa (SIGE 2013).
- HABERMANN, D., Hata, A., Wolf, D., Osório, F.S. (2013b). “3D Point Clouds Segmentation for Autonomous Ground Vehicle”. Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC 2013).
- HABERMANN, D., Hata, A., Wolf, D., Osório, F.S. (2013 c) “Artificial Neural Nets object recognition for 3D point clouds”. 2º Brazilian Conference on Intelligent Systems (BRACIS 2013).
- HABERMANN, D.; Vido, C.; Osorio, F.S.; Ramos, F. (2016a). “Road Junction Detection from 3D Point Clouds”. International Joint Conference on Neural Networks (IJCNN-2016).
- HABERMANN, D., Hata, A., Shinzato, P.; Wolf, D., Osório, F.S. (2016b) “Artificial Neural Network for 3D Object Recognition applied to Autonomous Vehicles”. Journal of Engineering Applications of Artificial Intelligence (“submetido”).
- HARRIS, M. (2015). “A cheaper way for robocars to avoid pedestrians”. IEEE Spectrum, ano 2015, volume 52.
- HATA, A., Habermann, D., Wolf, D., Osório, F. “Crossroad Detection Using Artificial Neural Nets”. 14º Engineering Applications of Neural Networks (2013).
- HATA, A.; Habermann, D.; Osorio, F.; Wolf, D. (2014). “Road geometry classification using ANN”. Intelligent Vehicles Symposium Proceedings, 2014 IEEE, 1319-1324.
- HAYKIN, S. “Neural Networks: A Comprehensive Foundation”, MacMillan Publishing Company (1994).
- HIMMELSBACH, M., Hundelshausen, F.v. and Wuensche, H.J.: Fast Segmentation of 3D Point Clouds for Ground Vehicles. 2010 IEEE Intelligent Vehicles Symposium,

- University of California, San Diego, CA, USA, June 21-24, 2010.
- HOLMES, W.I.H.; DONKIN, G. (1994). “Weka: a machine learning workbench”. Proceedings of the Second Australian and New Zealand Conference in Intelligent Information Systems, 1994, n° 0, 1994, pp 357-361.
- HUI, L., Liping Di, Huang Xianfeng, Li Deren. “Laser intensity used in classification of lidar point cloud data”. IEEE 2008.
- JORDAN, Michael I. (2002). “An Introduction to Probabilistic Graphical Models”, University of California, Berkeley.
- KLASING, K., Wollherr, D., Buss M. (2008). A Clustering Method for Efficient Segmentation of 3D Laser Data. IEEE International Conference on Robotics and Automation (ICRA 2008), pages 4043 - 4048.
- KLEINBERG, Jon. (2002). “An Impossibility Theorem for Clustering”, 2002. MIT Press.
- KOLLER, D., Friedman, N. (2009) “Probabilistic Graphical Models: Principles and Techniques”. MIT Press.
- KUTHIRUMMAL, S., Das, A. and Samarasekera, S.: A Graph Traversal based Algorithm for Obstacle Detection using Lidar or Stereo. IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.
- ZOU, Li-Huy, Chen Jie, Zhang, J. Dou, L (2008). “The Comparison of Two Typical Corner Detection Algorithms”. Intelligent Information Technology Application, 2008. IITA '08. Second International Symposium on.
- LUETTEL, T.; Himmelsbach, M.; Wuensche, H.-J. Autonomous Ground Vehicles—Concepts and a Path to the Future. Proceedings of the IEEE. May 2012. Vol: 100, Issue: Special Centennial Issue. Page(s): 1831 – 1839.
- MENDES, C.; (2012). “Navegação de robôs movidos utilizando visão stereo”. Dissertação de mestrado do ICMC-USP.
- MIRANDA NETO, A., ZAMPIERI, D., (2008) Sistema de Navegação Semi-Autônomo de Assistência ao Condutor (SAC), UNICAMP.
- MITCHELL, T. “Machine Learning”, McGraw Hill, 1997.
- MUELLER, A.; Himmelsbach, M.; Luettel, T.; Hundelshausen, F. and Hans-Joachim Wuensche (2011). “GIS-Based Topological Robot Localization through LIDAR Crossroad Detection”. IEEE Conference on Intelligent Transportation Systems.
- MURPHY, Kevin P. (2005). <http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf>
- MURPHY, Kevin P. (2012). “Machine Learning – A probabilistic Perspective”, 2012, MIT Press, ISBN 978-0-262-01802-9.

- NASAW, Daniel. (2012), “Driverless cars and how they would change motoring”, BBC News Magazine, Washington- <http://www.bbc.co.uk/news/magazine-18012812>, acessado em agosto de 2013.
- NAVAL Operations - Autonomous Vehicles in Support of Naval Operations, Committee on Autonomous Vehicles in Support of Naval Operations, National Research Council; ISBN: 0-309-55115-3, 256 pages,(2005).
- NSW – New South Wales Government – Australia (2012) . “Road safety in new south wales: Statistical statement for the year ended in 31 december 2012”. [Online]. Available: <http://roadsafety.transport.nsw.gov.au/downloads/crashstats2012.pdf>.
- PALI, Vivek; Goswami, S.; Bhayia, L. (2014).”Na Extensive Survey on Feature Extraction Techniques for Facial Image Processing”. VI International Conference on Computational Intelligence and Communication Networks.
- PEREIRA, F. L; “Sistemas e Veículos Autônomos”. Instituto de Defesa Nacional. Portugal, 2005.
- ROJAS,R.; (2005) “Pattern Recognition for Autonomous Driving”, Equipe Spirit of Berlin, Freie Universität Berlin.
- RUSSELL, Stuart; Norvig, Peter. (1995). “Artificial Intelligence – A modern Approach”. Ed Pretince Hall, 1995.
- SALES, Daniel Oliva; Leandro Carlos Fernandes, Fernando Santos Osório, Denis Fernando Wolf. “FSM-based Visual Navigation for Autonomous Vehicles”. IROS Workshop on Visual Control of Mobile Robots (ViCoMoR 2012).
- SAMPLES, M; JAMES, M. R.; “Learning a Real-Time 3D Point Cloud Obstacle Discriminator via Bootstrapping”; Toyota Research Institute, North America.
- SAMPLES, Michael; James, Michael R. (2010). “Learning a Real-Time 3D Point Cloud Discriminator via Bootstrapping”.
- SCHMIDT, M.; K. Swersky. (2008) CRFChain: Matlab code for chain-structured conditional random fields with categorical features.
<http://www.cs.ubc.ca/~schmidtm/Software/crfChain.html>, 2008.
- SCHNEIDER, F.E.; Wildermuth, D.; Wolf, H. (2012). “Professional ground robotic competitions from an educational perspective: A consideration using the example of the European Land Robot Trial(ELROB)”. Intelligent Systems (IS), 2012 6th IEEE International Conference, 2012.
- SCHÖLKOPF, B., Smola, A.J. (2001). “Learning with Kernels”, MIT Press, 2001.
- SEGAL, A., Hähnel, D., Thrun S.: Generalized-ICP. Robotics: Science and Systems 2009.

- SHINZATO, Patrick Yuri (2015). "Estimation of obstacles and road area with sparse 3D points". Tese de doutorado apresentada ao Instituto de Ciências de Computação e Matemática Computacional da Universidade de São Paulo.
- SUTTON, C., McCallum, A. (2011) "An Introduction to Conditional Random Fields", Foundations and Trends in Machine Learning, Vol. 4, No. 4 (2011) 267-373, DOI: 10.1561/22000000013.
- TATOGLU, A., Pochiraju, K. "Point Cloud Segmentation with LIDAR Reflection Intensity Behavior". IEEE International Conference on Robotics and Automation (2012).
- TEICHMAN, A. ; Levinson, J. ; Thrun, S. "Towards 3D object recognition via classification of arbitrary object tracks" (ICRA 2011).
- THRUN, S., Montemerlo, M., Dahlkamp, H., Stavens, Aron, D.A., Diebel, J., Fong, P. (2006.a) "Stanley: The robot that won the DARPA Grand Challenge: Research Articles," J. Robot. Syst., vol. 23, no. 9, pp. 661–692, 2006., Journal of Field Robotics 23(9), pag 661–692.
- THRUN, S.; Burgard, W.; Fox, D. (2006.b). "Probabilistic Robotics". MIT Press.
- TONGTONG, Chen, Dai Bin, Liu Daxue (2011). "3D LIDAR-based Ground Segmentation".
- URMSON, C.; ANHALT, J.; BAGNELL, D., (2008), "Autonomous Driving in Urban Environments: Boss and the Urban Challenge", Journal of Field Robotics 25(8), 425–466 (2008).
- VELODYNE HDL64E S2 – User's Manual and Programming Guide (63 HDL64E S2 Rev D, Maio 2011).
- WANG, D.; I. Posner, P. Newman, (2012). "What could move? Finding cars, pedestrians and bicyclists in 3d laser data", in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, 2012, pp. 4038-4044.
- WILLIAMS, M. (1992). "The PROMETHEUS Programme". Towards Safer Road Transport - Engineering Solutions, IEE Colloquium on, 1992, pp. 4/1 - 4/3.
- WOLF, D., Simões, E.V., Osório. F.S., Trindade Jr, O. (2009). "Robótica Móvel Inteligente: Da Simulação às Aplicações no Mundo Real", disponível em: <http://www.inct-sec.org/actrep/sites/default/files/highlights/Tutorial-JAI.pdf>
- ZHAN, Q. , Yu, L (2011) "Objects classification from laser scanning data based on multi-class support vector machine", in: Remote Sensing, Environment and Transportation Engineering (RSETE), 2011 International Conference on, 2011, pp. 520-523.

- ZHOU, W. "An Object-Based Approach for Urban Land Cover Classification: Integrating LiDAR Height and Intensity Data". IEEE Geoscience and Remote Sensing Letters, vol. 10, no. 4, july, 2013.
- ZHU, Quanwen , Long Chen ; Qingquan Li ; Ming Li. (2012). "3D LIDAR Point Cloud based Intersection Recognition for Autonomous Driving". Intelligent Vehicles Symposium (IV), 2012.
- ZHU,X.; ZHAO, H.; LIU, Y.; ZHAO,Y.; ZHA, H., (2010), " Segmentation and classification of range image from an intelligent vehicle in urban environment",In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010.

APÊNDICE I: Extração de features de nuvem de pontos 3D

```
% =====
% obtaining features from a 3d laser point cloud
%=====
function [features]=features_laser(nuvemPontos)

k=0;
i=2 ;
nuvem=nuvemPontos;
%=====
%Parameters:
N=length(nuvem);
Rmax= 25;
Gdist=2.5;
%=====

for i=1:N
    nuvem(i,4) = sqrt(sum(nuvem(i,1:3) .^ 2));
end
soma1=0; soma2=0; soma3=0;soma4=0;
soma10=0;
j=0;
for i=1:N
    soma1 = soma1+( (nuvem(i,4)/Rmax)^ 3);
    soma4=soma4+(nuvem(i,4));
    RnVector1(i)=nuvem(i,4)/Rmax;% f6
    if (nuvem(i,4)<Rmax)
        soma2=soma2+( (nuvem(i,4)/Rmax)^ 3);
        soma3=soma3+(nuvem(i,4));
        j=j+1;
        RnVector2(j)=nuvem(i,4)/Rmax;% f7
        f10Matrix(j,:)=nuvem(i,1:3);
    end
end
f1=soma1/N; f2=soma2/j;f3=soma3/(j*Rmax);f4=soma4/(N*Rmax);
MeanRange_f21=soma3/j;
MeanRange_f22=soma4/N;
N_f21=j;
N_f22=N;
f5=std(RnVector2,1);
disp(mean(RnVector1))
disp(sum(RnVector1))
f6=std(RnVector1);
mf10=mean(f10Matrix);
f10=(sum(mf10.^2) )^(1/2);
f13=N-j;
f14=j;

soma10=0;
j=0;

for i=1:length(f10Matrix)
    j=j+1;
    f11_f12Matrix(j)=sqrt(sum((f10Matrix(j,:) - mf10) .^ 2));
end
```

```

f11=mean(f11_f12Matrix);
f12=std(f11_f12Matrix);

f16=0;
f15=0;
f17=0;
j=0;
k=0;
for i=1:N-2

    if ((nuvem(i,4)<Rmax) && (nuvem(i+1,4)<Rmax))
        f16dist=sqrt(sum((nuvem(i,1:3) - nuvem(i+1,1:3)) .^ 2));
        f16=f16+ f16dist;
        j=j+1;
        f18Vector(j)=f16dist;
        if (f16dist<Gdist)
            f17=f17+f16dist;
        end
    end

    if ((nuvem(i,4)<Rmax) && (nuvem(i+1,4)<Rmax) &&
(nuvem(i+2,4)<Rmax))
        d1=sqrt(sum((nuvem(i,1:3) - nuvem(i+1,1:3)) .^ 2));
        d2=sqrt(sum((nuvem(i+1,1:3) - nuvem(i+2,1:3)) .^ 2));
        d3=sqrt(sum((nuvem(i,1:3) - nuvem(i+2,1:3)) .^ 2));
        s=(d1+d2+d3)/2;%semiperimetro do triangulo
        AreaTriangle=( s*(s-d1)*(s-d2)*(s-d3) )^(1/2);
        %Heron formula for triangle area

        if ((d1<Gdist) && (d2<Gdist) && (d3<Gdist))
            k=k+1;
            curvature(k)=4*AreaTriangle/(d1*d2*d3);
        end
    end

    f15=f15+ sqrt(sum((nuvem(i,1:3) - nuvem(i+1,1:3)) .^ 2));
end
f18=std(f18Vector);
f19=mean(curvature); %obs f19 e f20 estão errados....refazer
f20=std(curvature);

soma_m4_f22=0;
soma_m2_f22=0;
soma_m4_f21=0;
soma_m2_f21=0;
j=0;
k=0;
l=0;
for i=1:N
    soma_m4_f22=soma_m4_f22+ (nuvem(i,4) - MeanRange_f22)^4;
    soma_m2_f22=soma_m2_f22+ (nuvem(i,4) - MeanRange_f22)^2;

```

```

        if (nuvem(i,4)<Rmax)
            soma_m4_f21=soma_m4_f21+ (nuvem(i,4) -
MeanRange_f21)^4;
            soma_m2_f21=soma_m2_f21+ (nuvem(i,4) -
MeanRange_f21)^2;
            if (i<N)
                j=j+1;
                Vector_f25(j)=nuvem(i,4)/nuvem(i+1,4);
                Vector_f27(j)=abs(nuvem(i,4)-nuvem(i+1,4));
            end
        end

        if ( (nuvem(i,4)<Rmax*0.75) && (i<N) )
            k=k+1;
            Vector_f29(k)=abs(nuvem(i,4)-nuvem(i+1,4));
        end

        if ( (nuvem(i,4)<Rmax*0.50) && (i<N) )
            l=l+1;
            Vector_f31(l)=abs(nuvem(i,4)-nuvem(i+1,4));
        end

        if (i<N)
            Vector_f23(i)=nuvem(i,4)/nuvem(i+1,4);
        end
    end

    end

f22=( ((1/N_f22)*soma_m4_f22)/ ((1/N_f22)*soma_m2_f22)^2 )-3;
f21=( ((1/N_f21)*soma_m4_f21)/ ((1/N_f21)*soma_m2_f21)^2 )-3;
f23=mean(Vector_f23);
f24=std(Vector_f23);
f25=mean(Vector_f25);
f26=std(Vector_f25);
f27=mean(Vector_f27);
f28=std(Vector_f27);
f29=mean(Vector_f29);
f30=std(Vector_f29);
f31=std(Vector_f31);
f32=std(Vector_f31);
f33=hist(nuvem(:,4),[ 1 1.5 3 5 7 9 12 18 24]);
% f33 armazena um vetor com 9 elementos

features1=[f1 f2 f3 f4 f5 f6 f10 f11 f12 f13 f14 f15 f16 f17 f18
f19 f20 f21 f22 f23 f24 f25 f26 f27 f28 f29 f30 f31 f32];
features=[features1 f33];

disp('fim');

```


APÊNDICE II:
MAPA TOPOLÓGICO OBTIDO DO MAPA VIÁRIO DA FIGURA 6.1

```
% criação de um mapa topológico
function [ origem_2, destino_2] =
mapa_topologico(origem_0,destino_0, direcao )

%% destino C1 e origem C9
if strcmp(origem_0,'c9') && strcmp(destino_0,'c1')
    if strcmp(direcao,'frente')
        origem='c1';
        destino='c2';
    end
    if strcmp(direcao,'direita')
        origem='c1';
        destino='c8';
    end
    if strcmp(direcao,'esquerda')
        origem='c1';
        destino='F5';
    end
end

%% destino c1 origem c2
if strcmp(origem_0,'c2') && strcmp(destino_0,'c1')
    if strcmp(direcao,'frente')
        origem='c1';
        destino='c9';
    end

    if strcmp(direcao,'direita')
        origem='c1';
        destino='F3';
    end
    if strcmp(direcao,'esquerda')
        origem='c1';
        destino='c8';
    end
end

%% destino c1 origem c8
if strcmp(origem_0,'c8') && strcmp(destino_0,'c1')
    if strcmp(direcao,'frente')
        origem='c1';
        destino='F3';
    end

    if strcmp(direcao,'direita')
        origem='c1';
        destino='c2';
    end
    if strcmp(direcao,'esquerda')
        origem='c1';
        destino='c9';
    end
end
```

```

%% destino c2 origem c1
if strcmp(origem_0,'c1') && strcmp(destino_0,'c2')

    if strcmp(direcao,'frente')
        origem='c2';
        destino='c3';
    end

    if strcmp(direcao,'direita')
        origem='c2';
        destino='c6';
    end
    if strcmp(direcao,'esquerda')
        origem='c2';
        destino='F6';
    end
end

%% destino c2 origem c6
if strcmp(origem_0,'c6') && strcmp(destino_0,'c2')

    if strcmp(direcao,'frente')
        origem='c2';
        destino='f6';
    end

    if strcmp(direcao,'direita')
        origem='c2';
        destino='c3';
    end
    if strcmp(direcao,'esquerda')
        origem='c2';
        destino='c1';
    end
end

%% destino c3 origem c2
%{
if strcmp(origem_0,'c6') && strcmp(destino_0,'c2')
    disp('ola 6');
    if strcmp(direcao,'frente')
        origem='c2';
        destino='F6';
    end

    if strcmp(direcao,'direita')
        origem='c2';
        destino='c3';
    end
    if strcmp(direcao,'esquerda')
        origem='c1';
        destino='F7';
    end
end
end

```

```

%}

%% destino c3 origem c6
%{
if strcmp(origem_0,'c6') && strcmp(destino_0,'c2')
    disp('ola 7');
    if strcmp(direcao,'frente')
        origem='c2';
        destino='F6';
    end

    if strcmp(direcao,'direita')
        origem='c2';
        destino='c3';
    end
    if strcmp(direcao,'esquerda')
        origem='c1';
        destino='F7';
    end
end
%}
%% destino c3 origem c2
if strcmp(origem_0,'c2') && strcmp(destino_0,'c3')

    if strcmp(direcao,'frente')
        origem='c3';
        destino='F1';
    end

    if strcmp(direcao,'direita')
        origem='c3';
        destino='c4';
    end
    if strcmp(direcao,'esquerda')
        origem='c3';
        destino='F7';
    end
end

%% destino c4 origem c3
if strcmp(origem_0,'c3') && strcmp(destino_0,'c4')
    disp('ola 9');
    if strcmp(direcao,'frente')
        origem='c4';
        destino='c5';
    end

    if strcmp(direcao,'direita')
        origem='c4';
        destino='c6';
    end
    if strcmp(direcao,'esquerda')
        origem='c4';
        destino='F2';
    end
end
end

```

```

%% destino c5 origem c4
if strcmp(origem_0,'c4') && strcmp(destino_0,'c5')
    disp('ola 10');
    if strcmp(direcao,'frente')
        origem='c5';
        destino='F3';
    end

    if strcmp(direcao,'direita')
        origem='c5';
        destino='c7';
    end
    if strcmp(direcao,'esquerda')
        origem='c5';
        destino='=== ERRO ===';
    end
end
end

```

```

%% destino c6 origem c4
if strcmp(origem_0,'c4') && strcmp(destino_0,'c6')
    disp('ola 11 ');
    if strcmp(direcao,'frente')
        origem='c6';
        destino='F8';
    end

    if strcmp(direcao,'direita')
        origem='c6';
        destino='c2';
    end
    if strcmp(direcao,'esquerda')
        origem='c6';
        destino='c7';
    end
end
end

```

```

origem_2=origem;
destino_2=destino;

```

```

end

```