
Detecção de obstáculos usando fusão de dados de percepção 3D e radar em veículos automotivos

Luis Alberto Rosero Rosero

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Luis Alberto Rosero Rosero

Detecção de obstáculos usando fusão de dados de percepção 3D e radar em veículos automotivos

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Fernando Santos Osório

USP – São Carlos
Abril de 2017

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

R795d Rosero, Luis Alberto Rosero
Detecção de obstáculos usando fusão de dados
de percepção 3D e radar em veículos automotivos
/ Luis Alberto Rosero Rosero; orientador Fernando
Santos Osório. – São Carlos – SP, 2017.
85 p.

Dissertação (Mestrado – Programa de Pós-Graduação
em Ciências de Computação e Matemática Computacional)
– Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2017.

1. calibração de sensores. 2. detecção de
obstáculos. 3. fusão de sensores. 4. LiDAR 3D.
5. Radar. 6. visão computacional. 7. veículos
autônomos. I. Osório, Fernando Santos, orient. II.
Título.

Luis Alberto Rosero Rosero

Obstacle detection using 3D perception and radar data fusion in automotive vehicles

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master 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
April 2017

AGRADECIMENTOS

Agradeço a todos que de alguma forma ajudaram na execução deste projeto, especialmente agradeço ao meu orientador, o Professor Fernando Santos Osório pela oportunidade de trabalhar com ele e no grupo e principalmente pela amizade e o apoio em todo momento.

Quero agradecer a minha família, meus pais Cardenio e Lucia, meus irmãos Carlos e Dario, eles são o apoio e incentivo diário e o motor que me faz avançar a cada dia.

Agradeço também a todos os membros do Laboratório de Robótica Móvel (LRM) pela ajuda e companheirismo durante este tempo. Obrigado a todos os amigos criados em São Carlos pelos momentos de alegria que guardarei sempre.

Por fim, agradeço o apoio financeiro concedido pela INCT-SEC, Scania LA (Latin America), FIPAI, CNPq e CAPES.

RESUMO

ROSERO, L. A.. Detecção de obstáculos usando fusão de dados de percepção 3D e radar em veículos automotivos. 2017. 85 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Este projeto de mestrado visa a pesquisa e o desenvolvimento de métodos e algoritmos, relacionados ao uso de radares, visão computacional, calibração e fusão de sensores em veículos autônomos/inteligentes para fazer a detecção de obstáculos. O processo de detecção de obstáculos se divide em três etapas, a primeira é a leitura de sinais de Radar, do LiDAR e a captura de dados da câmera estéreo devidamente calibrados, a segunda etapa é a fusão de dados obtidos na etapa anterior (Radar+câmera, Radar+LiDAR 3D), a terceira etapa é a extração de características das informações obtidas, identificando e diferenciando o plano de suporte (chão) dos obstáculos, e finalmente realizando a detecção dos obstáculos resultantes da fusão dos dados. Assim é possível diferenciar os diversos tipos de elementos identificados pelo Radar e que são confirmados e unidos aos dados obtidos por visão computacional ou LiDAR (nuvens de pontos), obtendo uma descrição mais precisa do contorno, formato, tamanho e posicionamento destes. Na tarefa de detecção é importante localizar e segmentar os obstáculos para posteriormente tomar decisões referentes ao controle do veículo autônomo/inteligente. É importante destacar que o Radar opera em condições adversas (pouca ou nenhuma iluminação, com poeira ou neblina), porém permite obter apenas pontos isolados representando os obstáculos (esparsos). Por outro lado, a câmera estéreo e o LiDAR 3D permitem definir os contornos dos objetos representando mais adequadamente seu volume, porém no caso da câmera esta é mais suscetível a variações na iluminação e a condições restritas ambientais e de visibilidade (p.ex. poeira, neblina, chuva). Também devemos destacar que antes do processo de fusão é importante alinhar espacialmente os dados dos sensores, isto é calibrar adequadamente os sensores para poder transladar dados fornecidos por um sensor referenciado no próprio sistema de coordenadas para um outro sistema de coordenadas de outro sensor ou para um sistema de coordenadas global. Este projeto foi desenvolvido usando a plataforma CaRINA II desenvolvida junto ao Laboratório LRM do ICMC/USP São Carlos. Por fim, o projeto foi implementado usando o ambiente ROS, OpenCV e PCL, permitindo a realização de experimentos com dados reais de Radar, LiDAR e câmera estéreo, bem como realizando uma avaliação da qualidade da fusão dos dados e detecção de obstáculos com estes sensores.

Palavras-chave: calibração de sensores, detecção de obstáculos, fusão de sensores, LiDAR 3D, Radar, visão computacional, veículos autônomos.

ABSTRACT

ROSERO, L. A.. Detecção de obstáculos usando fusão de dados de percepção 3D e radar em veículos automotivos. 2017. 85 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

This master's project aims to research and develop methods and algorithms related to the use of radars, computer vision, calibration and sensor data fusion in autonomous / intelligent vehicles to detect obstacles. The obstacle detection process is divided into three stages, the first one is the reading of Radar, LiDAR signals and the data capture of the stereo camera properly calibrated, the second stage is the fusion of data obtained in the previous stage (Radar + Camera, Radar + 3D LIDAR), the third step is the extraction of characteristics of the information obtained, identifying and differentiating the support plane (ground) of the obstacles, and finally realizing the detection of the obstacles resulting from the fusion of the data. Thus it is possible to differentiate types of elements identified by the Radar and that are confirmed and united to the data obtained by computational vision or LIDAR (point cloud), obtaining a more precise description of the contour, format, size and positioning of these. During the detection task it is important to locate and segment the obstacles to later make decisions regarding the control of the autonomous / intelligent vehicle. It is important to note that Radar operates in adverse conditions (little or no light, with dust or fog), but allows only isolated points representing obstacles (sparse), while on the other hand, the stereo camera and LIDAR 3D allow to define the shape and size of objects. As for the camera, this is more susceptible to variations in lighting and to environmental and visibility restricted conditions (eg dust, haze, rain). It is important to spatially align the sensor data, calibrating the sensors appropriately, to be able to translate data provided by a sensor referenced in the coordinate system itself to another coordinate system of another sensor or to a global coordinate system. This project was developed using the CaRINA II platform developed by the LRM Laboratory ICMC / USP São Carlos. Finally, the project was implemented using the ROS, OpenCV and PCL environments, allowing experiments with real data from Radar, LiDAR and stereo camera, as well as performing an evaluation of the quality of the data fusion and detection of obstacles with these sensors .

Key-words: sensor calibration, obstacle detection, sensor fusion, 3D LiDAR, computer vision, autonomous vehicles.

LISTA DE ILUSTRAÇÕES

| | |
|--|----|
| Figura 1 – Fórmulas para o RCS | 24 |
| Figura 2 – Captura de duas imagens simultâneas com câmeras distantes entre si separadas por uma determinada distância/base line | 26 |
| Figura 3 – Distorções da lente | 26 |
| Figura 4 – Mapa de profundidade | 29 |
| Figura 5 – Exemplo da execução do “Sixel” | 29 |
| Figura 6 – Resultado do método de detecção em diferentes condições ambientais | 35 |
| Figura 7 – Resultados do método de detecção de pedestres usado pelos investigadores da Hamburg University of Technology | 35 |
| Figura 8 – Plataformas autônomas projeto CaRINA | 37 |
| Figura 9 – Detecção do plano do chão (em preto) e marcação dos obstáculos (em cores). Visualização (a) e projeção da visualização na cena real (b) | 38 |
| Figura 10 – Resultados do método de detecção de obstáculos no mundo imagem 2D. | 38 |
| Figura 11 – Pose dos sensores no CaRINA II | 42 |
| Figura 12 – Fluxograma do sistema proposto para fusão de sensores | 42 |
| Figura 13 – Modulo Sensor [Delphi ESR] | 43 |
| Figura 14 – Características dos Módulos Sensores – Mid e Long Range [Delphi ESR] | 44 |
| Figura 15 – Exemplo de detecções fornecidas pelo ESR na rodovia | 46 |
| Figura 16 – Camera bumblebee | 47 |
| Figura 17 – velodyne | 48 |
| Figura 18 – Diagrama de blocos do processo de calibração | 50 |
| Figura 19 – Modelo “pin-hole” | 51 |
| Figura 20 – Processo de calibração da câmera estéreo usando padrão xadrez | 51 |
| Figura 21 – Refletor tipo Square Tri hedral corner | 52 |
| Figura 22 – Detecção baseada em HSV do target na imagem 2D | 53 |
| Figura 23 – Detecção de n pontos para entrada do algoritmos PnP usando Radar e Câmera estéreo | 54 |
| Figura 24 – Detecção do retângulo na imagem 2D e na nuvem de pontos 3D do Velodyne | 54 |
| Figura 25 – Detecção de n pontos para entrada do algoritmos PnP usando LiDAR 3D e Câmera estéreo | 56 |
| Figura 26 – Alinhamento dos Sensores: Radar, Câmera Estéreo e LiDAR 3D | 56 |
| Figura 27 – Múltiplos objetos detectados na nuvem de pontos fornecida pelo LiDAR(Pedestres, carros e postes) | 59 |

| | |
|---|----|
| Figura 28 – “Stixel World” para detecção de objetos | 60 |
| Figura 29 – Dados do Radar destacando obstáculos identificados arvores e pedestre | 63 |
| Figura 30 – Oclusão devido a poeira (problema para sistemas baseados em imagens) | 64 |
| Figura 31 – Experimentos realizados para análise da influência de neblina/poeira na detecção de obstáculos usando Radar | 65 |
| Figura 32 – Comparação entre a posição original e a posição relativa obtida na calibração dos sensores (Radar e câmera estéreo) | 67 |
| Figura 33 – Comparação entre a posição original e a posição relativa obtida na calibração dos sensores (LiDAR e câmera estéreo) | 68 |
| Figura 34 – Exemplos de erro de projecção dos pontos detectados pelo Radar | 69 |
| Figura 35 – Exemplos de erro de projecção dos pontos detectados pelo LiDAR 3D | 70 |
| Figura 36 – Detecção de obstáculos usando Radar+Sensor 3D | 76 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – RCS para objetos em frequências de micro-onda | 25 |
| Tabela 2 – Mensagens CAN do modulo Delphi ESR | 45 |
| Tabela 3 – Refletores comumente usados para testar o Radar | 52 |
| Tabela 4 – Resultados (100 pontos) | 72 |
| Tabela 5 – Resultados (altura 50 cm) | 73 |
| Tabela 6 – Resultados (20 pontos) | 73 |
| Tabela 7 – Comparação dos resultados de fusão considerando o tamanho e a quantidade de pontos que representam o obstáculo - Usando Radar e Câmera Estéreo | 73 |
| Tabela 8 – Comparação dos resultados de fusão considerando o tamanho e a quantidade de pontos que representam o obstáculo - Usando Radar e Velodyne | 74 |

LISTA DE ABREVIATURAS E SIGLAS

- ADAS Advanced Driver Assistance Systems
Bumblebee Equipamento de visão estéreo fabricado pela empresa Point Grey
CAN Controller Area Network
CaRINA .. Carro Robótico Inteligente para Navegação Autônoma
DARPA .. Defense Advanced Research Projects Agency
DLT Direct Linear Transform
DoN Difference of normals
DSP Digital Signal Processor
ESR Electronically Scanning Radar
FOV Field of View
HFOV Horozntal Field of View
ICMC Instituto de Ciências Matemáticas e de Computação
LiDAR ... Light Detection and Ranging
LRM Laboratório de Robótica Móvel
MMW Milimeter Wave Radar
OpenCV .. Open Source Computer Vision
PCL Point Cloud Libray
PnP Perspective-n-Point
Radar RAdio Detection And Ranging
RCS Radar Cross Section
RF Rádio Frequênci
ROS Robot Operating System
SGM Semi Global Matching
SVD Single Value Decomposition
USP Universidade de São Paulo
Velodyne . Equipamento do tipo LiDAR 3D de múltiplos feixes laser

SUMÁRIO

| | | |
|---------|--|----|
| 1 | INTRODUÇÃO | 19 |
| 1.1 | Objetivos | 21 |
| 1.1.1 | Objetivo Geral | 21 |
| 1.1.2 | Objetivos específicos | 21 |
| 1.1.3 | Organização do texto | 22 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 23 |
| 2.1 | Radar | 23 |
| 2.1.1 | Radar Cross Section (RCS) | 24 |
| 2.2 | Visão estéreo | 25 |
| 2.2.1 | Modelo “pin hole” da câmera e calibração | 25 |
| 2.2.1.1 | Calibração intrínseca e extrínseca | 26 |
| 2.2.1.2 | Modelo da câmera implementado no OpenCV | 27 |
| 2.2.1.3 | Estimativa da pose da câmera | 27 |
| 2.2.2 | Mapa de disparidade | 28 |
| 2.2.2.1 | Métodos para calcular a disparidade. | 29 |
| 2.2.3 | Stixel World | 30 |
| 2.3 | LiDAR | 30 |
| 2.4 | Fusão de sensores | 31 |
| 2.4.1 | Fusão em nível de dados (Data Level Fusion) | 31 |
| 2.4.2 | Fusão em nível de características (Feature Level Fusion) | 31 |
| 2.4.3 | Fusão em nível de Declaração (Declaration Level Fusion) | 31 |
| 2.5 | Considerações Finais | 32 |
| 3 | TRABALHOS RELACIONADOS | 33 |
| 3.1 | Laboratório de Robótica Móvel (LRM) | 35 |
| 3.2 | Considerações Finais | 39 |
| 4 | METODOLOGIA | 41 |
| 4.1 | Arquitetura para a fusão de sensores | 41 |
| 4.2 | Hardware | 43 |
| 4.2.1 | Radar Automotivo | 43 |
| 4.2.1.1 | Protocolo de comunicações do Módulo ESR | 45 |
| 4.2.2 | Câmera Estéreo | 46 |

| | | |
|---------------------------|--|----|
| 4.2.3 | LiDAR 3D | 47 |
| 4.2.4 | Plataformas experimentais CaRINA II e Caminhão autônomo | 48 |
| 4.3 | Software | 48 |
| 4.3.1 | OpenCV | 48 |
| 4.3.2 | PCL | 49 |
| 4.3.3 | ROS | 49 |
| 4.4 | Calibração de sensores | 49 |
| 4.4.1 | Calibração da câmera estéreo | 50 |
| 4.4.2 | Calibração do Radar em relação à câmera estéreo | 52 |
| 4.4.3 | Calibração do LiDAR 3D em relação à câmera estéreo | 54 |
| 4.5 | Detecção de Obstáculos | 56 |
| 4.5.1 | Preprocessamento da nuvem de pontos | 57 |
| 4.5.1.1 | Filtros | 57 |
| 4.5.1.2 | Remoção de chão | 58 |
| 4.5.1.3 | Segmentação dos obstáculos | 58 |
| 4.5.2 | Detecção de obstáculos usando “Stixel World” | 60 |
| 4.6 | Metodologia de Avaliação | 61 |
| 4.7 | Considerações Finais | 61 |
| 5 | EXPERIMENTOS E RESULTADOS | 63 |
| 5.1 | Experimentos exploratórios realizados com Radar ESR e câmera | 63 |
| 5.2 | Calibração de sensores | 65 |
| 5.2.1 | Radar-câmera | 66 |
| 5.2.2 | LiDAR-câmera | 66 |
| 5.3 | Validação da calibração | 68 |
| 5.3.1 | Radar-câmera | 69 |
| 5.3.2 | LiDAR-câmera | 71 |
| 5.4 | Detecção de Obstáculos | 71 |
| 5.4.1 | Fusão de sensores | 71 |
| 5.5 | Considerações Finais | 74 |
| 6 | CONCLUSÃO | 77 |
| 6.1 | Principais Contribuições | 78 |
| 6.2 | Trabalhos Futuros | 78 |
| 6.3 | Publicações geradas | 79 |
| REFERÊNCIAS | | 81 |

CAPÍTULO
1

INTRODUÇÃO

O crescente aumento do número de automóveis nas vias tem como consequência um aumento significativo nos acidentes de trânsito entre veículos, mas também entre veículos e bicicletas ou motocicletas, bem como tem levado a um grande número de acidentes com pedestres. Todos os anos mais de 270.000 pedestres morrem em acidentes no mundo¹. Esse número equivale a 22% das mortes causadas pelo trânsito. Outros milhares de pessoas sofrem lesões causadas por acidentes de trânsito enquanto caminham, algumas das quais tornam-se permanentemente incapacitadas. Esses acidentes causam muito sofrimento e também dificuldades econômicas para as famílias e entes queridos (WHO, 2013).

Somente no Estado de São Paulo, no ano de 2011, das 5.394 mortes por acidentes de trânsito registradas, 2.114 foram de pedestres, 1.721 de motociclistas, 1.273 de passageiros em veículos, e 286 de ciclistas. No ano de 2010, o número de pedestres atropelados foi 9% menor, com registro de 1.968 mortes. Os automóveis são os principais causadores de acidentes com vítimas fatais².

Motivados por estes dados estatísticos, muitos pesquisadores na área de robótica móvel estão interessados na automatização das tarefas do motorista, entre elas a detecção de pedestres, carros e outros obstáculos, para que a máquina possa: (i) tomar decisões automaticamente e evitar a colisão com o referido objeto ou evitar o atropelamento de um pedestre; (ii) gerar avisos ao motorista (Sistema de apoio ao motorista) de modo que este possa reagir a tempo de evitar um acidente. Em certas situações o motorista pode estar distraído, com sua atenção focada em uma determinada reação e/ou atividade, ou mesmo, podemos ter um elemento localizado em um “ponto cego” do veículo, o que aumenta significativamente o risco de acidentes. O uso de sistemas de apoio ao motorista passivos (sinalizar) ou ativos (atuar diretamente no controle do

¹ WHO – World Health Organization (ONU) Media Centre. Referencia: (Publicado em: 2 MAY 2013) <http://www.who.int/mediacentre/news/notes/2013/make_walking_safe_20130502/en>

² Governo do Estado de São Paulo - Secretaria da Saúde, Janeiro de 2014 (Pedestres lideram ranking de óbitos por acidentes de trânsito em SP). Referência: <<http://www.saude.sp.gov.br/ses/noticias/2013/maio/pedestres-lideram-ranking-de-obitos-por-acidentes-de-transito-em-sp>>

veículo) do tipo Advanced Driver Assistance Systems (ADAS) podem contribuir para reduzir de modo significativo o número e a gravidade dos acidentes envolvendo veículos e pedestres.

O Radar Automotivo ou Electronically Scanning Radar (ESR)³ oferece vantagens únicas em relação a outros sensores, pois pode detectar remotamente a presença de obstáculos, e em quase todas as condições de luminosidade e meteorológicas. O Radar Automotivo usualmente não possui partes móveis ou dependentes de uma mecânica fina, suportando vibrações e sendo bastante robusto. Ao contrário dos sensores acústicos, visuais e infravermelhos é menos suscetível a interferências externas. Muitos sistemas utilizam técnicas onde se fundem diferentes tipos de sensores, usados em conjunto de forma a complementar suas funções. A fusão de sensores pode, por exemplo, integrar Radares combinados com sensores infravermelhos (LINZMEIER et al., 2005), ou câmeras de vídeos (BERTOZZI et al., 2008) (BENEZETH et al., 2010) (GERONIMO et al., 2010).

Atualmente constata-se que muitas soluções propostas para a implementação de veículos autônomos apresentadas por grandes empresas, como Google/Waymo, Tesla e Uber, consideram a fusão de sensores. No que diz respeito a fusão de sensores de diferentes tipos, em abordagens denominadas híbridas (juntando diferentes tipos de sensores), a Tesla anunciou muito recentemente que vai buscar a integração de câmeras e radares em seus veículos⁴. Este upgrade proposto para o piloto automático dos veículos autônomos da Tesla é ligado a um grave acidente ocorrido com um de seus veículos que resultou na morte do passageiro, e que foi atribuída, de certo modo, a uma limitação/falha no sistema de visão por câmera. A integração de câmera e radar pode contribuir para melhorar o desempenho do sistema como um todo.

A adoção de abordagens híbridas (em termos de sensores) junto aos sistemas de percepção de obstáculos, tem por objetivo explorar as diferentes qualidades de cada tipo de sensor. Um bom sistema híbrido é aquele que sabe explorar os pontos positivos de cada módulo, tirando proveito de suas características individuais, ao mesmo tempo em que compensa as limitações de um com as vantagens apresentadas pelo outro. A fusão de sensores diferentes, em uma abordagem híbrida, busca compensar as limitações de um sensor através das vantagens oferecidas pelo outro sensor, que deste modo se complementam.

Em testes realizados, durante os estudos iniciais deste trabalho de mestrado, foi determinado que a informação da detecção de objetos utilizando o Radar automotivo está representada na forma de um conjunto de pontos e “blobs”, onde não há nenhuma informação em relação ao formato exato do obstáculo (medidas mais precisas de largura, altura ou comprimento dos objetos). Os “blobs” indicam a presença de um obstáculo com sua posição definida por um único ponto no espaço 3D (ponto que representa o obstáculo como um todo). Por exemplo, um outro veículo na pista seria detectado e representado como sendo um ponto único. Por outro

³ Delphi Electronically Scanning Radar: <<http://delphi.com/manufacturers/auto/safety/active/electronically-scanning-radar/>>

⁴ Tesla - Radar and Camera. Referência: (Acesso em: Jan 2017) <<https://www.tesla.com/blog/upgrading-autopilot-seeing-world-radar>>

lado, a detecção de obstáculos pelo Radar é bastante robusta, pois volumes sólidos são quase invariavelmente detectados pelo Radar, mesmo na presença de uma fraca iluminação, com poeira ou chuva.

Por causa da obtenção limitada das dimensões exatas dos obstáculos impostas pelo Radar, propomos neste trabalho uma fusão de Radar-Câmera Estéreo e Radar-LiDAR (Sensor Laser), a fim de identificar as dimensões e os contornos dos obstáculos detectados pelo Radar. Esta fusão permite um melhor funcionamento das detecções, além de possibilitar a utilização de Algoritmos de Processamento de nuvens de pontos para a detecção (segmentação) de objetos, auxiliados pelas informações fornecidas pelo Radar.

O sistema híbrido composto por Radar e Câmera Estéreo permite integrar a robustez da detecção de obstáculos fornecida pelo Radar (mesmo em situações adversas de visibilidade), com a precisão da câmera estéreo na definição do contorno e dimensões do objeto. Por outro lado, a câmera pode ser prejudicada em situações de baixa visibilidade, porém sendo esta limitação compensada pela detecção de obstáculos pelo Radar, apesar deste não fornecer uma detecção tão detalhada quanto a fornecida pela câmera.

Como resultado deste trabalho, buscou-se obter um sistema capaz de detectar obstáculos, sua posição e tamanho, usando sensores do tipo “Radar automotivo”, Light Detection and Ranging (LiDAR) e câmera estéreo. Com isto será possível gerar alertas devido a detecção de elementos que possam apresentar um perigo na condução do veículo, e possibilitar também a tomada de decisões para a atuação no veículo quando conduzido de modo autônomo.

1.1 Objetivos

1.1.1 Objetivo Geral

O principal objetivo da abordagem proposta é, a partir de um ponto situado no sistema de coordenadas do Radar (presença de um obstáculo), transformá-lo para o sistema de coordenadas da câmera estéreo e/ou do LiDAR (Sensor Laser) e em seguida segmentar e identificar o contorno do objeto na nuvem de pontos associada a esse ponto. O Radar fornece uma detecção robusta de obstáculos e a fusão com os dados da câmera ou LiDAR, permite uma detecção detalhada do obstáculo. A nuvem de pontos gerada pelo LiDAR foi utilizada também para validação da detecção dos obstáculos, que foram manualmente segmentados para assim conseguir uma melhor análise e avaliação dos resultados obtidos.

1.1.2 Objetivos específicos

Os objetivos específicos deste projeto de mestrado que permitiram atingir o objetivo geral descrito acima são:

- Realizar um estudo sobre os equipamentos, técnicas e métodos usados para a detecção de obstáculos em Radares automotivos, LiDAR 3D e visão computacional;
- Realizar a captura de dados usando um Radar automotivo, sendo necessário desenvolver as interfaces para a leitura dos dados e posterior acesso a estes em um sistema computacional;
- Estudar e implementar os algoritmos de geração do mapa de disparidade e mapas de profundidade, obtidos a partir das imagens estéreo;
- Realizar a calibração entre câmera estéreo-Radar e câmera estéreo - LiDAR 3D para ter os dados dos sensores alinhados espacialmente e realizar uma fusão adequada;
- Processar nuvens de pontos para a detecção de obstáculos;
- Estudar e implementar métodos de fusão de sensores, neste caso entre câmera - Radar e câmera - LiDAR 3D;
- Realizar testes práticos de detecção de obstáculos.

1.1.3 Organização do texto

No [Capítulo 2](#) são apresentados conceitos teóricos e definições importantes para o desenvolvimento deste trabalho. No [Capítulo 3](#) proporciona-se uma visão geral de alguns trabalhos relacionados com os temas abordados. No [Capítulo 4](#) é apresentada a metodologia junto com as ferramentas usadas no desenvolvimento do projeto. No [Capítulo 5](#) apresenta os testes, validações e resultados obtidos. Por fim no [Capítulo 6](#) são feitas análise e conclusões dos resultados e apresentados os trabalhos futuros.



FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados fundamentos essenciais para o desenvolvimento deste trabalho de mestrado que envolve sensores como o Radar, LiDAR e visão computacional. Começamos por definir alguns conceitos importantes sobre a teoria por trás da operação do Radar e fundamentos de visão computacional e calibração. Finalmente abordamos a fusão de sensores.

2.1 Radar

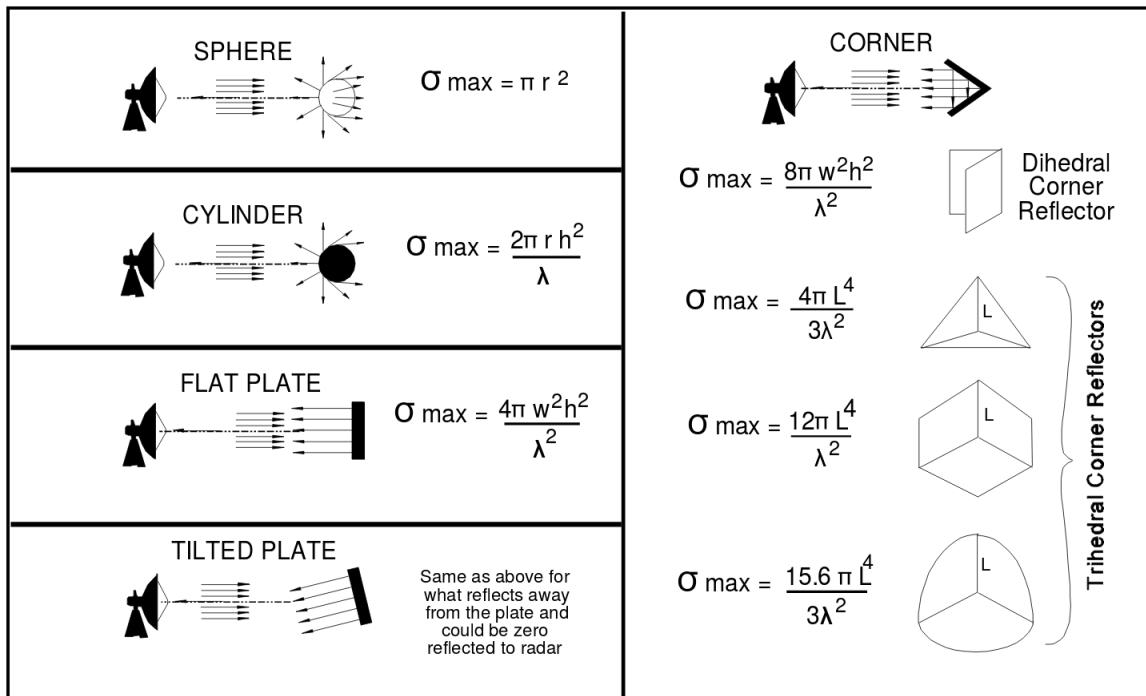
O Radar é um sistema eletrônico que envia ondas eletromagnéticas de rádio frequência (RF) a uma região de interesse e as recebe quando são refletidas desde objetos nessa região. Embora, os detalhes de um determinado sistema de Radar podem variar; estes devem incluir subsistemas como transmissor, antena, receptor e processador de sinais (RICHARDS; HOLM; SCHEER, 2010). O subsistema que gera as ondas eletromagnéticas é o transmissor. A antena interpreta essas ondas eletromagnéticas com entradas vindas do transmissor e as introduz em um meio de propagação (normalmente a atmosfera). O transmissor está conectado à antena através de um dispositivo de transmissão/recepção. O sinal transmitido se propaga através do meio ambiente em direção ao objetivo. As ondas eletromagnéticas induzem correntes ao obstáculo, essas correntes fazem com que o objeto também irradie ondas eletromagnéticas ao ambiente. Além do objetivo, outros elementos localizados sobre o chão e a atmosfera irradiarão o sinal. Estes sinais não são desejados e são conhecidos como “clutter”. Outros sinais não desejados são os originados fora do Radar, mas na mesma frequência do Radar conhecidos como “jamming”. A antena receptora do radar recebe as ondas eletromagnéticas que são refletidas desde um objeto (MELVIN; SCHEER, 2013).

2.1.1 Radar Cross Section (RCS)

O Radar Cross Section (RCS) é uma medida da capacidade de um alvo de refletir os sinais de Radar na direção do receptor, o RCS é a relação entre a densidade de potência que retorna na direção do Radar (vindo do alvo) e a densidade de potência (vindo do Radar) que é interceptada pelo alvo. O RCS é uma área representada por σ e as suas unidades são em $[m^2]$. O RCS de um alvo pode ser visto como uma comparação entre a intensidade do sinal refletido a partir de um alvo com o sinal refletido por uma esfera perfeitamente lisa de área transversal de 1 m^2 (NAWC, 1997).

A energia refletida a partir de um alvo na direção do receptor do Radar pode ser calculada através da resolução das equações de Maxwell aplicando as condições de fronteira adequadas ou usando modelagem por computador. O RCS também pode ser medido com base na equação do Radar utilizando modelos de tamanho real ou em escala. O RCS depende das características e dimensões do objeto em comparação com o comprimento de onda do Radar (SKOLNIK, 2001).

Figura 1 – Fórmulas para o RCS



Fonte: NAWC (1997).

Na Figura 1 são apresentadas as fórmulas para o cálculo do RCS de uma esfera, um cilindro e para vários refletores do tipo “corner” (“Dihedral” e “Trihedral”). Na Tabela 1 são apresentados alguns exemplos de objetos cotidianos e seus RCSs correspondentes.

Tabela 1 – RCS para objetos em frequências de micro-onda

| Alvo | RCS [m ²] |
|------------------|-----------------------|
| Pássaro | 0,01 |
| Pessoa | 1 |
| Automóvel | 100 |
| Caminhão | 200 |
| Corner reflector | 23562 |

Fonte: Adaptada de [Skolnik \(2001\)](#).

Nota: O RCS do “Corner reflector” foi calculado usando um refletor triangular com L = 1.5 m e 10 GHz.

2.2 Visão estéreo

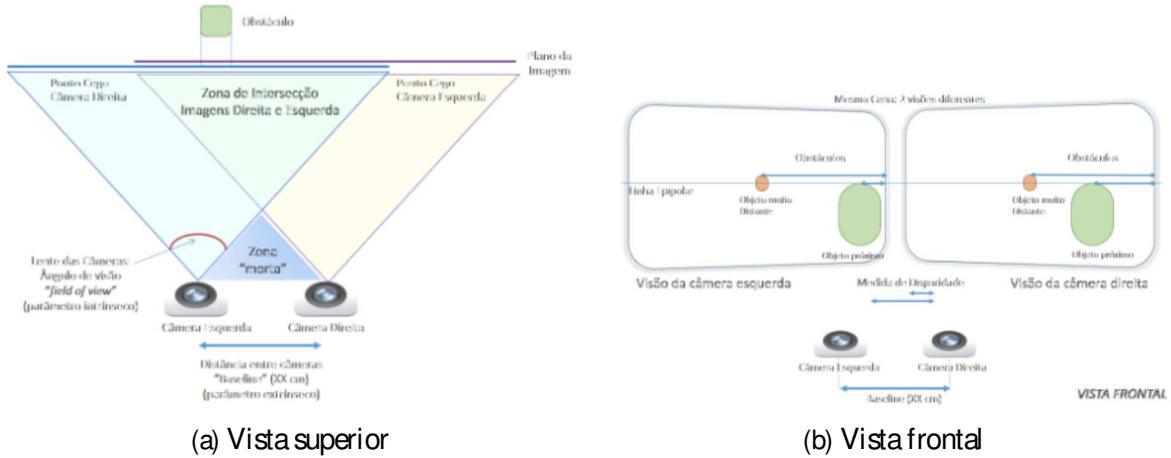
Um sistema de visão estéreo opera de uma maneira análoga ao modo como funciona o sistema de visão humana. No caso da visão estéreo são utilizadas duas câmeras que capturam imagens semelhantes da mesma cena, uma das câmeras tem um deslocamento em relação à outra. Sabendo a posição relativa de uma câmera em relação à outra e os parâmetros intrínsecos de cada câmera, é possível calcular a profundidade de cena. Através da identificação do deslocamento (disparidade) de um mesmo ponto da imagem nas duas cenas capturadas, e considerando-se o efeito da projeção perspectiva das cenas, é possível estimar a disparidade entre os elementos da cena, obtendo-se assim uma estimativa de profundidade. Elementos mais próximos da câmera possuem uma maior disparidade, e elementos mais distantes, devido a perspectiva da cena, possuem uma menor deslocamento em ambas imagens.

A visão estéreo possui algumas limitações que são as mesmas das câmeras comuns, tais como a necessidade de uma iluminação boa, além disso, na visão estéreo podemos ter as câmeras muito perto dos objetos, e com isto podem ser capturadas apenas por uma única câmera. Neste caso as imagens serão completamente diferentes e sem uma correspondência nas duas câmeras. Uma outra desvantagem é que, se os objetos estão distantes das câmeras, as imagens serão iguais. Em ambos casos o cálculo da disparidade é nula. Na [Figura 2](#) podemos ver um exemplo da configuração de duas câmeras capturando o mesmo objeto e sua visualização da posição relativa com relação as duas câmeras ([Figura 2a](#)) e a posição do objeto nas duas imagens ([Figura 2b](#)).

2.2.1 Modelo “pin hole” da câmera e calibração

É apresentada a seguir uma visão geral de um modelo geométrico (“pin-hole”) para uma câmera e sua respectiva calibração. Existe uma grande gama de câmeras e configurações das mesmas, assim como diferentes tipos e formas das lentes que são o principal componente óptico das câmeras. As câmeras não são perfeitas e dependendo da qualidade apresentam variações no ângulo de abertura e distorções na curvatura da lente. Quando se deseja executar a detecção de objetos ou a compreensão da cena em visão computacional, essas variações na câmera podem interferir com o processo de detecção. Por exemplo, a forma de um objeto pode ter um

Figura 2 – Captura de duas imagens simultâneas com câmeras distantes entre si separadas por uma determinada distância/baseline

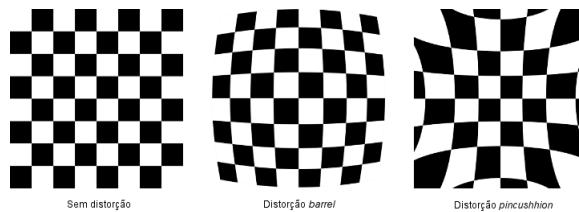


Fonte: Elaborada pelo autor.

aspecto diferente ao que tem na realidade devido à distorção introduzida pela lente da câmera. Conhecendo a geometria que causa tais distorções é possível realizar um processo de correção da distorção e eliminar a influência da geometria da lente a fim de melhorar a detecção dos objetos e a compreensão da cena. Este processo também é conhecido como retificação.

A maioria das lentes da câmera introduz dois tipos de distorções: a radial e a tangencial. As distorções radiais são simétricas em torno do centro da imagem e mostram um zoom maior ou menor no centro da imagem. Esses efeitos são conhecidos como distorção de barril (barrel) e almofada (pincushion), como apresentado na Figura 3. Nas distorções tangenciais, a imagem pode mostrar um nível de zoom maior na parte superior ou na parte inferior da imagem.

Figura 3 – Distorções da lente



Fonte: Adaptada de [Itseez \(2016a\)](#).

2.2.1.1 Calibração intrínseca e extrínseca

O modelo da câmera define como a imagem é formada a partir de raios de luz incidentes sobre o sensor da câmera. O modelo “pin-hole” disponível em OpenCV ([ITSEEZ, 2016a](#)) pode ser usado para a maioria das câmeras. O termo calibração se refere ao processo de encontrar os parâmetros para o modelo da câmera. Entre eles estão os parâmetros intrínsecos e extrínsecos. Os parâmetros intrínsecos são a distância focal, o centro da imagem e a distorção. Os parâmetros

extrínsecos definem a posição (translação e orientação (rotação) da câmera em relação a um sistema de coordenadas de referência.

2.2.1.2 Modelo da câmera implementado no OpenCV

O modelo utilizado nas bibliotecas da OpenCV ([ITSEEZ, 2016c](#)) é baseada no trabalho apresentado pela Microsoft Research em [Zhang \(2000\)](#) e foi expandido várias vezes para modelar diferentes tipos de distorções. O modelo é definido através do “pin-hole” e uma função de distorção que mapeia os pontos da imagem ideal projetados pelo “pin-hole” para a imagem distorcida utilizando os parâmetros de distorção. A [Equação 2.1](#) é uma das formas de representar o modelo “pin-hole”.

$$S = \begin{bmatrix} F_x & F_y \\ F_z & M \end{bmatrix} = \begin{bmatrix} f_x & 0 \\ f_y & 0 \end{bmatrix} + \begin{bmatrix} O_x & r_{11} & r_{12} & r_{13} & t_1 \\ O_y & r_{21} & r_{22} & r_{23} & t_2 \\ 1 & r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} + \begin{bmatrix} z \\ K \end{bmatrix} \begin{bmatrix} z \\ [R|t] \end{bmatrix} \quad (2.1)$$

A matriz da câmera (de parâmetros intrínsecos) é a matriz $K \in \mathbb{R}^{3 \times 3}$. Esta matriz é usada para transformar um ponto $(X_w, Y_w, Z_w) \in \mathbb{R}^3$ com referência às coordenadas tridimensionais da câmera para o sistema de coordenadas da imagem final $(u, v) \in \mathbb{R}^2$ onde f_x, f_y são as distâncias focais em unidades de pixel e (c_x, c_y) é o ponto principal, geralmente localizado no centro da imagem, também em pixels. O fator de escala S é igual a Z ([Equação 2.1](#)). Existem implementações de métodos de calibração automáticos que permitem estimar os parâmetros da matriz K , como por exemplo o método da OpenCV que captura um conjunto de imagens de um padrão xadrez usando duas câmeras e a partir disso encontra os parâmetros intrínsecos das câmeras (mais detalhes em [Itseez \(2016a\)](#)).

2.2.1.3 Estimativa da pose da câmera

Se o sistema de coordenadas do mundo (X_w, Y_w, Z_w) é diferente do sistema de coordenadas da câmera (no caso de tomar o sistema de coordenadas de um outro sensor como sistema de coordenadas do mundo) deve-se calcular a pose da câmera que é a matriz de transformação que permite fazer translações e rotações em uma única matriz ($[R|t]$). Os parâmetros r_{ij} são rotações nos três eixos e os parâmetros t_i são translações que representam um offset relativo ao eixo da câmera.

A estimativa da pose de uma câmera usando a correspondência de n pontos 3D para n pontos 2D é um problema de visão computacional amplamente estudado e já resolvido. Para resolver o problema precisa-se estimar os seis graus de liberdade da câmera e os parâmetros (intrínsecos) de calibração da câmera. Uma primeira abordagem para resolver o problema é usando um mínimo de 6 pares de correspondências entre pontos 3D do mundo e 2D da imagem.

que pode ser feita usando o conhecido algoritmo Direct Linear Transform (DLT) ([HARTLEY; ZISSEMAN, 2004](#)). O algoritmo DLT estima uma matriz de projeção P resolvendo um sistema de equações lineares. A partir de pontos (X_i, Y_i, Z_i) 3D e a matriz de projeção P encontrar seu correspondente ponto (u_i, v_i) no plano 2D da imagem. A [Equação 2.2](#) e [Equação 2.3](#) são duas equações linearmente independentes que representam a solução do problema usando DLT.

$$\frac{P_{11}X_i + P_{12}Y_i + P_{13}Z_i + P_{14}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = u_i, \quad (2.2)$$

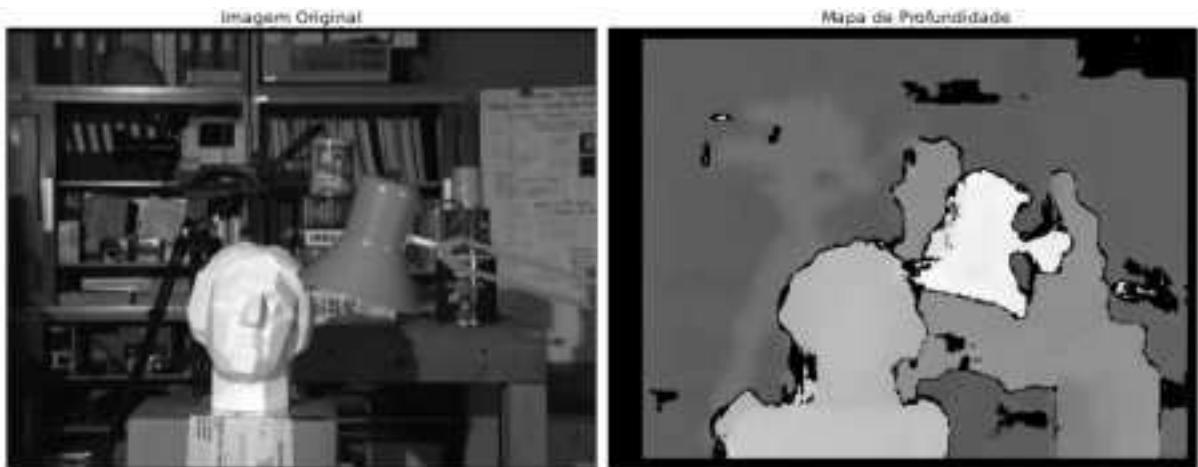
$$\frac{P_{21}X_i + P_{22}Y_i + P_{23}Z_i + P_{24}}{P_{31}X_i + P_{32}Y_i + P_{33}Z_i + P_{34}} = v_i, \quad (2.3)$$

Este sistema de equações pode ser escrito de forma matricial como $AP = 0$, onde P é uma matriz de coeficientes P_{ij} e (X_i, Y_i, Z_i) são os pontos 3D que se pretende projetar no plano da imagem. O problema pode ser resolvido usando o Single Value Decomposition (SVD) de A para encontrar P. Da matriz P podemos extraír a pose da câmera (matriz $[R|t]$) usando $[R|t] \approx K^{-1}P$ como uma aproximação. Assim o algoritmo DLT requer o cálculo dos parâmetros de calibração intrínsecos da câmera (K), por causa disso foram propostas numerosas simplificações ao problema para melhorar a precisão e eficiência do cálculo do algoritmo. Uma variante do DLT apresentado é o algoritmo Perspective-n-Point (PnP), que pressupõe que os parâmetros intrínsecos da câmera são conhecidos. No PnP apenas três correspondências de pontos são necessárias para recuperar a pose da câmera ([GAO et al., 2003](#)). Existem outras abordagens de soluções iterativas para o problema usando $n > 3$ pontos ([DEMENTHON; DAVIS, 1995](#)) ([HORAUD; DORNAIKA; LAMIROY, 1997](#)) ([LU; HAGER; MJOLSNESS, 2000](#)). No entanto, as soluções não-iterativas são muito usadas por sua complexidade computacional e precisão de $O(n^3)$ ([ANSAR; DANIILIDIS, 2003](#)), $O(n^2)$ ([FIORE, 2001](#)), e um dos mais eficientes é o algoritmo EPnP com complexidade $O(n)$ ([LEPETIT; MORENO-NOGUER; FUA, 2009](#)).

2.2.2 Mapa de disparidade

O mapa de disparidade tem sido amplamente utilizado para a reconstrução 3D, e pode ser calculado usando diferentes abordagens e algoritmos (matching), destacando-se algoritmos globais e locais com suas respectivas vantagens e desvantagens em termos de precisão e velocidade de cálculo. Os mapas de disparidade são calculados usando um par de imagens estéreo, geralmente em tons de cinza, e com elas podemos obter um mapa de profundidade da cena tendo em conta os parâmetros intrínsecos resultantes da calibração da câmera estéreo. Na [Figura 4](#) é mostrado um exemplo de uma imagem gerada na câmera esquerda do par estéreo e o seu correspondente mapa de profundidade. Neste mapa de profundidade os pontos mais claros representam elementos mais próximos da câmera, e os mais escuros representam os elementos mais distantes (imagem em tons de cinza).

Figura 4 – Mapa de profundidade

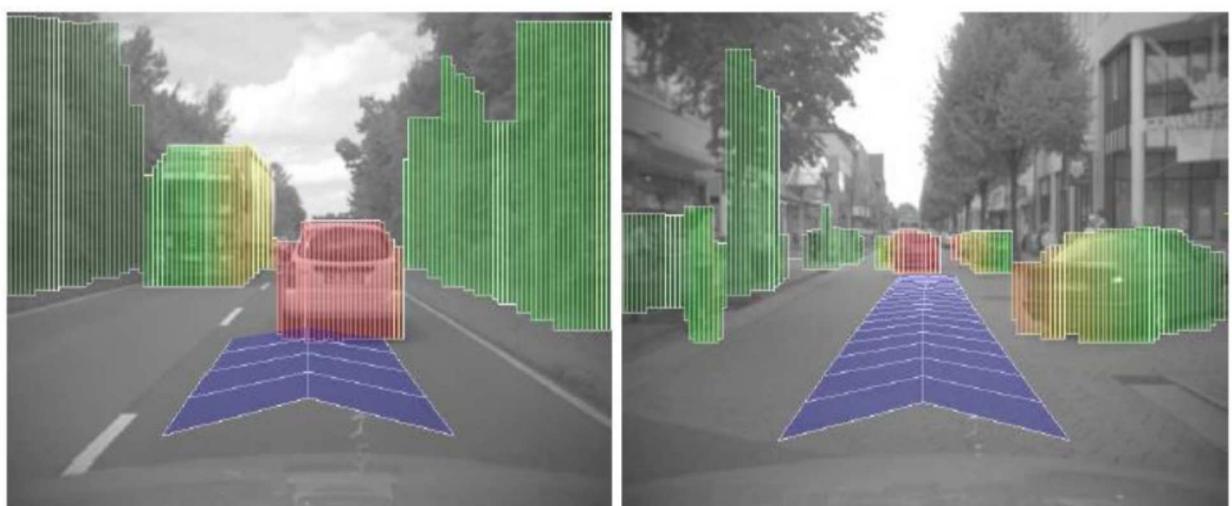


Fonte: Adaptada de [Itseez \(2016b\)](#).

2.2.2.1 Métodos para calcular a disparidade.

Uma das maneiras de classificar os métodos de cálculo da disparidade é considerar como as combinações entre pixels são selecionadas. Estes métodos podem ser globais ou locais ([CYGANEK; SIEBERT, 2011](#)). Outros autores como ([HIRSCHMULLER, 2005](#)) e ([HALLER et al., 2010](#)) consideram também métodos semi-globais, que executam várias otimizações parciais (em uma dimensão) para aproximar a solução para uma otimização global. Um desses algoritmos é o algoritmo Semi Global Matching (SGM), disponível no OpenCV.

Figura 5 – Exemplo da execução do “Stixel”



Fonte: [Badino, Franke e Pfeiffer \(2009\)](#).

2.2.3 Stixel World

O conceito do “Stixel” foi introduzido por [Badino, Franke e Pfeiffer \(2009\)](#). Esta abordagem representa informação tridimensional da cena usando agrupamento de volumes que representam objetos. O “Stixel” assume planaridade do chão e a detecção dos objetos é calculada iniciando no plano do chão. Apesar de ter restrições como a suposição de que o terreno é plano tem mostrado vantagens na detecção de aplicações de obstáculos e reconhecimento de cena ([BENENSON et al., 2012](#)).

Esta abordagem baseada em Stixels permite realizar a detecção de obstáculos, e sendo assim, foi usada neste trabalho para fins de comparação da qualidade de detecção baseada neste método e nos demais métodos estudados e propostos. Os resultados dos experimentos de detecção de obstáculos são apresentados no [Capítulo 5](#), comparando o Stixel com abordagens individuais e de fusão de sensores.

2.3 LiDAR

Nesta seção, fazemos uma introdução ao sistema LiDAR e incluímos uma visão geral de como o sistema funciona. O LiDAR usa pulsos de feixes de laser. O tempo de retorno do feixe é medido para encontrar uma medição precisa da distância até o objeto onde o laser bate. Também é medida a intensidade do feixe com o qual podemos saber quan reflexiva é a superfície do objeto. Existem diferentes tipos de LiDAR, os que podem medir um único ponto (uma dimensão), os que podem medir vários pontos situados num plano (duas dimensões), e os que medem vários pontos no espaço 3D (tridimensional). De particular interesse neste trabalho é o LiDAR 3D que pode ler uma grande quantidade de pontos para cada escaneio.

Sensores, tais como os fabricados pela Velodyne tem uma precisão de 2 cm e resoluções entre 300.000 a 2.2 milhões de pontos por segundo para o modelo HDL-64E, com alcances entre 80 e 120 metros e 360 graus de Field of View (FOV). Embora a densidade de pontos 3D do ambiente gerados por uma câmera estéreo é muito mais densa do que a gerada por um LiDAR 3D ainda pode-se determinar a forma e tamanho dos objetos ao contrário de sensores como o Radar Automotivo onde seus dados são mais esparsos para cada objeto detectado. O sensor reporta um único ponto e assim não é possível saber o tamanho e a forma do objeto. O LiDAR 3D é incapaz de detectar outras características do ambiente além da posição e a intensidade da reflexão do objeto atingido pelo feixe de laser. Com uma câmera estéreo podem ser obtidas outras informações, tais como cor e textura, mas no caso da estimativa da profundidade não é tão preciso como o LiDAR 3D. As câmeras são também sensíveis às condições de iluminação. Os sensores LIDAR também são sensíveis a essas condições mas em menor grau. Apesar do LiDAR 3D ser muito preciso, uma desvantagem na utilização destes sensores é o seu preço elevado em comparação com outros sensores, tais como câmeras e Radar, embora em anos recentes há uma tendência para os preços caírem e a fabricação de sensores LiDAR 3D torna-se mais barata.

2.4 Fusão de sensores

A ideia de fusão de sensores não é nova e é amplamente encontrada em seres vivos, sendo usada pelos humanos no dia-a-dia. Para a maioria das tarefas cotidianas que realizamos usamos diferentes sentidos que são análogos a sensores, por exemplo: usamos a percepção visual e o tato para levar um objeto de outro lugar a outro. A ideia geral da fusão de sensores é compensar as deficiências de um sensor e tirar proveito das vantagens de outro para melhorar a tarefa de detecção. As técnicas de fusão fundem dados a partir de vários sensores e identificam as informações relacionadas, para alcançar uma melhor inferência da que poderia ser alcançada utilizando um sensor separado ([II; HALL; LLINAS, 2008](#)).

Em [II, Hall e Llinas \(2008\)](#) são definidos três níveis principais para fusão de sensores que são apresentados a seguir:

2.4.1 Fusão em nível de dados (Data Level Fusion)

Também conhecido como fusão de sensores de baixo nível “Low level fusion”, é um método que utiliza dados “raw” de vários sensores e os funde para gerar uma hipótese ou informações mais ricas sobre a cena. Utilizando, por exemplo, um sensor de profundidade como um sensor “Kinect” e uma câmera monocular RGB devidamente calibrados, os pontos da distância medidos pelo “Kinect” são mapeados na imagem 2D obtendo assim um mapa de profundidade do tipo RGB-D.

2.4.2 Fusão em nível de características (Feature Level Fusion)

Também conhecido como fusão de nível médio “mid level fusion”, neste método de fusão de dados, primeiramente são extraídos vetores de características dos dados de cada sensor individualmente. Esses vetores são a base da fusão de dados. Por exemplo, podemos ter um vetor de dados resultado da segmentação de uma nuvem de pontos gerada por um sensor laser e projetar essas segmentações numa imagem para encontrar regiões de interesse (ROI) onde se podem classificar objetos baseados nas características da região da imagem.

2.4.3 Fusão em nível de Declaração (Declaration Level Fusion)

Também conhecido como fusão de alto nível, é o método de fusão de sensores onde as hipóteses geradas por diferentes sensores são fundidas. Cada sensor estima o seu estado individualmente. O estado final do objeto detectado é a fusão das hipóteses de cada sensor. Muitas das abordagens utilizadas para fusão de alto nível são baseadas em métodos probabilísticos, um dos mais conhecidos é o filtro Kalman ([KALMAN; BUCY, 1961](#)).

2.5 Considerações Finais

Neste capítulo foram apresentadas as principais ferramentas e métodos que serão utilizados no desenvolvimento do projeto. Embora o foco deste projeto não seja o comportamento e a física do Radar, é importante conhecer conceitos básicos de sua operação, principalmente para ter uma boa escolha dos alvos a serem utilizados como padrões de calibração. Da mesma forma, é importante considerar alguns fundamentos de visão computacional principalmente modelos de câmeras, calibração e visão estéreo. Todos estes conceitos teóricos são importantes para a análise e a escolha do nível de fusão de dados.

CAPÍTULO
3

TRABALHOS RELACIONADOS

Um dos principais eventos que fomentaram a investigação de veículos autônomos foram as competições promovidas pelo Defense Advanced Research Projects Agency (DARPA), que pertence ao Departamento de Defesa dos Estados Unidos. A primeira edição dessa competição de veículos autônomos foi realizada em 2004 no deserto de Mojave, onde não houve ganhadores. Na segunda edição, realizada em 2005 o vencedor foi o veículo “Stanley” da Universidade de Stanford; e na terceira edição da competição, denominada de “The Urban Challenge” (2007), onde o ganhador foi a equipe “Tartan Racing” composta pelas Universidades Carnegie Mellon, Pittsburgh e Pennsylvania. Para esta última edição os veículos autônomos de cada equipe deviam percorrer ambientes urbanos, portanto, era necessário que pudessem evitar obstáculos estáticos e em movimento, respeitando inclusive as regras de trânsito locais da Califórnia.

Com o objetivo de competir e vencer no “The Urban Challenge”, a equipe “Tartan Racing” desenvolveu um sistema de percepção baseado na fusão de sensores LiDAR e Radar ([URMSON et al., 2007](#)). Este sistema foi dividido em três camadas: camada de sensores, camada de fusão de sensores e análise da situação. A camada de sensores do sistema se encarregava de coletar os dados de todos os sensores e estes eram guardados em uma lista de maneira independente, proporcionando uma classificação inicial dos objetos, a qual dependia das características de cada sensor. Por exemplo, os Radares não são capazes de definir precisamente a forma de um objeto enquanto que o LiDAR pode detectar se o objeto tem a forma mais bem definida de um automóvel. Na camada de fusão de sensores, o algoritmo de fusão é responsável pelo processamento dos dados coletados na primeira camada e cada nova medição é associada com um objeto rastreado. Na camada de análise da situação é prevista qual é a “intenção” do objeto rastreado, mediante a integração das estimativas de posição, deslocamento, juntamente com o conhecimento sobre o modelo da via.

O sistema de percepção do veículo autônomo é capaz de detectar objetos estáticos e em movimento. O algoritmo de detecção de objetos estáticos faz uma comparação de dois pontos gerados por sensores laser. Essa comparação calcula um custo baseado em duas coisas: a primeira

é a diferença de elevação entre os pontos e a segunda é o ângulo que forma o vetor que conecta os dois pontos em relação ao solo. Os maiores valores de elevação e ângulo são as características que determinam se o obstáculo que se apresentou é considerado ou não “perigoso”. Para a detecção de objetos em movimento é feito o uso de um modelo dinâmico simples, no qual se leva em consideração a posição do centro do objeto, a velocidade e a aceleração linear do objeto. Usa-se um Filtro de Kalman Estendido para estimar e atualizar o estado de cada objeto seguido (URMSON et al., 2007).

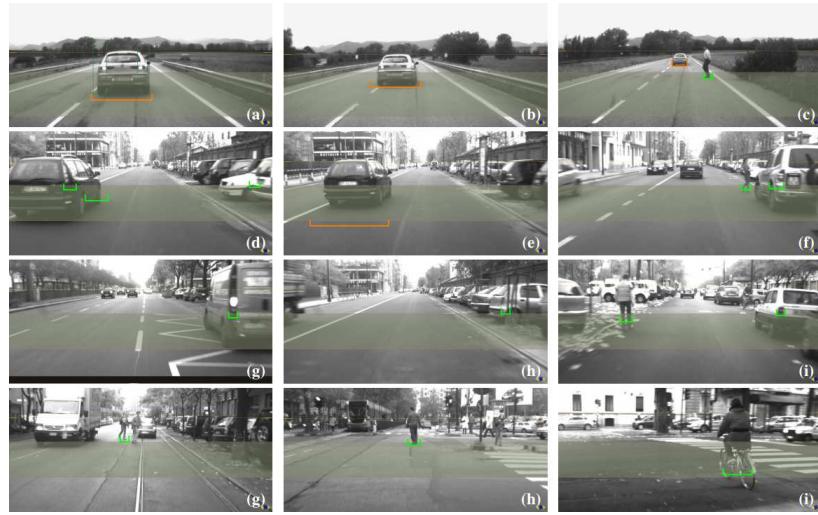
Muitas abordagens para a detecção de obstáculos têm sido propostas baseadas no uso de sensores de som, imagens, LiDAR e Radar. Outros realizam a tarefa de detecção de obstáculos usando dois ou mais sensores, tais como LiDAR, Radar ou câmeras e fundem características destes sensores para melhorar as detecções. Algumas abordagens usam algoritmos de aprendizado de máquina que são treinados para detectar objetos específicos, como pessoas, ou carros (XIANG et al., 2016) usando imagens, outros autores utilizam técnicas onde se fundem diferentes tipos de sensores, usados em conjunto de forma à complementar suas funções.

A fusão de sensores pode, por exemplo, integrar Radares combinados com sensores infravermelhos (LINZMEIER et al., 2005), câmeras de vídeos (BERTOZZI et al., 2008), (BENEZETH et al., 2010), (GERONIMO et al., 2010) ou LiDAR (FRITSCHE et al., 2016). Em Alencar et al. (2015) foi usado o MMW Radar para encontrar hipóteses de obstáculo no mundo 3D. Essas hipóteses encontradas pelo Radar são projetadas em uma imagem utilizando a matriz de projeção da câmera que foi encontrada pela calibração da câmera, dado um conjunto de imagens de padrão xadrez. A pose da câmera em relação ao Radar é estimada pela medição manual das distâncias entre a câmera e o Radar.

Em muitos dos trabalhos relacionados ao desenvolvimento de veículos inteligentes foi usado um enfoque de fusão de sensores para reconhecimento de pedestres, principalmente sendo usados sinais de Radar em conjunto com imagens de câmeras estereoscópicas (KATO; NINOMIYA; MASAKI, 2002) e monoculares (WANG et al., 2014). Em Langer e Jochem (1996) a detecção de pessoas é feita usando Radar e visão artificial para classificar os obstáculos segundo a periculosidade que representam para os veículos (BERTOZZI et al., 2008).

Pesquisadores da Hamburg University of Technology conseguiram fazer a detecção de pedestres usando unicamente sinais de Radar de 24 Ghz (HEUEL; ROHLING, 2013). Eles fazem a detecção de pedestres e veículos. Este processo é composto de três etapas: i) leitura de sinais de Radar, ii) extração de características, iii) classificação (HEUEL; ROHLING, 2012) (HEUEL; ROHLING, 2013). Além disso, para a extração usam modelos dinâmicos de pedestres e veículos (RITTER; ROHLING, 2007) (ROHLING; HEUEL; RITTER, 2010). Estes trabalhos apresentam um bom exemplo de aplicação do Radar, além de prover informações relevantes para o desenvolvimento da atual proposta de pesquisa. A Figura 7 mostra resultados obtidos pelo grupo de pesquisadores.

Figura 6 – Resultado do método de detecção em diferentes condições ambientais



Fonte: [Bertozzi et al. \(2008\)](#).

Figura 7 – Resultados do método de detecção de pedestres usado pelos investigadores da Hamburg University of Technology



Fonte: [Heuel e Rohling \(2013\)](#).

3.1 Laboratório de Robótica Móvel (LRM)

Dentro do Laboratório de Robótica Móvel (LRM)¹ do ICMC/USP, onde se realiza esse trabalho, estão sendo desenvolvidos diversos projetos no âmbito de robôs móveis autônomos, onde se destacam projetos com robôs de pequeno e médio porte usados em ambientes fechados (indoor) e projetos de pesquisa sobre veículos autônomos (outdoor). Os veículos autônomos em desenvolvimento são capazes de estarem imersos em ambientes urbanos e mesmo em ambientes semi-estruturados, como por exemplo, em aplicações agrícolas.

¹ Laboratório LRM – Site: <<http://www.lrm.icmc.usp.br/>>

No caso do projeto Carro Robótico Inteligente para Navegação Autônoma (CaRINA)² em desenvolvimento junto ao LRM-ICMC/USP, e que é mais diretamente relacionado com a proposta deste projeto, este visa o desenvolvimento de um veículo autônomo inteligente capaz de navegar em ambientes urbanos sem a necessidade de um condutor humano. O projeto CaRINA conta atualmente com duas plataformas de pesquisa experimental, que já se encontram automatizadas e capazes de realizar uma navegação autônoma (FERNANDES et al., 2014), sendo apresentadas na Figura 8. Dentre os objetivos do projeto CaRINA destacam-se: a diminuição do número de acidentes em ruas e rodovias, o aumento na mobilidade de idosos e portadores de necessidades especiais e o aumento da eficiência do trânsito em geral. Outro objetivo desse projeto é o desenvolvimento de um sistema de auxílio ao motorista, notificando o motorista de uma situação de risco durante a condução do veículo, bem como fazer o controle deste quando o perigo é iminente. Nos últimos resultados obtidos, o carro elétrico CaRINA I foi capaz de percorrer mais de 1,0 Km em modo autônomo no Campus da USP, e o veículo CaRINA II foi capaz de percorrer sem motorista mais de 20 Km em ambientes de vias urbanas³. Além dos veículos do projeto CaRINA, também foi desenvolvido um caminhão autônomo em parceria com a empresa Scania (projeto LRM-CROB/USP em parceria com a Scania Latin-America). O caminhão autônomo foi desenvolvido visando a adoção de percepção por Radar e visão estéreo como seus principais dispositivos de detecção de obstáculos, tendo sido apresentado em várias ocasiões para várias empresas e a mídia. O sistema de detecção de obstáculos do caminhão Scania é fortemente baseado nos trabalhos desenvolvidos junto a esta dissertação de mestrado.

Atualmente o CaRINA I e II contam apenas com sensores baseados em câmeras de vídeo (monocular e estéreo) e sensores a Laser (SICK e Velodyne), o caminhão autônomo foi equipado com Radar e câmera estéreo para detecção de obstáculos. Como resultado desta pesquisa foi desenvolvido um sistema para a fusão de Radar e visão computacional para usar nestas plataformas.

Um outro trabalho desenvolvido no LRM-ICMC/USP por Klasler (2014), propõe um sistema de navegação autônoma em ambientes não estruturados utilizando visão computacional. A percepção do sistema é baseada em uma câmera estéreo. O mapa de custos nesse sistema de navegação é obtido através de um modelo de ocupação probabilística fazendo uso da técnica OctoMap (HORNUNG et al., 2013). Nesse trabalho é descrito um modelo sensorial que usa informação espacial proveniente de nuvens de pontos obtidas a partir de visão estéreo para atualizar o OctoMap. O sistema foi validado em simulação e em ambiente real com cenários de terreno irregular e diversos obstáculos. A Figura 9 mostra resultados obtidos nesse trabalho.

O trabalho de Shinzato, Wolf e Stiller (2014), também desenvolvido junto ao LRM-ICMC/USP, propõe um sistema baseado na fusão de sensores capazes de detectar obstáculos em uma variedade de cenários, utilizando um número mínimo de parâmetros. A proposta baseia-

² Projeto CaRINA – Site: <<http://www.lrm.icmc.usp.br/carina/>>

³ Fonte: Laboratório de Robótica Móvel – LRM, ICMC – USP 2014 (Projeto CaRINA) <<http://www.lrm.icmc.usp.br/web/index.php?n=Port.Midia>>

Figura 8 – Plataformas autônomas projeto CaRINA



(a) CaRINA I (Veículo elétrico Club Car adaptado) (b) CaRINA II (Veículo comercial Fiat Palio Adventure adaptado)



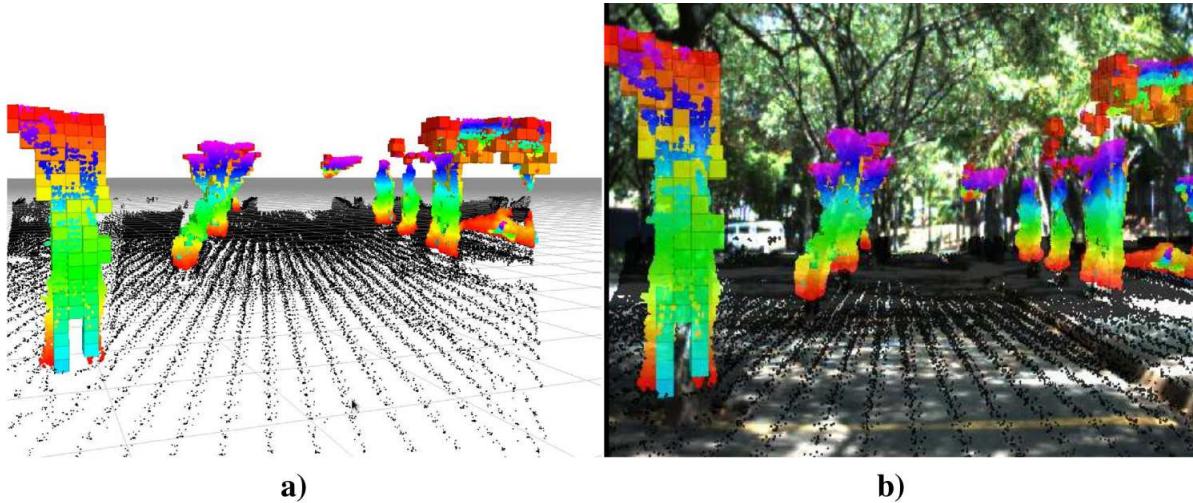
(c) Caminhão autônomo (Scania G360 de 9 ton.)

Fonte: Elaborada pelo autor.

se na relação espacial entre imagens em perspectiva fornecidas por uma única câmera e um LiDAR 3D. O sistema segue uma sequência de cinco passos: o primeiro passo é a fusão, onde cada ponto 3D é projetado em uma imagem 2D; o segundo passo gera um grafo que cria uma reação espacial local entre todos os pontos; o terceiro passo é a classificação dos pontos como obstáculos ou não obstáculos; a quarta etapa gera diversos histogramas polares que estimam as áreas livres; finalmente, a quinta etapa cria um mapa de confiança combinando todas as áreas livres determinadas pelos histogramas polares criados no ponto anterior. A [Figura 10](#) mostra os resultados obtidos com esse sistema.

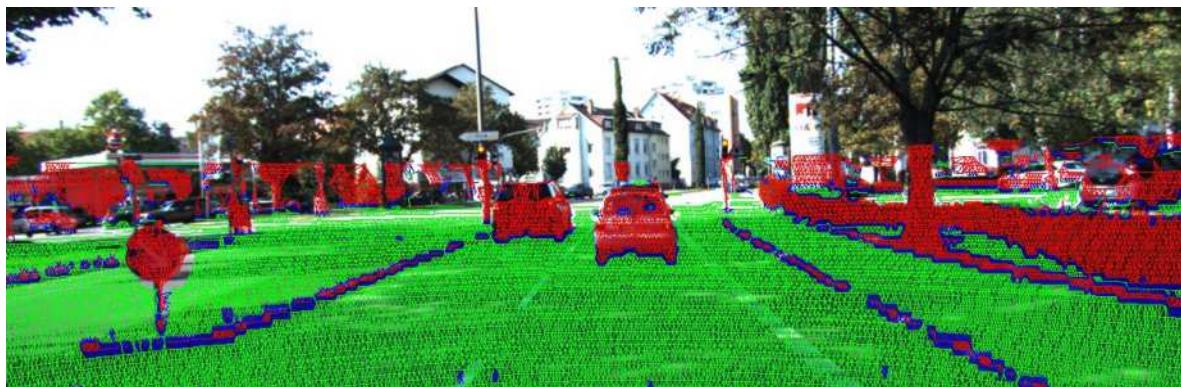
O trabalho de [Utino, Wolf e Osorio \(2014\)](#), desenvolvido junto ao LRM-ICMC/USP, propõe a fusão de informações provenientes do processamento dos dados de duas câmeras, uma estéreo e outra térmica, onde visa a navegação autônoma de veículos inteligentes em ambientes agrícolas. Nessa abordagem, três módulos de detecção de obstáculos foram desenvolvidos. O

Figura 9 – Detecção do plano do chão (em preto) e marcação dos obstáculos (em cores). Visualização (a) e projeção da visualização na cena real (b)



Fonte: [Klaser \(2014\)](#).

Figura 10 – Resultados do método de detecção de obstáculos no mundo imagem 2D.



Fonte: [Shinzato, Wolf e Stiller \(2014\)](#).

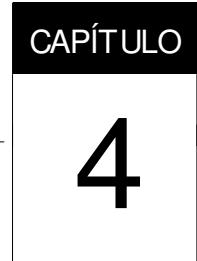
primeiro utiliza imagens da câmera monocular (imagens oriundas de uma das câmeras do par estéreo) para detectar novidades na cena considerando as cores dos elementos do ambiente (em sua maioria compostos por vegetação), por meio da comparação dos estados atual e anterior. Este módulo identifica objetos que possuem uma cor distinta do modelo de cor das imagens anteriores, como por exemplo, pessoas e equipamentos que “aparecem” (novidade) em uma nova imagem. O segundo módulo usa a câmera estéreo e a técnica Stixel para encontrar obstáculos e diferenciá-los do chão.

Finalmente, o terceiro módulo utiliza as imagens da câmera térmica para encontrar obstáculos que se distinguem do cenário pela sua temperatura. Uma vez que pessoas e equipamentos emitem calor, estes podem ser identificados nas imagens mesmo a noite e sem uma iluminação adequada da cena. As informações desses três módulos são fundidas usando a teoria Dempster-Shafer, onde as informações dos sensores se completam, uma vez que a câmera monocular

(imagens coloridas) tem uma boa resposta de dia e sob boa iluminação, a câmera termal tem uma boa resposta na noite e sob condições de fraca iluminação, e por fim, a câmera estéreo (stixel) permite uma análise 3D da cena e dos obstáculos, ao contrário das outras duas imagens que não carregam consigo informações de dimensionamento e profundidade dos obstáculos.

3.2 Considerações Finais

Os trabalhos apresentados nesta seção demonstram diferentes possibilidades de aplicações de detecção de obstáculos, usando sensores como Radares e sistemas de visão computacional. É importante destacar que diversos destes trabalhos, e principalmente os mais atuais, são baseados na fusão de sensores. Nesta dissertação de mestrado foram considerados estes trabalhos relacionados e as aplicações descritas como referência para desenvolver uma proposta de fusão de sensores. Na nossa proposta foi testada a detecção de obstáculos usando sensores LiDAR, Radar e câmera estéreo individualmente e fazendo fusão de LiDAR-Radar e Câmera estéreo-Radar. Muitos dos trabalhos da literatura implementam fusão de sensores mas essas abordagens são testadas sem usar ou detalhar um método de calibração mais específico para suas abordagens. Em outros casos, os “datasets” que já fornecem uma calibração de sensores pronta. Nesta dissertação de mestrado foram usados métodos para calibração de sensores e alinhamento espacial dos dados para melhorar detecção de obstáculos baseada em fusão de sensores. No capítulo seguinte será apresentada a metodologia que foi adotada para o desenvolvimento do sistema proposto e que está baseada em estudos prévios e nos trabalhos apresentados neste capítulo.



METODOLOGIA

Neste capítulo, mostramos a estrutura e a metodologia seguidas neste trabalho para realizar a fusão de sensores usando Radar, LiDAR 3D e câmera estéreo. Começamos por explicar a estrutura geral do sistema, em seguida apresentamos uma descrição do hardware utilizado (Radar, LiDAR e câmera 3D estéreo) e o trabalho realizado com cada um deles para a captura de dados. Também identificamos os problemas particulares de cada sensor. Descrevemos os frameworks que suportam o desenvolvimento das aplicações para calibração e detecção de obstáculos. Depois apresentamos uma proposta para a calibração de sensores, que se destina a encontrar a pose da câmera estéreo em relação aos sensores 3D (LiDAR e Radar). Finalmente, apresentamos o método de detecção de objetos baseado na fusão de sensores em nível de características implementado neste trabalho e a forma de avaliação.

4.1 Arquitetura para a fusão de sensores

A [Figura 11](#) mostra o modelo simulado da plataforma experimental CaRINA II e a configuração dos sistemas de coordenadas dos diferentes sensores. No passado, essa configuração era feita manualmente com base nas dimensões fornecidas pelo fabricante do carro, mas muitas vezes, estas medidas não são corretas porque usamos suportes que podem não estar alinhados corretamente e é difícil medir com precisão os ângulos de rotação entre o sistema de coordenadas de um sensor (câmera estéreo) e outro (LiDAR 3D ou Radar). Neste trabalho, propomos um método de calibração para dois pares de sensores (LiDAR 3D-Radar e câmera estéreo-Radar). A ideia é colocar os dados de todos os sensores em um sistema de coordenadas comum e alinhar os dados espacialmente. Este alinhamento serve para fazer posteriormente a fusão de dados com um único sistema de coordenadas. O método de calibração é baseado no algoritmo PnP que é usado entre outras aplicações em realidade aumentada ([ITSEEZ, 2016d](#)). Ele calcula a transformação entre o sistema de coordenadas do mundo em que é feita a renderização 3D e o sistema de coordenadas da câmera para depois projetar a imagem da renderização na imagem

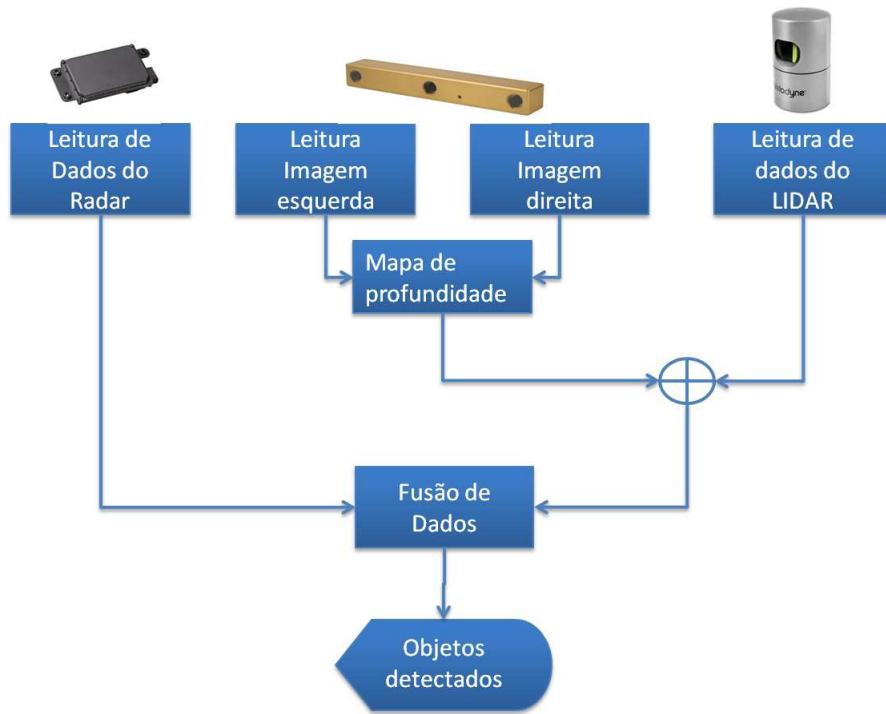
Figura 11 – Pose dos sensores no CaRINA II



Fonte: Elaborada pelo autor.

2D da câmera

Figura 12 – Fluxograma do sistema proposto para fusão de sensores



Fonte: Elaborada pelo autor.

Na Figura 12 é mostrado o fluxograma do processo de fusão. Primeiramente são coletados os dados dos sensores (Radar, LiDAR 3D e câmera estéreo). No caso do Radar o módulo Delphi ESR executa o processamento das ondas emitidas e refletidas pelos objetos e entrega 64 pontos pertencentes as detecções com as respectivas posições no espaço. No caso do LiDAR 3D, os

dados coletados são nuvens de pontos gerados pelo sensor. Estas nuvens de pontos representam posições 3D no espaço, com cerca de 70.000 pontos por amostragem (varredura de 360 graus ao redor do sensor). Os dados da câmera estéreo são imagens de um par de câmeras deviamente calibradas. Para gerar informações 3D precisamos processar as imagens usando um algoritmo de matching (SGM) para encontrar correspondências entre as duas imagens e depois precisamos usar o modelo pin-hole da câmera, juntamente com os parâmetros intrínsecos gerados na calibração, para obter o mapa de disparidade e o mapa de profundidade que representam a cena. O mapa de profundidade pode ser convertido em nuvem de pontos semelhante à gerada pelo LiDAR 3D. Finalmente, no processo de fusão de dados os pontos fornecidos pelo Radar são usados como hipóteses para iniciar a separação dos obstáculos nas nuvens de pontos emitida pelos sensores 3D. Realizando a dita separação (segmentação), é possível obter também um perfil e as dimensões para cada objeto.

4.2 Hardware

Para o desenvolvimento deste projeto temos disponíveis no Laboratório de Robótica Móvel (LRM) um Radar Delphi ESR, um LiDAR 3D Velodyne HDL-32E e uma câmera estéreo Bumblebee XB3 ([POINTGREY, 2014](#)). Os três sensores capacitados para uso na detecção de objetos usando diferentes métodos. Para realizar a coleta de dados o LRM possui duas plataformas experimentais, o CaRINA II (Veículo Fiat Palio Weekend) e o caminhão autônomo (Scania G360 de 9 ton.). As duas plataformas foram automatizadas para permitir a aquisição de dados e o controle computacional.

4.2.1 Radar Automotivo

Figura 13 – Modulo Sensor [Delphi ESR]



Fonte: [Delphi \(2016\)](#).

O módulo de Radar usado permite a detecção de obstáculos através de pulsos de Rádio Frequência (RF), sem elementos móveis no sensor, sendo portanto um dispositivo bastante

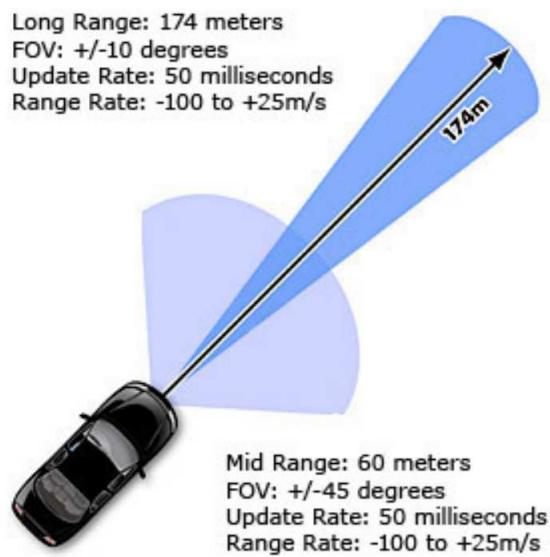
adequado para aplicações automotivas. A Figura 13 apresenta um dos módulos sensores adotado neste projeto. A interface entre o Radar e os dispositivos computacionais, que fazem uso das informações por ele providas é feita através do barramento CAN, baseado em um protocolo pré-definido de mensagens.

O módulo usado foi o Delphi ESR que é um dispositivo do tipo Millimeter Wave Radar (MMW), bimodal, que funciona a uma frequência de onda de 76,5 GHz. O Radar Delphi ESR possui um Digital Signal Processor (DSP) interno que faz o pré-processamento próprio do Radar. No caso do Radar empregado, isto corresponde a capturar apenas os dados de até 64 objetos detectados (o Radar fornece uma lista de tamanho máximo pré-definido, contendo os objetos detectados). Estes dados são disponibilizados através do barramento Controller Area Network (CAN), que serve de interface de comunicação entre o Radar e outras unidades de processamento.

O módulo ESR tem tecnologia de estado sólido e não tem partes móveis, tornando o sensor muito resistente a vibrações e robusto. Este dispositivo trabalha simultaneamente com a transmissão e recepção dos pulsos Doppler.

O ESR da Delphi combina dois tipos de medições: o primeiro tipo inclui uma medição de distância média (“Mid Range”, alcance de 60 metros) com um campo de visão(Field of View (FOV)) de $\pm 45^\circ$, e o segundo tipo de medição tem um alcance mais longo (“Long Range”, até 170 metros) e um FOV de $\pm 10^\circ$. As características gerais do Radar são ilustradas na Figura 14.

Figura 14 – Características dos Módulos Sensores – Mid e Long Range [Delphi ESR]



Fonte: Adaptada de [Delphi \(2016\)](#).

No modo “Long Range” o ESR consegue detectar muito bem veículos e principalmente obstáculos metálicos, no “Mid Range” é a área onde são detectados com maior acurácia os pedestres. Em todos os casos, o erro da detecção é de $\pm 0,5$ metros. O sensor fornece dados de

Tabela 2 – Mensagens CAN do modulo Delphi ESR

| | Inicio | ... | | | | | | | | | | | | Fim |
|----|--------|------|------|------|------|-----|------|------|------|------|------|------|------|-----|
| ID | 4E0h | 4E1h | 4E2h | 4E3h | 500h | ... | 53Fh | 540h | 5E4h | 5E5h | 5E6h | 5E7h | 5E8h | |

Fonte: Dados da pesquisa.

ângulo e velocidade do objeto em relação ao sensor e discrimina até 64 objetos na frente do veículo.

4.2.1.1 Protocolo de comunicações do Módulo ESR

Foi feito inicialmente um estudo teórico, seguido de testes práticos do protocolo, visando identificar mensagens e dados úteis para este projeto. Conseguimos ter acesso às informações do protocolo e mensagens para que fosse assim feita uma interpretação correta dos dados disponibilizados pelo Radar através do barramento CAN.

O ESR utiliza um protocolo CAN proprietário baseado na norma internacional ISO 11898 ([ISO, 1993](#)), sendo esta adaptada às necessidades específicas do módulo. O CAN descreve a camada física e a camada de enlace de dados e foi desenvolvido especialmente para a troca rápida de dados entre dispositivos eletrônicos em veículos automotivos, mas também pode ser usado em redes industriais. A ordem das mensagens CAN do modulo Delphi ESR é detalhada na [Tabela 2](#).

O DSP dentro do modulo ESR processa esse grupo de mensagens em um tempo de $50\text{ms} \pm 5\text{ms}$ (frequência de comunicação 20 Hz). Após este processo, o ESR transmite essas mensagens em grupo. De particular interesse para nós são as mensagens com ID (4EXh), que contém o estado do Radar (on-off), e as mensagens com ID (500h) até (53Fh) cada uma tendo 8 bytes, com informações dos 64 pontos que pode detectar o Radar. Dentre todas as informações fornecidas em cada mensagem, as mais importantes são a distância, o ângulo e a velocidade do objeto em relação ao Radar. As mensagens com ID 5EXh são dedicadas a mostrar o status das tarefas de alienação que o software proprietário da Delphi realiza. No nosso caso essas mensagens não são utilizadas.

A partir da informação do protocolo estudado foi desenvolvido um driver que permite a captura dos dados do Radar pelo computador para disponibilizá-los no framework ROS. Além disto, foram realizados testes visando verificar se as especificações eram respeitadas em testes de campo (onde foi inclusive identificado que usualmente o alcance do Radar tende a ser inferior ao alcance máximo definido pelo fabricante). Também foi realizada a fixação do Radar no veículo de testes visando o melhor posicionamento em relação a sua altura e buscado um melhor modo de fixação ao veículo.

Por fim, também foram feitas pesquisas sobre as informações providas pelo fabricante sobre o funcionamento do Radar, com dados que demonstram a forma como os objetos são

detectados. Isto motivou a discussão e direcionamento de pesquisas visando a fusão com outros sensores.

Figura 15 – Exemplo de detecções fornecidas pelo ESR na rodovia



Fonte: Elaborada pelo autor.

A Figura 15 ilustra o resultado do software desenvolvido para a detecção de obstáculos com o uso do Radar. É possível perceber que os obstáculos são identificados, porém sua dimensão mais exata não é retornada de modo preciso pelo dispositivo. Apesar da detecção de obstáculos com o Radar não permitir que as dimensões destes sejam mais bem definidas, as principais vantagens do Radar em relação aos demais sensores são: (i) robustez na detecção de obstáculos, mesmo em condições adversas (p.ex. iluminação fraca ou ausente, presença de poeira ou neblina, chuva fraca/moderada), condições estas que prejudicam o uso de abordagens baseadas em câmeras; (ii) baixo custo em comparação com os sensores do tipo LIDAR; (iii) simplicidade e robustez de instalação e uso do dispositivo sensor, uma vez que não depende de partes móveis ou elementos mecânicos de precisão.

4.2.2 Câmera Estéreo

A câmera estéreo adotada (Pointgrey Bumblebee XB3) mostrada na Figura 16, tem três lentes com afastamento horizontal de 12 centímetros entre si. Esta câmera possui lentes com distância focal de 6 milímetros e um campo de visão horizontal (Horizontal Field of View (HFOV)) de 60 graus. A resolução da imagem usada neste trabalho foi de 1280 x 960 pixels.

Apesar da detecção de obstáculos com o Radar não permitir que as dimensões destes sejam mais bem definidas, as principais vantagens do Radar em relação aos demais sensores são: (i) robustez na detecção de obstáculos, mesmo em condições adversas (p.ex. iluminação fraca ou ausente, presença de poeira ou neblina, chuva fraca/moderada), condições estas que prejudicam o uso de abordagens baseadas em câmeras; (ii) baixo custo em comparação com os sensores do tipo LIDAR; (iii) simplicidade e robustez de instalação e uso do dispositivo sensor, uma vez que não depende de partes móveis ou elementos mecânicos de precisão.

A câmera é conectada ao computador através de uma interface Firewire, sendo dados adquiridos e processados através do uso de funções da biblioteca do OpenCV integrada ao

Figura 16 – Camera bumblebee



Fonte: [PointGrey \(2014\)](#).

framework ROS. Apesar da câmera possuir três lentes, as imagens adquiridas são sempre processadas em pares, duas a duas, para a obtenção do mapa de profundidade. A variação do baseline (distância horizontal entre o par de câmeras selecionado) define a profundidade que é possível estimar a partir do mapa de distância gerado.

A câmera estéreo permite que se obtenha uma nuvem densa de pontos 3D (point cloud densa), pelo menos mais densa que os dados 3D do Radar e Velodyne, com uma melhor descrição das dimensões e do volume dos obstáculos presentes em seu campo de visão. No entanto, a câmera possui limitações referentes à precisão (erro) e ao alcance da estimativa da posição quanto à profundidade dos obstáculos na cena (de 10 a 20 mts). Além disto, câmeras por serem sensores passivos, dependente de iluminação externa, são sensíveis a variações da iluminação da cena, além de não operarem sob condições adversas (p.ex. poeira, neblina, chuva, fraca iluminação, visão ofuscada por excesso de luz).

4.2.3 LiDAR 3D

O LiDAR Velodyne HDL-32E mostrado na [Figura 17](#) possui 32 feixes laser, com uma abertura total vertical de 40 graus e espaçados verticalmente com 1.33 graus.

O Velodyne HDL-32E permite a captura de 70.000 pontos amostrados a partir da rotação de 360 graus de 32 feixes laser. A leitura dos dados do Velodyne é feita através da interface Ethernet, a uma taxa de amostragem de 10Hz, resultando em aproximadamente 700.000 pontos - Point Cloud 3D (coordenadas X,Y,Z) que representam a distância dos obstáculos até o sensor. A precisão do Velodyne é bastante grande, possuindo um erro de no máximo 1 a 2 cm, com um alcance máximo de 80 a 100 metros. O Velodyne possui drivers e pacotes de integração dos dados junto ao framework ROS.

O Velodyne apresenta a vantagem de ser extremamente preciso nas medições que realiza (erro muito baixo na estimativa de distâncias), e robusto a variações de iluminação. Além disso, opera inclusive em condições adversas (p.ex. poeira, neblina, chuva), desde que estas condições sejam de fracas a moderadas. No entanto, seu custo é relativamente elevado e a nuvem de pontos

Figura 17 – velodyne



Fonte: [Velodyne \(2016\)](#).

3D gerada é bastante esparsa (apenas 32 linhas), contra 480, 768 ou 960 linhas usualmente encontradas nas câmeras.

4.2.4 Plataformas experimentais CaRINA II e Caminhão autônomo

Atualmente o LRM possui duas plataformas experimentais para teste de aplicações para veículos autônomos. Na primeira etapa deste trabalho, onde se implementaram as tarefas básicas de detecção de objetos usando Radar, foi utilizada a plataforma de caminhão autônomo. Na fase de calibração e fusão de sensores, foi utilizada a plataforma CaRINA II que foi adaptada e automatizada de forma a operar tanto conduzido por um motorista, quanto em modo autônomo. O CaRINA II permite que este seja instrumentado com diferentes sensores, de acordo com as necessidades de cada projeto de pesquisa. No caso deste projeto, o mesmo foi instrumentado com o Radar Automotivo, LiDAR 3D(Velodyne) e Câmera Estéreo (bumblebee XB3). O CaRINA II também é equipado com computador a bordo para coleta e processamento de dados de todos os sensores envolvidos. A arquitetura geral da plataforma CaRINA II é descrita em [Fernandes et al. \(2014\)](#). As plataformas CaRINA II e caminhão autônomo e são mostradas na [Figura 8b](#) e [Figura 8c](#), respectivamente.

4.3 Software

Nesta secção apresentamos os principais frameworks usados neste trabalho.

4.3.1 OpenCV

Para a programação de algoritmos de visão computacional é utilizada a biblioteca Open Source Computer Vision (OpenCV) [Bradski \(2000\)](#) como principal ferramenta. OpenCV é

uma biblioteca de visão computacional mantida pela [Itseez \(2016c\)](#) e Intel. A biblioteca é multi-plataforma, gratuita para uso acadêmico e comercial podendo ser usada sob a licença open-source BSD. A biblioteca tem interfaces de C++, C, Python e Java, podendo ser integrada em aplicações desenvolvidas junto ao framework ROS. Neste projeto é usada principalmente para processamento de imagens e calibração de câmeras.

4.3.2 PCL

A Point Cloud Library (PCL) ([RUSU; COUSINS, 2011](#)) é um framework open source para processamento de nuvens de pontos 3D. Neste trabalho a biblioteca PCL será usada para processar nuvens de pontos de sensores 3D (câmera estéreo e “Velodyne”) uma vez que a PCL contém inúmeros algoritmos do estado da arte para filtragem, estimativa de features, reconstrução de superfície, registro, ajuste de modelos e segmentação. Esses algoritmos podem ser usados, por exemplo, para filtrar outliers de dados ruidosos, unir nuvens de pontos 3D, segmentar partes relevantes de uma cena, extrair pontos-chave e calcular descritores para reconhecer objetos no mundo com base em sua aparência geométrica ([POINTLOUDS.ORG, 2016](#)). A PCL pode ser usada para fins acadêmicos e comerciais usando a licença BSD.

A PCL é multi-plataforma disponível em plataformas Linux, MacOS, Windows e Android. No nosso caso, usamos a versão 1.8 desta biblioteca rodando em uma plataforma Linux Ubuntu 14.04.

4.3.3 ROS

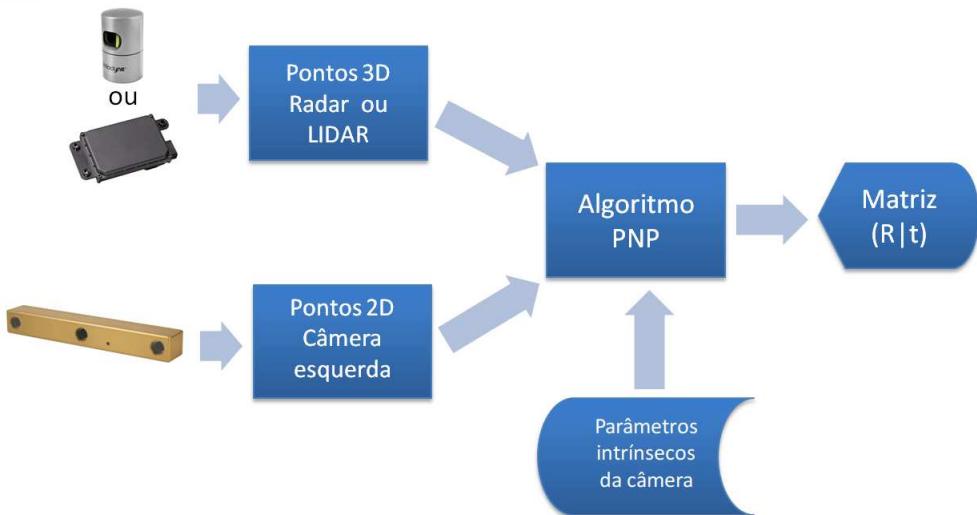
O desenvolvimento do sistema baseou-se no framework do Robot Operating System (ROS) ([QUIGLEY et al., 2009](#)) versão “Indigo”. O ROS é um framework que provê ferramentas para criar aplicações em robótica. É uma coleção de ferramentas, bibliotecas e convenções que visam simplificar a tarefa de criar um comportamento complexo e robusto em um robô podendo por exemplo capturar dados de diferentes sensores e disponibilizá-los em tópicos para depois ser usados em tarefas como fusão dos dados e detecção de objetos. Dentre as ferramentas disponíveis em ROS, neste trabalho são usadas principalmente o RViz que serve para visualizar os tópicos e mensagens enviadas pelos sensores, o Rosbag que serve para armazenar os dados e mensagens dos sensores para seu posterior processamento e análise off-line e o TF (Transform) responsável por armazenar os sistemas de referências dos sensores. O TF é responsável por realizar as transformações (rotação e translação) necessárias para passar de um sistema de referência para outro, operação importante para o caso da fusão de dados de sensores.

4.4 Calibração de sensores

Antes de realizar a detecção de obstáculos com uma câmera estéreo, deve-se realizar um procedimento de calibração para encontrar os parâmetros intrínsecos e extrínsecos da

câmera que permitam gerar os mapas de distância e de profundidade, e também projetar os pontos detectados pelo Radar na imagem. Também será importante a calibração para realizar transformações do sistema de coordenadas do Radar ou LiDAR 3D para coordenadas do sistema de coordenadas da câmera (como é explicado na subseção 2.2.1.3) para que, posteriormente os dados das detecções sejam fundidos.

Figura 18 – Diagrama de blocos do processo de calibração



Fonte: Elaborada pelo autor.

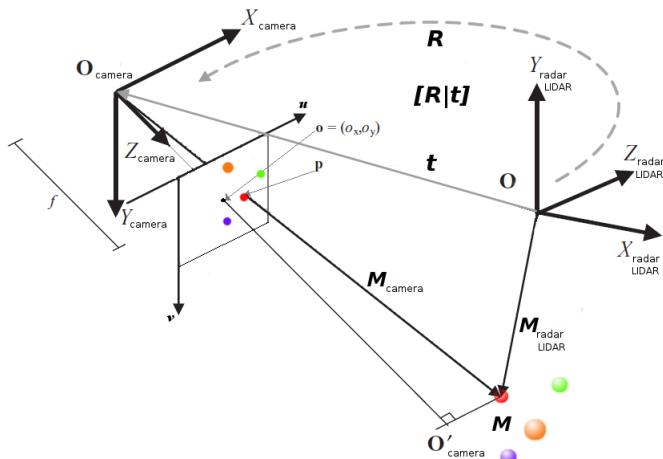
Neste trabalho é proposta uma metodologia para encontrar a pose da câmera em relação ao LiDAR ou Radar, respectivamente. Na Figura 18 é mostrado o processo para encontrar a transformação que permite levar pontos do sistema de coordenadas do Radar para o sistema de coordenadas da câmera ou pontos do sistema de coordenadas do LiDAR 3D para o sistema de coordenadas da câmera. Nos dois casos são necessários uma câmera calibrada (parâmetros intrínsecos) e o algoritmo usado para encontrar a pose (PnP).

Na Figura 19 podemos ver o modelo “pin-hole”, que serve como base para a calibração dos pares LiDAR 3D-câmera estéreo e Radar-câmera estéreo. Nele tem-se o sistema de coordenadas do sensor (LiDAR ou Radar) e um conjunto “M” de pontos que precisamos transladar/rotacionar para o sistema de coordenadas da câmera, no caso a matriz de transformação que precisamos encontrar é $[R|t]$, como mostrado na Figura 19. Considera-se a pose (posição+orientação) da câmera como sendo O_{camera} e a pose do Radar/Velodyne como sendo O . Através da matriz $[R|t]$ realiza-se uma transformação dos pontos relativos a origem O do Radar/Velodyne para o frame da câmera, relativo a sua pose, definida por O_{camera} .

4.4.1 Calibração da câmera estéreo

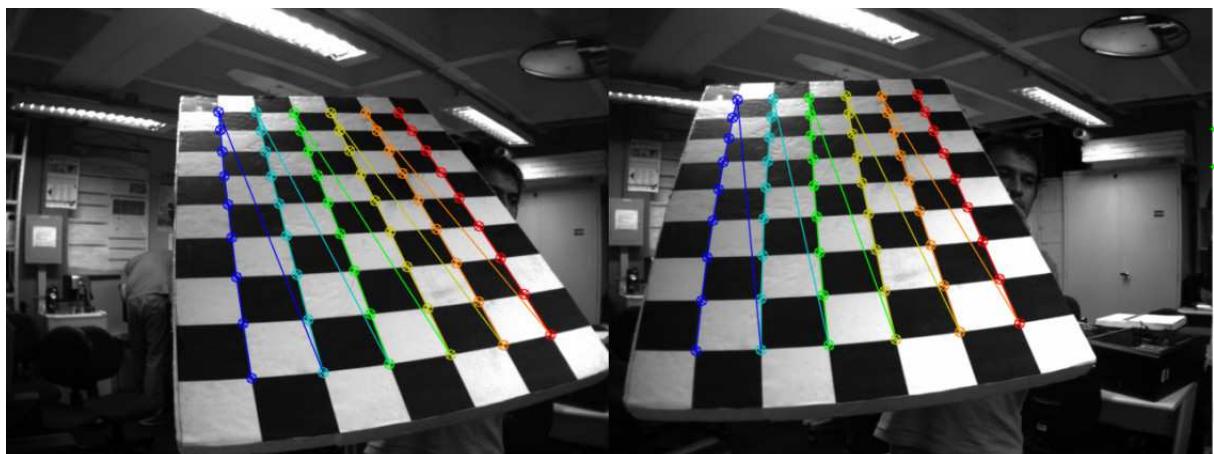
Para executar o processo de calibração da câmera estéreo, obtendo-se os parâmetros intrínsecos e extrínsecos desta, é usado um padrão xadrez composto por 10 quadros na vertical

Figura 19 – Modelo “pin-hole”



Fonte: Elaborada pelo autor.

Figura 20 – Processo de calibração da câmera estéreo usando padrão xadrez



Fonte: Elaborada pelo autor.

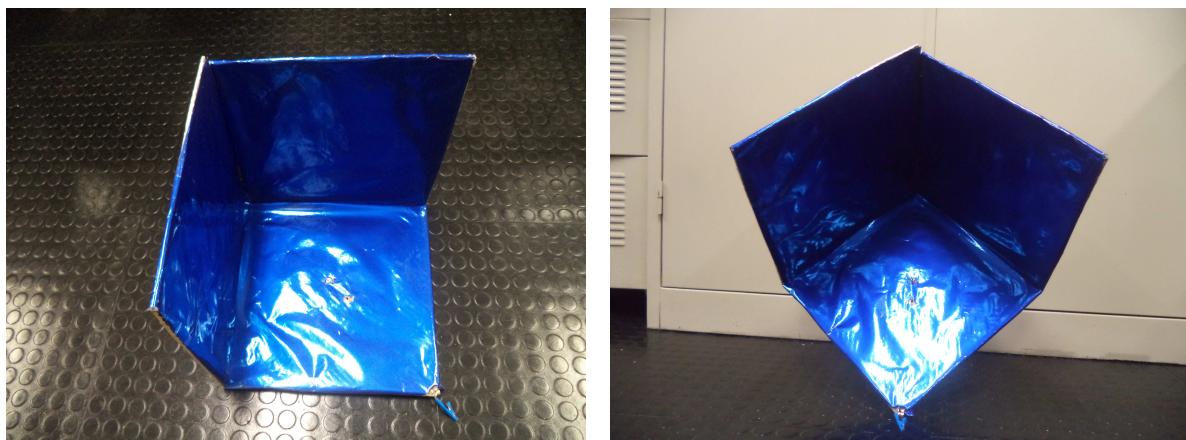
e 7 quadros na horizontal, cada quadro tendo uma área de 10 centímetros quadrados, como mostrado na Figura 20. Este padrão gera 54 features visíveis para a câmera. A biblioteca OpenCV é usada para realizar a calibração das câmeras, gerando os valores das matrizes que definem os parâmetros intrínsecos/extrínsecos para um determinado par de câmeras. Com estes parâmetros ajustados é possível fazer a correspondência entre os pixels da imagem e gerar o mapa de profundidade.

O modelo da câmera utilizado nas bibliotecas OpenCV é definido pelo modelo pin-hole, juntamente com os parâmetros de distorção. A função solvePnP mapeia pontos no mundo para pontos na imagem, baseada no modelo pin-hole e obtém os parâmetros de distorção e uma matriz de rotação e translação, representada na Equação 2.1.

4.4.2 Calibração do Radar em relação à câmera estéreo

Para executar satisfatoriamente a fusão dos dados do Radar e câmera estéreo é necessário alinhar os sensores. Nós usamos a estimativa da pose da câmera com relação a algum(s) objeto(s), neste caso, os objetos detectados pelo MMW Radar. Precisamos estimar a pose da câmera $[R|t]$ conforme indicado na [subseção 2.2.1.3](#) baseados no modelo pin-hole, [Figura 19](#). É preciso encontrar a pose da câmera para então executar tarefas como projetar/transformar um ponto do sistema de coordenadas do Radar (3D) para o sistema de coordenadas da imagem monociliar (2D) e depois para outros sistemas de referência. Este é um problema da estimativa da correspondência da pose de n pontos em 3D para n pontos 2D. Para tanto, será empregado o "Perspective- * n * -Point problem" (PnP), explicado na [subseção 2.2.1.3](#), que identifica vários pontos no sistema de coordenadas do mundo 3D (pontos do Radar) e suas correspondências no plano 2D da imagem. O que é desconhecido (a ser determinado) é a matriz de rotação e translação $[R|t]$ ([GAO et al., 2003](#)), ([LEPETIT; MORENO-NOGUER; FUA, 2009](#)).

Figura 21 – Refletor tipo Square Trihedral corner



Fonte: Elaborada pelo autor.

Tabela 3 – Refletores comumente usados para testar o Radar

| Tipo de Reflector (Corner) | RCS(m^2) |
|----------------------------|----------------------------------|
| Rectangular Dihedral | $\frac{8\pi w^2 h^2}{\lambda^2}$ |
| Triangular Trihedral | $\frac{4\pi L^4}{\lambda^2}$ |
| Square Trihedral | $\frac{12\pi L^4}{\lambda^2}$ |
| Circular Trihedral | $\frac{15.6\pi L^4}{\lambda^2}$ |

Fonte: Adaptada de [NAWC \(1997\)](#).

Para executar o algoritmo PnP corretamente, precisamos identificar as posições de n pontos 3D detectados pelo Radar em seu próprio sistema de coordenadas e seus equivalentes n pontos 2D na imagem da câmera esquerda. Para a tarefa de detecção dos pontos no sistema de coordenadas do Radar foram usados refletores do tipo Square trihedral corner colocados

sobre um cone de sinalização de trânsito. No nosso caso, foi construído um refletor usando três placas de alumínio (pelo fato de ser um bom condutor) com $L = 30\text{cm}$, espessura de 3mm e área de 30cm^2 ([Figura 21](#)). O RCS do refletor construído é de $19,873.6\text{m}^2$. Este valor foi calculado utilizando a fórmula da [Tabela 3](#), baseada na [Figura 1](#). O uso deste refletor assegura uma detecção contínua do alvo que será um dos pontos do Radar utilizado pelo algoritmo PnP. Também tiramos proveito da estabilidade na detecção para calcular uma posição mais precisa do alvo (para o mesmo alvo realizamos aproximadamente 100 leituras e a média é o ponto guardado). A detecção do refletor na posição adequada apresenta uma dispersão e desvio padrão menor do que aquela feita com outros alvos. Isso é uma vantagem quando se precisa de uma calibração adequada Radar-câmera estéreo. Para realizar a calibração, primeiramente os n pontos 3D (pontos brancos no meio dos pontos vermelhos na [Figura 23b](#)) são capturados em diferentes posições na área de operação do Radar para a mesma cena da câmera (carro estático). Para estas n diferentes posições 3D do Radar também são guardadas as suas correspondentes posições 2D na imagem.

[Figura 22 – Detecção baseada em HSV do target na imagem 2D](#)

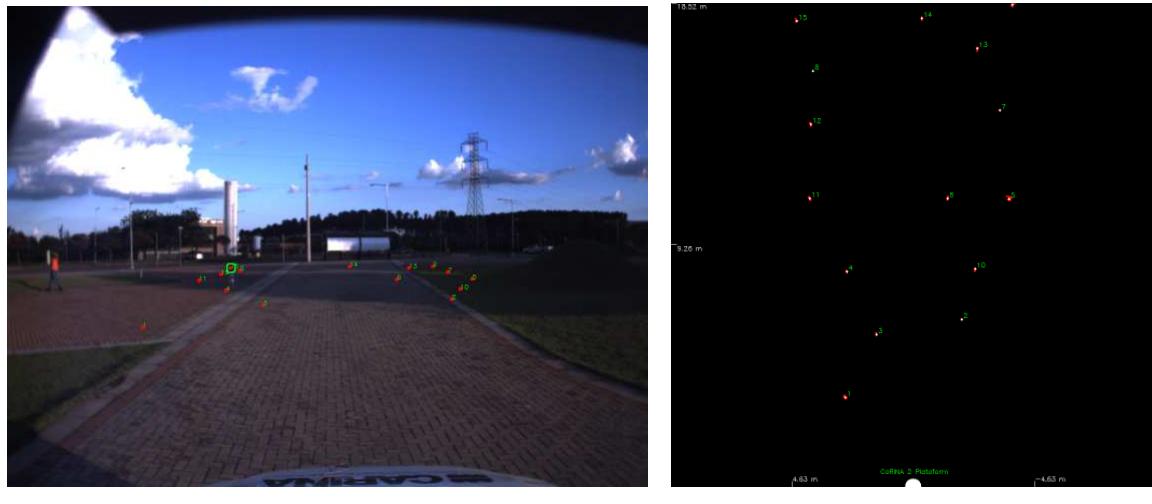


Fonte: Elaborada pelo autor.

Para capturar os pontos 2D realizamos a detecção automática de alvos com base em um limiar de cor (azul) utilizando o espaço HSV. A aplicação segmenta o objeto de cor azul e seu contorno, em seguida, o “momento” da área azul é calculado, o que nos dá o centro de massa do alvo [Figura 22](#). Finalmente o centro de massa é o ponto no sistema de coordenadas 2D da imagem (pontos vermelhos na [Figura 23a](#)) que são guardados para usar como os n pontos 2D da entrada do algoritmo PnP. A [Figura 23](#) mostra a captura de $n = 15$ pontos 3D no sistema de coordenadas do Radar e na imagem.

Como resultado deste processo de calibração, obtém-se a matriz $[R|t]$, que nos permite agora, dado qualquer ponto do Radar, realizar o seu mapeamento para as coordenadas de uma imagem (câmera). É importante destacar também que, uma determinada coordenada da imagem obtida pela câmera estéreo corresponde a uma posição no espaço 3D (x, y, z profundidade).

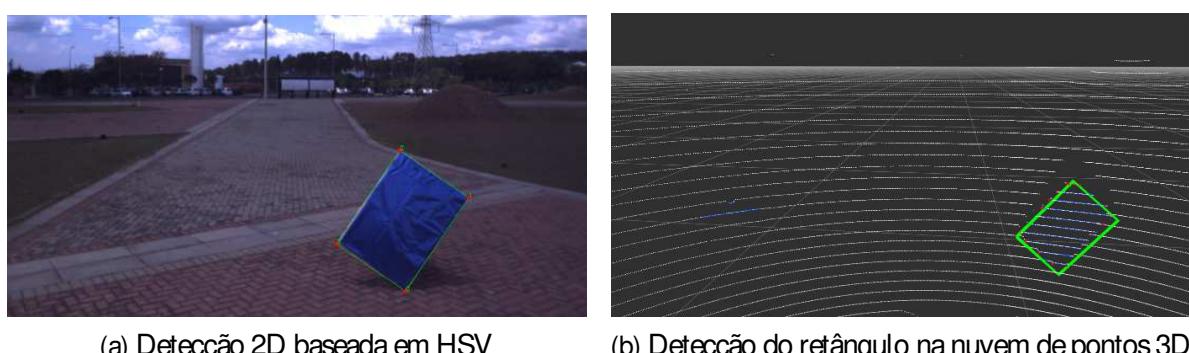
Figura 23 – Detecção de n pontos para entrada do algoritmos PnP usando Radar e Câmera estéreo



Fonte: Elaborada pelo autor.

4.4.3 Calibração do LiDAR 3D em relação à câmera estéreo

Figura 24 – Detecção do retângulo na imagem 2D e na nuvem de pontos 3D do Velodyne



Fonte: Elaborada pelo autor.

Para realizar a calibração LiDAR-câmera estéreo, precisamos n pontos 3D e seus n pontos correspondentes no plano 2D da imagem. Nesta parte foi usado o algoritmo PnP de maneira análoga à calibração Radar-câmera estéreo. Encontramos a pose ($[R|t]$) da câmera em relação ao LiDAR 3D. Como alvo (padrão de calibração) usamos um retângulo de “isopor” pintado de azul com dimensões de 100cmx70cm. O alvo é colocado em diferentes posições na frente do carro para permitir a detecção das coordenadas 3D dos quatro cantos do retângulo na nuvem de pontos do alvo gerada pelo LiDAR e as coordenadas 2D das quatro esquinas do retângulo azul da imagem.

Para capturar os n pontos 3D necessários para a entrada do PnP no sistema de coordenadas do LiDAR realizamos a segmentação da nuvem de pontos fornecida pelo sensor. Para facilitar o processo, os dados foram coletados em um lugar com o chão da cena totalmente plano e um único

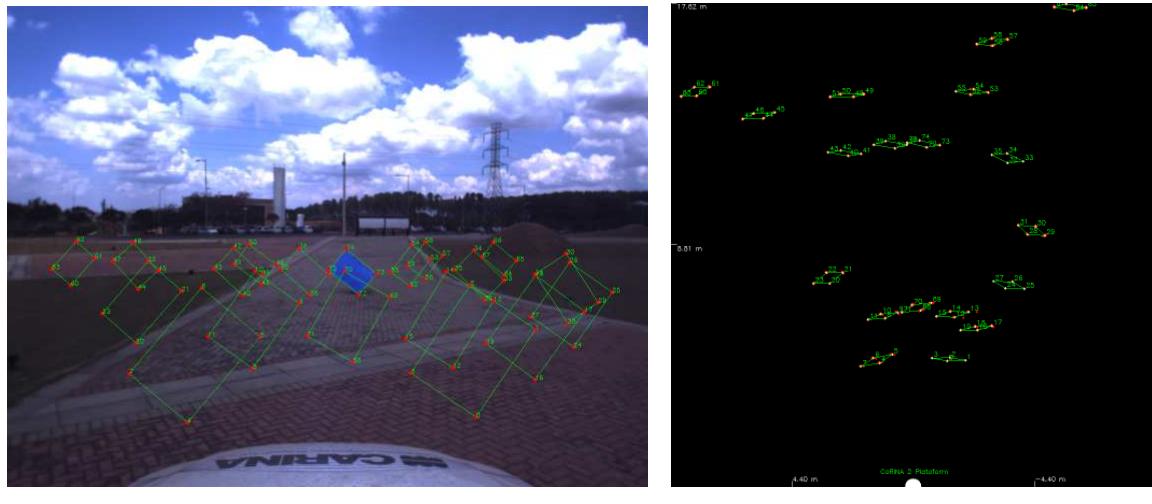
obstáculo (retângulo de isopor). O chão é removido da cena usando o algoritmo Difference of normals (DoN) (IOANNOU et al., 2012). O algoritmo DoN é utilizado para localizar regiões com similaridade nos vetores normais dos pontos. Essa similaridade serve neste caso para detectar e remover a área do chão. Após a remoção do chão e a aplicação de filtros (para remover ruído e outliers) obtemos os pontos que representam o retângulo colocado na frente do LiDAR como alvo. Os pontos 3D do alvo que queremos detectar são as coordenadas 3D dos quatro cantos do retângulo segmentado. Para obter essas quatro coordenadas 3D precisamos encontrar o retângulo que melhor se ajusta à nuvem de pontos resultante do passo anterior. Para realizar esse processo primeiro ajustamos a nuvem de pontos a um plano da forma $ax + by + cz + d = 0$ para facilitar cálculos geométricos. Depois disso encontramos o “convex hull” (conjunto de pontos que é necessário para formar um polígono de área mínima que inclui todos os pontos do plano) da nuvem de pontos do alvo. Usando os pontos do “convex hull” rodamos o algoritmo Rotating Calipers (DAVID, 2016) que serve para ajustar o mínimo retângulo que contém todos os pontos do conjunto. Na função `minAreaRect()` da OpenCV existe uma implementação do algoritmo Rotating Calipers (ITSEEZ, 2016e) que se baseia em Freeman e Shapira (1975) e é usado para ajustar o mínimo retângulo para um conjunto de pontos 2D. No nosso caso os pontos do “convex hull” são tridimensionais. Por essa razão para usar a implementação da OpenCV, fazemos uma transformação dos pontos 3D para uma outra posição 3D onde o valor para o eixo Z seja 0 e assim encontrar o retângulo em 2D e seus cantos. Para ter os cantos do retângulo em coordenadas 3D de novo, usamos a inversa da transformação usada e assim encontraremos as coordenadas dos quatro cantos do alvo original em 3D que servem como entrada para o algoritmo PnP. A Figura 24 mostra um exemplo da detecção do retângulo e de seus cantos.

À semelhança do que foi feito para encontrar o alvo na imagem 2D para a calibração Radar- câmera estéreo, aqui também usamos um detector do alvo com base na detecção da cor azul e sua borda. Uma vez encontrada a borda que envolve o retângulo (alvo) azul na imagem, aplicamos o algoritmo “Rotating Calipers” para encontrar o retângulo 2D de área mínima 2D que inclui o conjunto de pontos de borda do alvo e tomamos as coordenadas de seus quatro cantos, onde quatro dos n pontos 2D são usados como entrada para o algoritmo PnP. Na Figura 24a é apresentado um exemplo da detecção do retângulo na imagem da câmera esquerda e na Figura 24b temos as detecções correspondentes para o retângulo do alvo (linhas verdes) na nuvem de pontos 3D gerada pelo “Velodyne”. Os pontos vermelhos são o convex hull da nuvem de pontos do alvo.

A Figura 25 mostra a segmentação de 19 alvos e seus quatro cantos para um total de $n = 76$ pontos 3D no sistema de coordenadas do LiDAR e na imagem 2D.

A Figura 26 apresenta a visualização do alinhamento dos dados dos três sensores (Radar, câmera e LiDAR 3D) na imagem (2D) e no espaço 3D. Nos seguintes links podem ser visualizados vídeos do resultado da calibração: Para 2D: <https://www.youtube.com/watch?v=Oyg_w8d4JKc>.

Figura 25 – Detecção de n pontos para entrada do algoritmos PnP usando LiDAR 3D e Câmera estéreo



os sensores, a segmentação dos obstáculos é feita nas nuvens de pontos geradas pelos sensores 3D. Pequenas nuvens de pontos são recortadas em torno de cada detecção de Radar e para cada pequena nuvem de pontos é aplicado o filtro de Diferença de Normais (DoN do Inglês Difference of Normals ([IOANNOU et al., 2012](#))) para remover a superfície navegável do chão, que apesar de detectado pelo LiDAR 3D e a câmera estéreo não é considerado um obstáculo. Vários objetos podem aparecer nessa área filtrada. No entanto, utilizar um método simples de agrupamento para isoler o objeto na nuvem de pontos referenciados pelo Radar é suficiente. Um simples algoritmo baseado no “min-cut” ([GOLOVINSKIY; FUNKHOUSER, 2009](#)) foi aplicado com uma tolerância σ de 5.

4.5.1 Preprocessamento da nuvem de pontos

Quando se tem muitos objetos para identificar, como é o caso dos veículos autônomos em um ambiente urbano (pedestres, automóveis etc.) é necessário poder diferenciar os sinais que refletem em cada um destes. Estes devem ser separados completamente, já que cada um representa uma característica diferente própria a cada objeto, como por exemplo o seu tamanho, posição ou o movimento.

4.5.1.1 Filtros

Para reduzir o tempo de processamento, a nuvem de pontos é inicialmente recortada. Para isso usamos o sistema de coordenadas da câmera que é um sistema de mão direita com o X_{camera} alinhado com o u da imagem e Y_{camera} alinhado com o v da imagem (Y_{camera} para baixo, e X_{camera} orientado para a direita), o Z_{camera} fica para frente. O eixo Y_{camera} positivo é limitado a 2 metros que é uma medida ligeiramente maior que a altura do carro e limitado a 0,5 metros no eixo Y_{camera} negativo. Com este processo é possível eliminar os pontos que estão embalado dos pontos que representam o chão e os que estão 0,5 metros acima da altura do carro e que não são de interesse para a nossa detecção de objetos, como por exemplo, as copas das árvores e elementos que ficam em direção ao céu. Também são eliminados os pontos que ficam além de 40 metros de profundidade no eixo Z_{camera} e ± 6 metros no eixo X_{camera} . Este processo ajuda a eliminar pontos que não são de nosso interesse, pontos estes que ficam longe da rua e valores extremos da nossa nuvem de pontos.

Após a redução do tamanho da nuvem de pontos, um processo de reamostragem é aplicado para diminuir ainda mais o número total de pontos, denominado 3D voxel grid. Esta abordagem adota um conjunto de pequenas caixas 3D no espaço aplicadas sobre a point cloud de entrada. Em cada voxel todos os pontos são aproximados para o centro de deste (downsampled), a fim de reduzir o número de pontos a serem processados antes de realizar o agrupamento para cada objeto. A dimensão do voxel utilizado é de 0,01 m.

Também é necessário um pré-processamento de cada nuvem de pontos obtida a fim de eliminar o ruído nos dados, para cada nova nuvem de pontos recortada e reamostrada, é

aplicado um "Filtro da Média" para eliminar pontos ruidosos que podem afetar a segmentação dos obstáculos. Este filtro só foi necessário aplicar junto aos dados gerados pela câmera estéreo, pois nos dados fornecidos pelo sensor Velodyne não foi necessário esse processamento extra, uma vez que os dados obtidos por este sensor não são tão ruidosos como os provenientes da câmera. Para a obtenção dos dados 3D com a câmera estéreo foi adotado um algoritmo padrão para cálculo da disparidade (Semi Global Matching), tendo sido usado neste trabalho para obter os pontos 3D a partir das imagens de um par de câmeras estéreo.

4.5.1.2 Remoção de chão

Além dos processos de recorte, reamostragem e filtragem, descritos acima, antes de executar o algoritmo de segmentação de objetos, também é aplicado o algoritmo DoN para remover os pontos do chão conforme detalhado em ([IOANNOU et al., 2012](#)).

4.5.1.3 Segmentação dos obstáculos

Para realizar a segmentação dos obstáculos usamos um algoritmo de segmentação de nuvens de pontos baseado em “min-cut” ([GOLOVINSKIY; FUNKHouser, 2009](#)). Este algoritmo é implementado e disponível nas bibliotecas da PCL (Point Cloud Library) ([RUSU; COUSINS, 2011](#)). O algoritmo min-cut faz uma segmentação binária da nuvem de pontos de entrada baseada na distância dos pontos ao centro de referência definido pelo usuário (no caso o ponto da detecção de um obstáculo fornecido pelo Radar). Tendo as coordenadas das detecções dos objetos como sendo o centro de referência e definindo um raio em volta desses centros, são separadas (segmentadas) nuvens de pontos menores que a nuvem de pontos original. Depois o algoritmo classifica estes pontos dividindo cada nuvem de pontos em dois conjuntos: Pontos do objeto e junto ao objeto (chão ou outros objetos).

Foi definido um parâmetro $\sigma = 0,01m$, sendo este o limiar do espaçamento na nuvem de pontos após a reamostragem. Neste caso, o Radar fornece pontos que são o centro (do objeto) a ser extraído da nuvem de pontos, com um tamanho de 6x6 metros em torno de cada detecção. Para cada nova nuvem de pontos gerada, o algoritmo “min-cut” é aplicado para diferenciar entre os pontos pertencentes ao objeto detectado pelo Radar e pontos que não pertencem a esse objeto, como o chão. Nesse processo são processadas unicamente as áreas onde o Radar teve detectados obstáculos, assim é evitado um processamento extra dos dados de toda a nuvem de pontos original. O algoritmo de “min-cut” funciona da seguinte maneira ([RUSU; COUSINS, 2011](#)):

1. Para cada um dos 64 pontos detectados pelo Radar¹ é extraída uma nuvem de pontos quadrada (6x6 metros), tendo como centro cada um destes pontos de detecção do Radar;

¹ Radar Delphi ESR: detecta até 64 blobs representando obstáculos ao redor do sensor

2. A partir destas 64 novas nuvens de pontos, construímos um grafo para cada uma contendo todos os pontos e arestas, e mais duas arestas chamadas de source e sink. Cada aresta exceto source e sink estão ligadas com os seus vizinhos mais próximos;
3. O algoritmo atribui três tipos de pesos para cada aresta em cada nuvem de pontos:

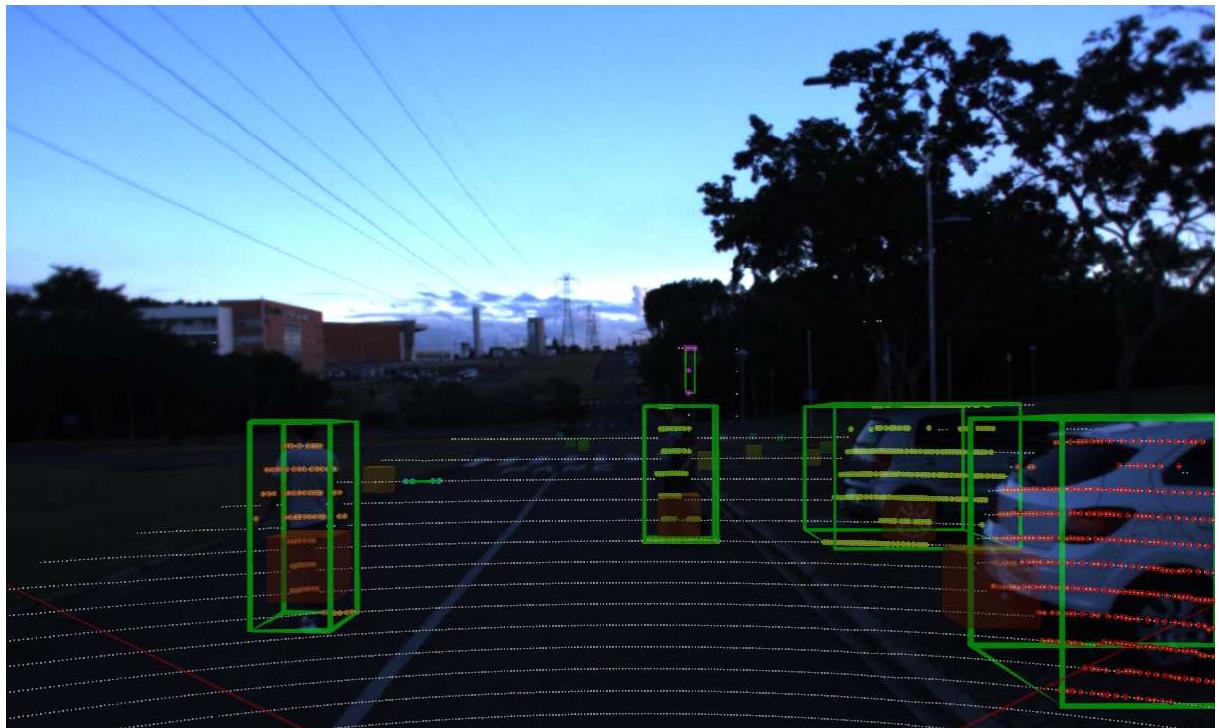
$$\text{smoothCost} = e^{-(\text{dist}/\sigma)^2} \quad (4.1)$$

$$\text{backgroundPenalty} = (\text{distanceToCenter}/\text{radius}) \quad (4.2)$$

$$\text{distanceToCenter} = \frac{q}{\sqrt{(x - \text{centerX})^2 + (y - \text{centerY})^2}} \quad (4.3)$$

4. Conforme os pesos atribuídos na etapa anterior para as arestas, os pontos ligados a estas são então diferenciados.

Figura 27 – Múltiplos objetos detectados na nuvem de pontos fornecida pelo LiDAR(Pedestres, carros e postes)



Fonte: Elaborada pelo autor.

A Figura 27 apresenta um exemplo do objetivo deste trabalho, realizar a fusão de uma imagem com os dados 3D. No caso, é apresentada a fusão da imagem da câmera com o LiDAR, já calibrados, a fim de obter a sobreposição dos mesmos. Além de sobrepor as informações capturadas, também é realizado um processamento de forma a detectar os obstáculos, destacados

pelas caixas (bounding box) que representam cada um destes obstáculos identificados na cena. Também é importante destacar que o chão, ou seja, a superfície navegável de suporte do veículo, não foi considerado um obstáculo, pois foi previamente detectado e tratado pelo algoritmo de detecção de obstáculos. Neste trabalho buscamos realizar o “encaixe” (fusão) dos dados 3D vindos do Radar, Câmera Estéreo e LiDAR. Para fins de visualização, serão apresentados exemplos onde é feita a sobreposição dos dados sobre uma das imagens do par estéreo (como apresentado na Figura 27).

4.5.2 Detecção de obstáculos usando “Stixel World”

Neste trabalho também é usado o cálculo do “Stixel World” para detecção de obstáculos. Foi usada a implementação apresentada em Benenson et al. (2012) onde se realiza o cálculo rápido do “Stixel World” para detectar obstáculos em uma imagem monocromática sem necessidade de calcular o mapa de profundidade. A aplicação executa automaticamente o cálculo da área pertencente ao solo e a distância dos Stixels pertencentes aos objetos. Na Figura 28 podemos ver um exemplo da aplicação rodando no nosso cenário de teste. Após o cálculo dos Stixels, estes são convertidos em nuvem de pontos usando a calibração da câmera para realizar a segmentação de objetos no mundo 3D. Para este método a aplicação de filtros adicionais não é necessária, devido a que a segmentação de obstáculos e remoção do solo é feita pelo método.

Figura 28 – “Stixel World” para detecção de objetos



Fonte: Elaborada pelo autor.

4.6 Metodologia de Avaliação

Para avaliar a proposta deste projeto de mestrado foram capturados dados de Radar, LiDAR 3D e da câmera estéreo em situações reais e controladas permitindo executar os algoritmos de calibração apresentados neste capítulo. Posteriormente o framework ROS pode salvar bags (arquivos de log) de dados dos sensores sincronizados, que podem ser processados off-line, sendo possível também criar um ground truth manualmente, no caso sobre a nuvem de pontos do Velodyne. O que permite uma comparação dos resultados obtidos pelos métodos de detecção com os resultados anotados manualmente (ground truth). Os experimentos, resultados e a avaliação são apresentados no próximo capítulo.

4.7 Considerações Finais

Neste capítulo foram apresentadas as principais ferramentas e métodos que foram utilizados no desenvolvimento do projeto. Foi possível descrever cada sensor e identificar seus pontos fortes e fracos e propor uma estratégia de fusão de sensores para melhorar o desempenho geral do sistema. Assim, podemos explorar os pontos fortes de um sensor, contrapondo seus pontos fracos com a fusão de um outro sensor, que pode compensar suas limitações: resolução, precisão, alcance, densidade de pontos e robustez à condições adversas.

Também propomos uma metodologia de calibração geral, simples e fácil de reproduzir, para dois pares de sensores (Radar-câmera e LiDAR-câmera).

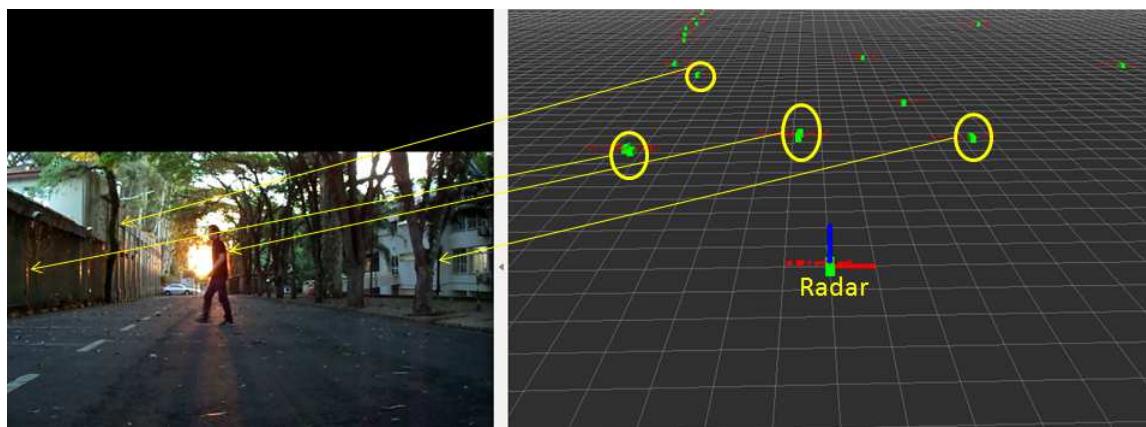
Baseados na calibração apresentamos as ferramentas e uma proposta para desenvolver um sistema de detecção de obstáculos através do uso de dados fornecidos pela câmera estéreo e Radar ou LiDAR. Foi enfatizado o uso de técnicas de fusão de sensores como proposta para obter melhores resultados nas detecções. Finalmente apresentamos o método de avaliação dos sistemas desenvolvidos.

EXPERIMENTOS E RESULTADOS

Neste capítulo serão apresentados os experimentos e os resultados práticos desenvolvidos bem como a análise e as conclusões finais deste trabalho de mestrado. Apresentamos também os artigos gerados e os trabalhos futuros.

5.1 Experimentos exploratórios realizados com Radar ESR e câmera

Figura 29 – Dados do Radar destacando obstáculos identificados árvores e pedestre



Fonte: Elaborada pelo autor.

No inicio deste projeto foram realizados experimentos para a aquisição dos dados fornecidos pelo Radar e com as imagens de uma câmera. Nesses testes foi utilizada apenas a imagem da câmera esquerda de uma Bumblebee XB3 ([POINTGREY, 2012](#)) da fabricante PointGrey.

Os experimentos realizados permitiram obter “logs” de dados (“bags” do ROS) gerados pelo Radar para a detecção de caminhões, carros, motocicletas, pedestres e outros objetos na

Figura 30 – Oclusão devida a poeira (problema para sistemas baseados em imagens)



(a) Cena sem poeira



(b) Oclusão ocasionada pela poeira em frente da pessoa

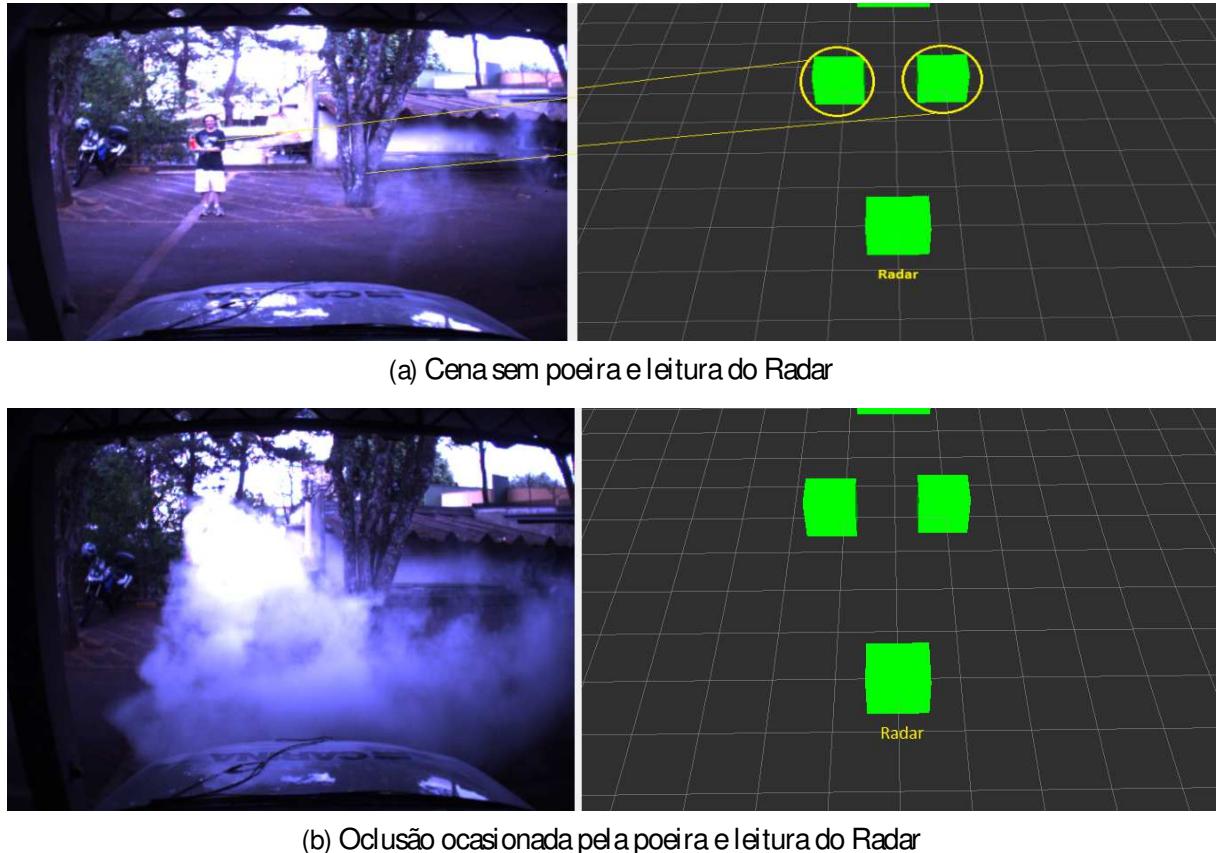
Fonte: Elaborada pelo autor.

rodovia e em ruas da cidade. Esses “logs” foram utilizados para avaliar o correto funcionamento e a correta configuração do Radar. Também através destes “bags” foi possível identificar as possíveis vantagens e desvantagens de usar esses dados para a detecção de objetos como carros, caminhões e pedestres. Esses testes serviram como base para propor uma fusão de dados com outros sensores como câmera estéreo.

Constatava-se pela imagem da [Figura 29](#) que os obstáculos detectados são representados por um ponto único junto a imagem, não sendo possível identificar o volume completo deste obstáculo que se apresenta junto a imagem. Esta visualização obteve um resultado similar ao programa de demonstração do fabricante do Radar, e como já foi indicado anteriormente, este tipo de detecção de obstáculos é limitado pois não permite identificar todo o contorno do objeto detectado.

A [Figura 29](#) apresenta um experimento inicial visando avaliar a detecção de obstáculos

Figura 31 – Experimentos realizados para análise da influência de neblina/poeira na detecção de obstáculos usando Radar



Fonte: Elaborada pelo autor.

pelo Radar em uma cena comum: uma via com muros, árvores e pedestre na frente do Radar. A Figura 30 apresenta um experimento que foi realizado com o objetivo de avaliar o impacto de uma nuvem de poeira (névoa, fumaça) em relação a visualização de obstáculos, usando um sistema baseado em câmera/imagens. Podemos constatar a dificuldade em identificar os obstáculos presentes na Figura 30b, devido a nuvem de fumaça. A Figura 31 apresenta uma nova cena deste experimento, mas agora usando Radar e câmera. Nesta figura podemos ver que o Radar é capaz de identificar corretamente a posição dos obstáculos mesmo na presença de poeira (fumaça/neblina).

5.2 Calibração de sensores

Esta seção descreve os experimentos realizados para validar a implementação das metodologias de calibração para os dois pares de sensores LiDAR 3D-câmera e Radar-câmera. Primeiro realizamos a calibração da câmera estéreo utilizando o método implementado na OpenCV (padrão xadrez), como resultado deste processo usando a função `calibrateCamera()` foram encontrados os parâmetros intrínsecos e extrínsecos da câmera estéreo (Bumblebee XB3).

A matriz K representa os parâmetros intrínsecos para cada câmera. Para calibrar os sensores Radar-câmera e Lidar 3D-câmera usamos o algoritmo PnP sendo necessário utilizar a matriz K de parâmetros intrínsecos da câmera da esquerda bem como a coleta de pontos no sistema de referência de cada sensor e assim encontrar a pose da câmera em relação aos sensores, tal como foi apresentado na [seção 4.4](#). A [Equação 5.1](#) mostra a matriz K encontrada no processo de calibração da câmera esquerda do par com menor separação (baseline) da bumblebee XB3.

$$K = \begin{bmatrix} \text{[F0 F1 F2 F3]} \\ 972.254700 & 0.0 & 645.829590 \\ 0.0 & 972.254700 & 468.294891 \\ 0.0 & 0.0 & 1.0 \end{bmatrix} \quad (5.1)$$

Dois experimentos foram realizados para a calibração de cada par de sensores, Radar-câmera e LiDAR 3D-câmera. Os sensores são colocados acima da plataforma experimental CaRINA II, o LiDAR 3D e a câmera estéreo ficam no teto do carro e o Radar automotivo na parte frontal da plataforma.

5.2.1 Radar-câmera

Para calibrar o Radar junto à câmera estéreo, são coletados 17 pontos 3D comuns na área de detecção do Radar e na área de visualização da câmera, o processo é feito deslocando o alvo 17 vezes e usando o método de coleta de dados descrito na [subseção 4.4.2](#). Esses 17 pontos detectados no espaço 3D e na imagem 2D são utilizados em conjunto com os parâmetros intrínsecos (matriz K) acima mencionados para alimentar o algoritmo PnP e assim estimar a matriz $[R|t]$ que é mostrada na [Equação 5.2](#). Na [Figura 32c](#) apresentamos graficamente a pose da câmera com relação ao Radar (matriz $[R|t]$), a [Figura 32a](#) e a [Figura 32b](#) são fotos da posição real dos sensores. Podemos conferir que o resultado obtido para a matriz $[R|t]$ coincide com a posição e rotação do câmera em relação ao Radar que temos na configuração da plataforma CaRINA II.

$$\begin{matrix} h & i \\ R|t & | \cdot \{z\} \\ \text{Radar - camera} \end{matrix} = \begin{bmatrix} 0.14301632 & -0.98971617 & 0.00287405 & 0.07903594 \\ 0.01275539 & -0.00106049 & -0.9999181 & 0.85587469 \\ 0.98963815 & -0.14304127 & -0.01247255 & 1.81385178 \end{bmatrix} \quad (5.2)$$

5.2.2 LiDAR-câmera.

Para realizar a calibração LiDAR 3D- câmera são usados 116 pontos encontrados a partir da detecção dos cantos de 29 quadrados de “isopor” como foi descrito no método da [subseção 4.4.3](#). Similar a como realizamos a calibração anterior, nesta parte usamos o algoritmo PnP. A [Equação 5.3](#) apresenta a matriz $[R|t]$ que determina a pose da câmera com relação ao LiDAR. A [Figura 33c](#) mostra o resultado de calibração LiDAR-câmera e representa graficamente a matriz $[R|t]$. A [Figura 33a](#) e a [Figura 33b](#) mostram a posição real da câmera em relação ao

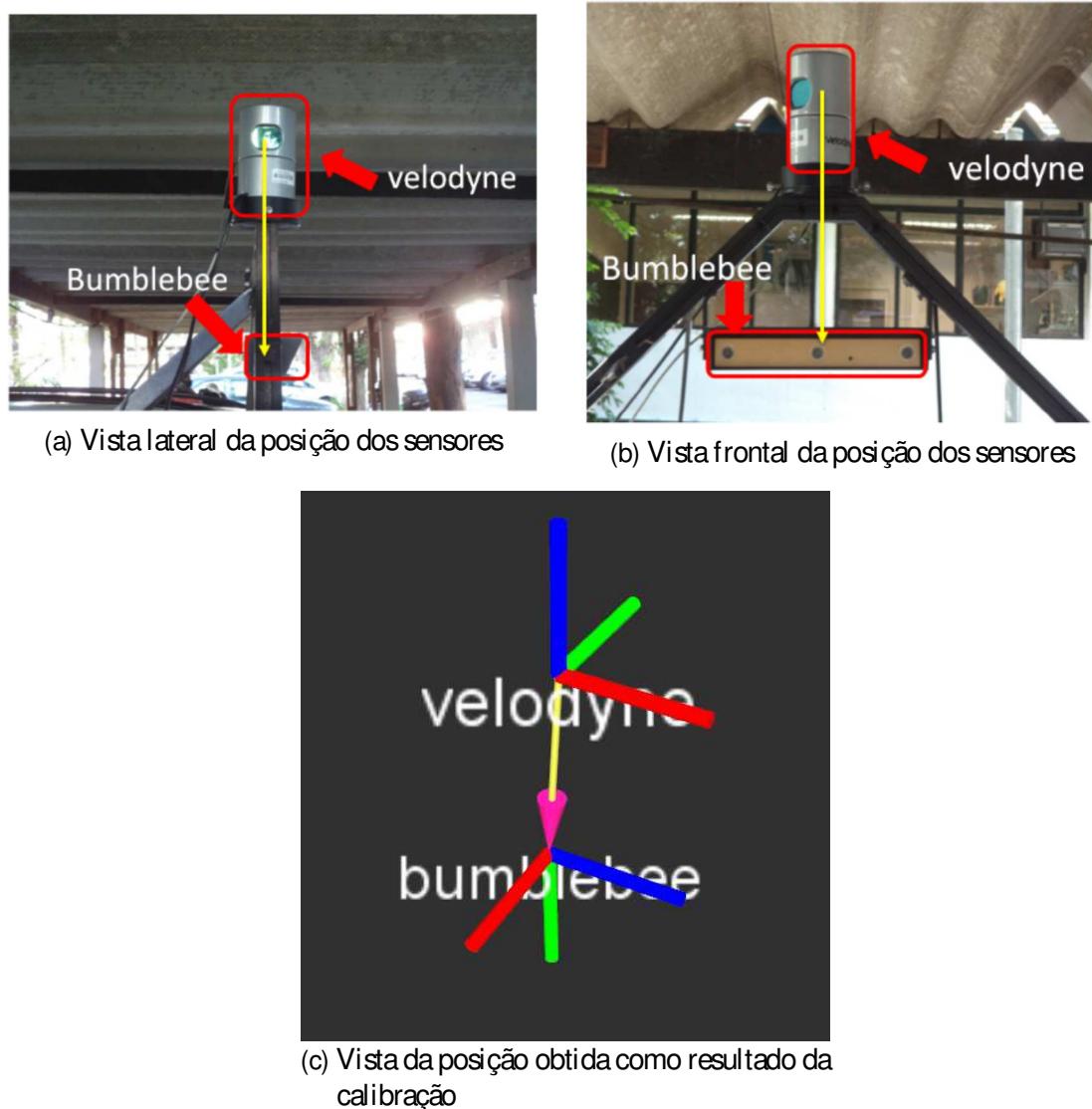
Figura 32 – Comparação entre a posição original e a posição relativa obtida na calibração dos sensores (Radar e câmera estéreo)



Fonte: Elaborada pelo autor.

LiDAR na configuração real do carro. Na Figura 33 podemos ver que na configuração real dos sensores, o eixo y da câmera da esquerda do par usado (câmera central das três presentes na bumblebee XB3) está alinhado com o eixo z do LiDAR e que a distância entre os sensores medida manualmente no eixo y da câmera é de aproximadamente -30 centímetros entre os dois sensores, igualmente na imagem da vista lateral (Figura 33a) podemos conferir que o deslocamento nos eixos x e z da câmera é quase nulo, valores todos estes que coincidem com os do vetor de translação t (coluna 4 da Equação 5.3) da matriz $[R|t]$ encontrada como resultado da calibração e cujas unidades são em metros. Visualmente podemos conferir que as rotações e translações da

Figura 33 – Comparação entre a posição original e a posição relativa obtida na calibração dos sensores (LiDAR e câmera estéreo)



Fonte: Elaborada pelo autor.

Figura 33c coincidem com as rotações dos sensores sobre a plataforma Carina II.

$$\begin{matrix}
 h & i \\
 R|t & = \\
 |. \{ z \} \\
 \text{LiDAR-camera}
 \end{matrix}
 \begin{matrix}
 \begin{smallmatrix} F_8 \\ EB \end{smallmatrix} & 0.0159695 & -0.99981761 & -0.01047323 & 0.00751909 \\
 \begin{smallmatrix} F_8 \\ EB \end{smallmatrix} & 0.00707279 & 0.01058726 & -0.99991894 & -0.29864089 \\
 & 0.99984747 & 0.01589413 & 0.00724058 & 0.00420434
 \end{matrix}
 \quad (5.3)$$

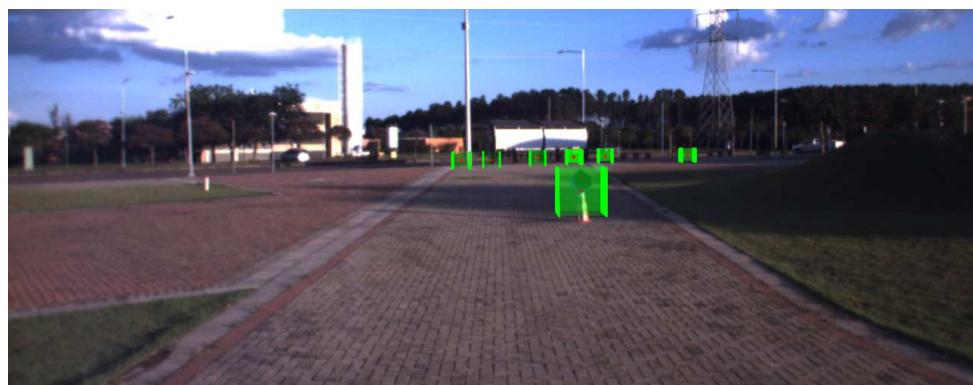
5.3 Validação da calibração

Para validar a calibração dos sensores, projetamos pontos 3D provenientes de obstáculos da cena em frente do carro e comparamos com a posição esperada.

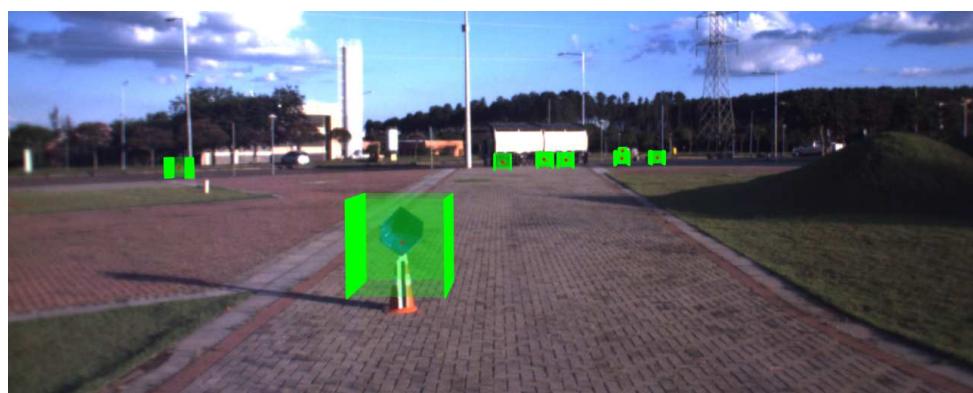
Figura 34 – Exemplos de erro de projeção dos pontos detectados pelo Radar



(a) Alvo a 6.69 m. aproximadamente do Radar.



(b) Alvo a 16.97 m. aproximadamente do Radar.



(c) Poste a aproximadamente 30 m. do Radar.

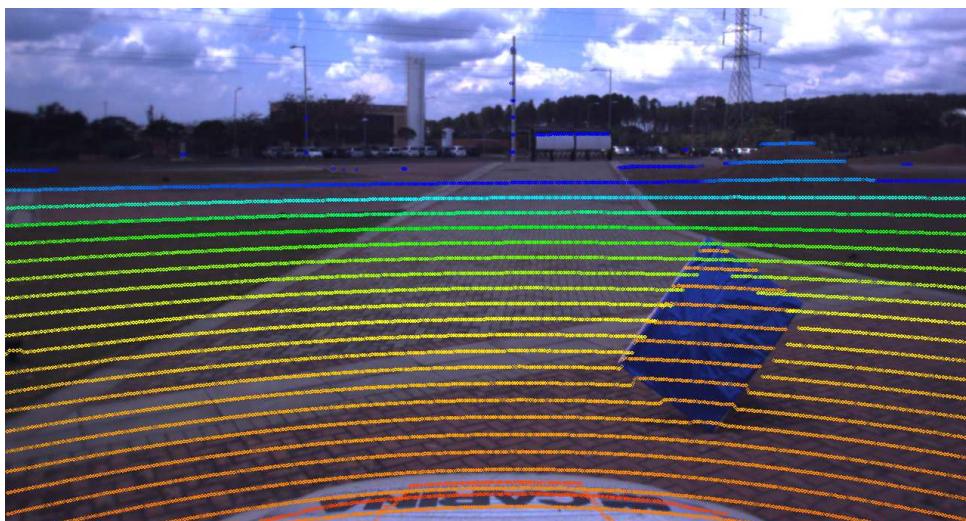
Fonte: Elaborada pelo autor.

5.3.1 Radar-câmera

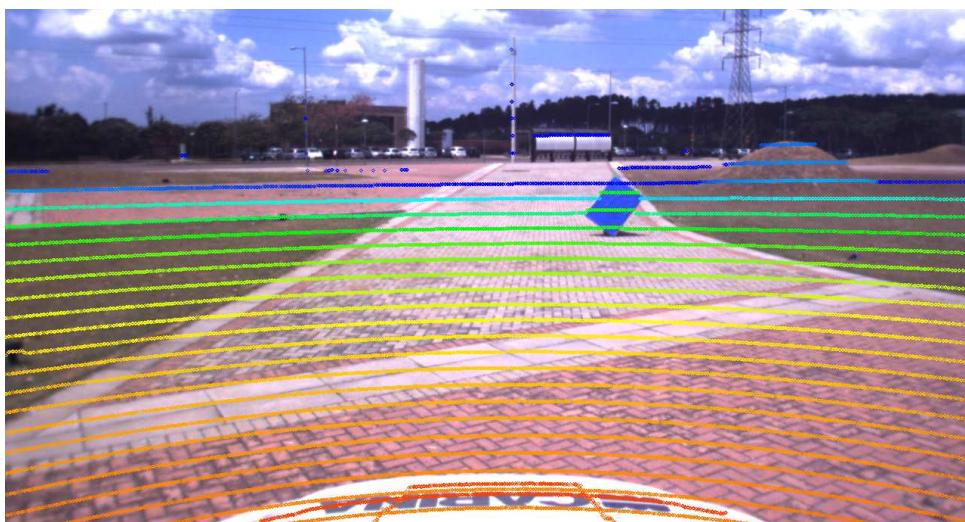
No caso da calibração Radar-câmera, projetamos as coordenadas de cada detecção feita pelo Radar (pontos vermelhos) usando as matrizes K e $[R|t]$ como mostrado na [Equação 2.1](#), essas projeções são comparadas com marcações feitas manualmente na imagem (pontos brancos) no lugar onde deveria estar a projeção do ponto da detecção feita pelo Radar. Para este experimento, verificou-se o erro usando a distância entre cada ponto projetado e o ponto marcado manualmente. O erro médio no eixo u da imagem foi de 9.7 pixels e no eixo v de 28.9 pixels. É importante

notar que os erros maiores ocorrem nas projeções de pontos que estão mais perto da câmera ([Figura 34a](#)) e o erro em pixels diminui em objetos mais distantes como acontece na [Figura 34b](#) e nos obstáculos no limite da área de detecção do Radar (postes no fundo da cena ([Figura 34c](#))) o erro é quase nulo. Grandes erros em pontos pertencentes a objetos próximos da imagem não são tão críticos, 30 pixels não vão significar valores maiores de 30 cm no mundo 3D. No experimento podemos notar que para todos os casos as projeções dos pontos pertencentes aos alvos azuis sempre ficaram dentro da região colorida do corner box construído para ser usado na detecção com Radar. A [Figura 34](#) mostra alguns exemplos das projeções dos pontos detectados. Para nossa aplicação essa tolerância de 30cm. é aceitável desde que o erro que o fabricante especifica para leituras de Radar é de ± 50 cm. O cubo verde representa esses ± 50 cm. em volta da detecção do Radar.

Figura 35 – Exemplos de erro de projeção dos pontos detectados pelo LiDAR 3D



(a) Alvo a 5.16 m. aproximadamente pelo LiDAR 3D.



(b) Alvo a 14.60 m. aproximadamente pelo LiDAR 3D.

Fonte: Elaborada pelo autor.

5.3.2 LiDAR-câmera

No caso da calibração LiDAR-câmera são projetados na imagem 2D todos os pontos da nuvem de pontos gerada pelo sensor. Como no experimento anterior são usadas as matrizes $K[R|t]$ para projetar os pontos da nuvem de pontos. Também são detectados os retângulos de “isopor” em frente do carro e projetados os seus cantos na imagem, e para fazer a comparação marcamos manualmente os cantos do quadrado na imagem, essas marcações servem como ground truth. Para este caso o erro médio de calibração foi de 4,54 pixels no eixo u e 4,93 pixels no eixo v da imagem. Do mesmo modo que no experimento com Radar-câmera, o maior erro ocorre nos pontos mais próximos da câmera ([Figura 35a](#)) e nos pontos distantes o erro é quase zero como por exemplo nos pontos pertencentes aos postes e a parada de ônibus no fundo da cena (indicados na [Figura 35b](#)). Outra observação que é importante fazer dos resultados obtidos é que aproximadamente o 100% dos pontos da nuvem de pontos pertencentes a cada quadrado projetados na imagem ficam sempre dentro da área da imagem correspondente (pontos laranja projetados sobre o quadrado na [Figura 35a](#)). Podemos ver exemplos da projeção da nuvem de pontos na [Figura 35](#) usando as matrizes $[K]e[R|t]$.

5.4 Detecção de Obstáculos

Em função da detecção limitada das dimensões exatas dos obstáculos, buscou-se estudar uma nova técnica que permitisse identificar junto a imagem, as dimensões dos contornos dos obstáculos identificados pelo Radar. O objetivo desta abordagem foi o de, considerando um ponto do Radar junto a imagem, tentar identificar o contorno do objeto na imagem associada a este ponto resultante da fusão Radar-câmera.

5.4.1 Fusão de sensores

Para avaliar a proposta deste trabalho foram capturados dados de Radar, Velodyne e câmera estéreo em situações reais e controladas utilizando o "framework" ROS ([QUIGLEY et al., 2009](#)). Os dados foram processados off-line. O “ground truth” foi criado manualmente em cima dos dados da nuvem de pontos do Velodyne, a fim de poder comparar os resultados obtidos pelos métodos de detecção automática do Radar e câmera estéreo juntamente com os resultados anotados manualmente do Velodyne.

Para este trabalho foram registrados 50 "frames" de dados com as informações dos três sensores. Foram marcados 325 objetos numa área de 30x5 metros (30 metros é a distância limite até aonde se pode ter uma boa estimativa da disparidade para a detecção de objetos com a câmera estéreo, e 5 metros é a distância para cada lado indo até a calçada). A área escolhida na frente do carro é uma área sensível onde os objetos podem ser considerados como potencialmente perigosos, pois podem interferir na trajetória prevista para o carro, e assim provocar um acidente. Os objetos que se encontram nesta área de 30x5 metros em frente ao

Tabela 4 – Resultados (100 pontos)

| Sensores | Verdadeiros Positivos | Falsos Positivos | Falsos Negativos |
|---------------------|-----------------------|------------------|------------------|
| Radar | 320 | 113 | 5 |
| Velodyne | 315 | 22 | 10 |
| Câmera stereo | 282 | 150 | 43 |
| Radar-Velodyne | 308 | 10 | 17 |
| Radar-Câmera stereo | 271 | 61 | 54 |
| Stixel World | 252 | 29 | 73 |

carro devem ser segmentados e monitorados, podendo no caso ser outros veículos, pedestres, ciclistas etc. No caso dos testes foram considerados como obstáculos outros veículos e pedestres. A Figura 36 apresenta exemplos da detecção de obstáculos usando a combinação Radar+Sensor 3D.

Mostra-se nas Tabelas (4, 5, 6) o desempenho de cada um dos três sensores separados (Radar, LIDAR 3D e Câmera Estéreo) e da fusão de sensores de Radar 3D (Radar-Velodyne, Radar-câmera estéreo). Os objetos foram segmentados e detectados de modo automático, sendo verificado o acerto/erro em relação ao ground-truth anotado à mão.

Na Tabela 4 mostra-se os resultados contemplando 100 pontos como número mínimo de pontos para que uma detecção seja considerada um obstáculo. Portanto, obstáculos muito pequenos (compostos por poucos pontos, abaixo de 100) são desconsiderados. Esta abordagem acaba gerando um maior número de Falsos Negativos (objetos não encontrados) e de Verdadeiros Positivos (acertos), pois este limitar aplicado ao número de pontos necessário para a detecção de um objeto, faz com que alguns objetos sejam erroneamente descartados na abordagem de fusão de Radar-Câmera Estéreo. Também ocorre um número relativamente elevado de erros, que estimou-se ser devido à detecção de pontos junto ao chão e bordas da pista, que não deveriam ser considerados.

Para a Tabela 5 uma detecção é considerada obstáculo a partir de uma altura de 50 cm, o que permitiu aumentar o número de Verdadeiros Positivos (acertos), uma vez que ocorreram menos erros na classificação. Por fim, na Tabela 6 foram considerados obstáculos a partir de 20 pontos, conformando assim uma detecção mais ampla de obstáculos menores, porém houve um pequeno aumento no número de Falsos Positivos na detecção Radar-Câmera Estéreo. É importante destacar que a melhor detecção em termos de Verdadeiros Positivos foi obtida com o uso do Radar, individualmente e em conjunto com outros sensores. O mesmo também ocorre como número de Falsos Negativos nas abordagens baseadas no uso do Radar nas Tabelas (5, 6). Um sistema de detecção de obstáculo voltado para a aplicação em veículos autônomos e inteligentes deve buscar ao máximo identificar todos obstáculos em seu caminho.

Para facilitar a visualização e comparação dos dados das Tabelas acima, foram criadas

Tabela 5 – Resultados (altura 50 cm)

| Sensores | Verdadeiros Positivos | Falsos Positivos | Falsos Negativos |
|----------------------|-----------------------|------------------|------------------|
| Radar | 320 | 113 | 5 |
| Velodyne | 315 | 22 | 10 |
| Câmera stereo | 282 | 150 | 43 |
| Radar-Velodyne | 320 | 39 | 5 |
| Radar-Câmera estereo | 320 | 68 | 5 |
| Stixel World | 252 | 29 | 73 |

Tabela 6 – Resultados (20 pontos)

| Sensores | Verdadeiros Positivos | Falsos Positivos | Falsos Negativos |
|----------------------|-----------------------|------------------|------------------|
| Radar | 320 | 113 | 5 |
| Velodyne | 315 | 22 | 10 |
| Câmera stereo | 282 | 150 | 43 |
| Radar-Velodyne | 320 | 51 | 5 |
| Radar-Câmera estereo | 320 | 78 | 5 |
| Stixel World | 252 | 29 | 73 |

mais duas Tabelas ([7](#), [8](#)) que agregam apenas os resultados das abordagens usando Radar e Câmera Estéreo (Tabela [7](#)) e depois Radar e Velodyne (Tabela [8](#)).

Tabela 7 – Comparação dos resultados de fusão considerando o tamanho e a quantidade de pontos que representam o obstáculo - Usando Radar e Câmera Estéreo

| Sensores | Verdadeiros Positivos | Falsos Positivos | Falsos Negativos |
|-------------------------------|-----------------------|------------------|------------------|
| Radar | 320 | 113 | 5 |
| Câmera stereo | 282 | 150 | 43 |
| Radar-Câmera stereo (100 pts) | 271 | 61 | 54 |
| Radar-Câmera estereo (50 cm) | 320 | 68 | 5 |
| Radar-Câmera estereo (20 pts) | 320 | 78 | 5 |

A partir dos resultados pode-se ver que 98,46% dos obstáculos presentes na cena foram detectados pelo Radar, porém com um relativo alto índice de falsos positivos. Utilizando-se apenas 3D LIDAR foram detectados 96,92% dos obstáculos e 86,76% usando a Câmera Estéreo.

Nota-se que o número de falsos positivos detectados individualmente para cada sensor, principalmente no caso do Radar e da Câmera Estéreo, é bastante alto. Este problema ocorre em nosso conjunto de dados, devido à presença das bordas da rua (guias) que são às vezes detectadas pelos sensores como se fossem obstáculos. Nos dados do Radar também aparecem

Tabela 8 – Comparação dos resultados de fusão considerando o tamanho e a quantidade de pontos que representam o obstáculo - Usando Radar e Velodyne

| Sensores | Verdadeiros Positivos | Falsos Positivos | Falsos Negativos |
|--------------------------|-----------------------|------------------|------------------|
| Radar | 320 | 113 | 5 |
| Velodyne | 315 | 22 | 10 |
| Radar-Velodyne (100 pts) | 308 | 10 | 17 |
| Radar-Velodyne (50 cm) | 320 | 39 | 5 |
| Radar-Velodyne (20 pts) | 320 | 51 | 5 |

falsos positivos em um grande número no meio da rua, estes falsos positivos podem ter sido provocados pela posição da instalação do Radar, relativamente baixa junto ao chassi do carro, e com uma inclinação que também pode ser inadequada para um melhor funcionamento do sensor. Está sendo considerado um melhor posicionamento deste sensor a fim de obter uma melhor coleta de dados do Radar.

Os resultados da fusão do Radar com os sensores 3D mostram uma redução de 91,15% dos falsos positivos para Radar-Velodyne e 46,02% para Radar-Câmera estéreo. Ambos valores são calculados em relação ao uso individual do Radar. Este resultado é bastante relevante, demonstrando a importância de trabalhar com mais de um sensor, visando melhorar a acurácia da detecção de obstáculos. Houve também uma redução de 54,54% de falsos positivos para a fusão Radar-Velodyne em relação aos valores calculados em relação ao uso individual do Velodyne. Para a fusão de Radar- Câmera Estéreo temos uma redução de 59,33% dos falsos positivos em relação à utilização de apenas Câmera Estéreo.

5.5 Considerações Finais

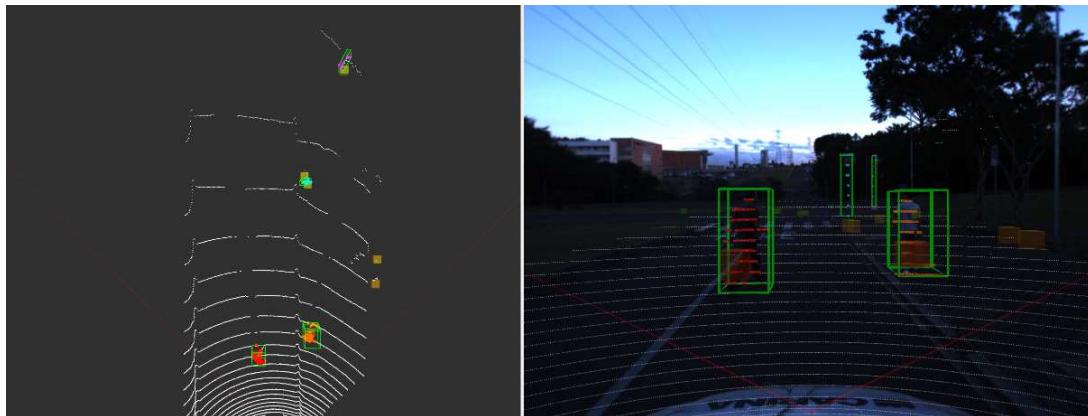
Os resultados obtidos e apresentados neste capítulo demonstram que a fusão de sensores, em abordagens híbridas (fusão de diferentes tipos de sensores), podem trazer vantagens quanto a melhoria da detecção de obstáculos. Comparando a detecção individual de cada sensor, com a abordagem de fusão de sensores, constata-se uma melhoria na qualidade de detecção dos obstáculos, principalmente no que se refere a fusão Radar-Câmera Estéreo.

É importante destacar que o sistema proposto permite atualmente que um determinado ponto detectado pelo Radar possa ser descrito através de sua representação espacial, ou seja, deixa de ser apenas um ponto de detecção no espaço 3D e passa a ter uma forma que representa a sua ocupação no espaço 3D (bounding box). Além disto, é possível tratar melhor as informações integrando inclusive a informação visual (imagem colorida), uma vez que existe um mapeamento entre as coordenadas de Radar, câmera estéreo e Velodyne.

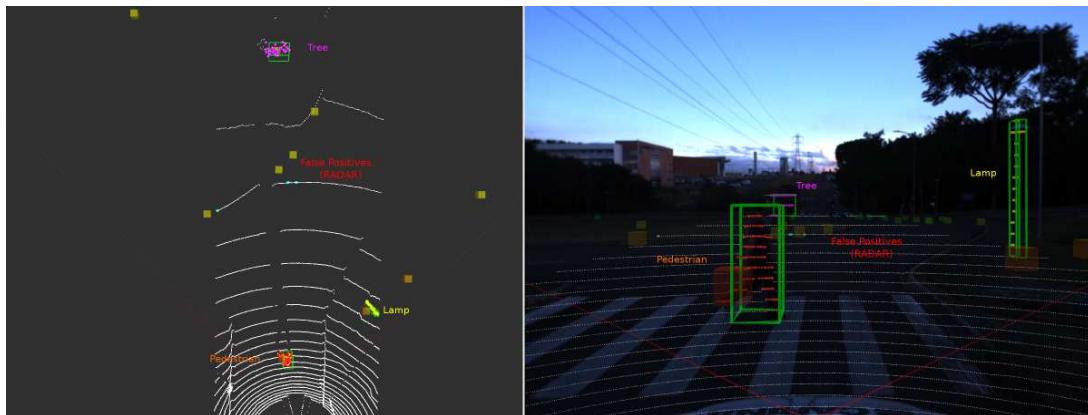
O nosso método é comparável com outros métodos rápidos de detecção de objetos como

o "stixel world" apresentado em [Benenson et al. \(2012\)](#) em termos de qualidade das detecções.

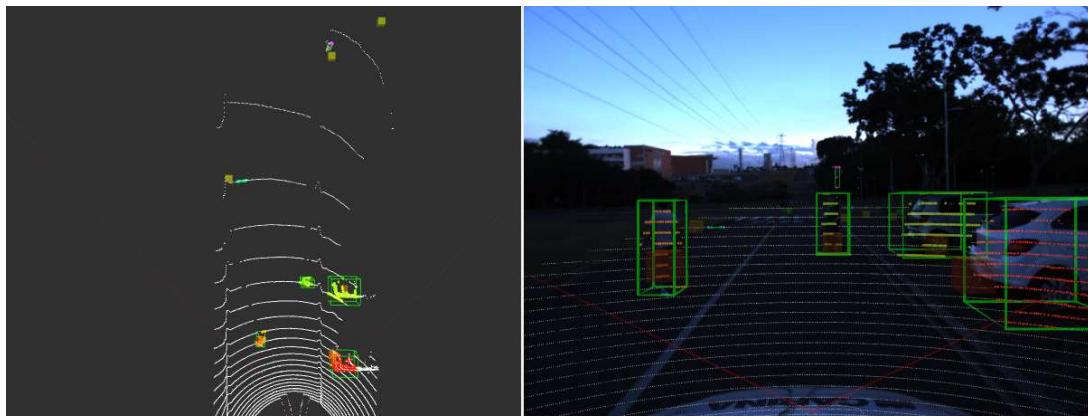
Figura 36 – Detecção de obstáculos usando Radar+Sensor 3D



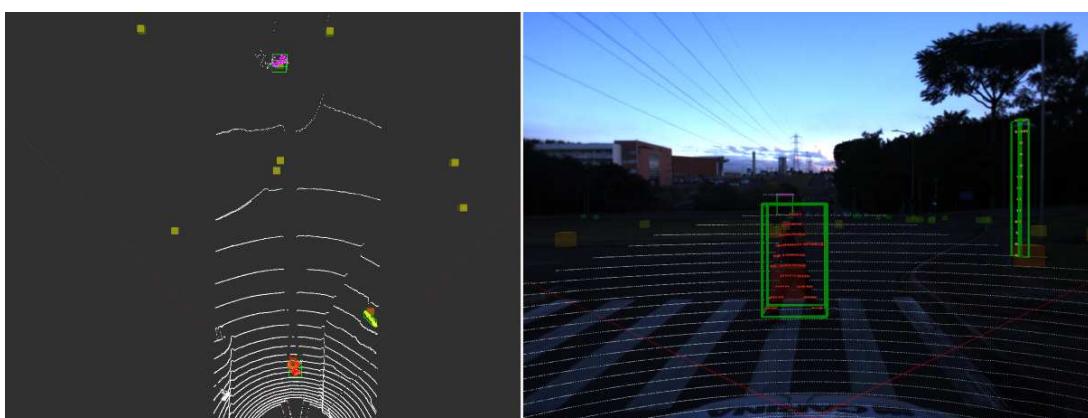
(a) Detecção de obstáculos (Radar e LIDAR)



(b) Eliminação de falsos positivos detectados pelo Radar.



(c) Detecção de obstáculos (Radar e Câmera Estéreo).



(d) Eliminação de falsos positivos detectados pelo Radar.

CAPÍTULO

6

CONCLUSÃO

Considerando os resultados deste trabalho, obtidos através da realização de diversos experimentos (conforme descrito acima), conclui-se que: (i) É de grande importância a fusão de informações originárias de diferentes sensores, uma vez que um único sensor apresenta sempre algum tipo de limitação (p.ex. Radar é pontual e não identifica todo o contorno dos obstáculos; câmeras estão mais sujeitas a problemas de reconhecimento de obstáculos – falsos positivos/falsos negativos – e a problemas de oclusão em caso de poeira e névoa, ou a variações na iluminação do ambiente); (ii) É importante que seja possível identificar os obstáculos com precisão, onde devemos minimizar ao máximo os falsos negativos (não detecção) e buscar uma definição mais precisa de suas dimensões; (iii) A fusão de informações entre o radar e uma câmera estéreo deve possibilitar uma melhor identificação de obstáculos, bem como deve permitir a identificação do contorno (3D) destes obstáculos.

Foi proposta uma metodologia de calibração para dois pares de sensores (Radar-câmera estéreo e LiDAR 3D-câmera estéreo) com base na estimativa da pose da câmera em relação à do sensor de profundidade. Esta metodologia pode ser estendida a qualquer outro sensor de profundidade (para exemplo um sensor de infravermelhos, câmera térmica, ultra-som, etc.) em conjunto com uma câmera. O método de calibração é geral e da mesma maneira como foi utilizado na plataforma Carina II pode ser portado para qualquer outra plataforma, sem precisar fazer modificações drásticas.

A qualidade de alinhamento dos dados obtidos neste trabalho é aceitável para a fusão de sensores proposta, uma vez que o alinhamento dos dados do Radar está dentro da faixa de 30 centímetros quadrados em relação aos dados da câmera. No caso do LiDAR é ainda mais baixa, com um erro menor a 10 pixels nos objetos perto da câmera que, em muitos casos, esse valor não supera os 10 centímetros no mundo 3D.

Os resultados da fusão do Radar com os sensores 3D mostram uma redução de 91,15% dos falsos positivos para Radar-Velodyne e 46,02% para Radar-Câmera estéreo. Ambos valores

são calculados em relação ao uso individual do Radar. Este resultado é bastante relevante, demonstrando a importância de trabalhar com mais de um sensor, visando melhorar a acurácia da detecção de obstáculos. Houve também uma redução de 54,54% de falsos positivos para a fusão Radar-Velodyne em relação aos valores calculados em relação ao uso individual do Velodyne. Para a fusão de Radar- Câmera Estéreo temos uma redução de 59,33% dos falsos positivos em relação à utilização de apenas Câmera Estéreo.

É importante destacar que o sistema proposto permite atualmente que um determinado ponto detectado pelo Radar possa ser descrito através de sua representação espacial, ou seja, deixa de ser apenas um ponto de detecção no espaço 3D e passa a ter uma forma que representa a sua ocupação no espaço 3D (bounding box). Além disto, é possível tratar melhor as informações integrando inclusive a informação visual (imagem colorida), uma vez que existe um mapeamento entre as coordenadas de Radar, câmera estéreo e Velodyne.

6.1 Principais Contribuições

As principais contribuições deste trabalho podem ser descritas de forma resumida como sendo:

- O desenvolvimento de uma metodologia própria para a calibração de dois pares de sensores (Radar-Câmera estéreo ou LiDAR 3D-câmera estéreo);
- O desenvolvimento de uma metodologia para detecção de obstáculos, sendo baseada no método de calibração de sensores proposto, e permitindo a fusão dos dados destes sensores;
- A melhoria do desempenho na detecção de obstáculos, através da adoção de uma abordagem híbrida de fusão de sensores, quando comparada ao uso individual de cada sensor;

O software desenvolvido neste trabalho é atualmente de grande importância no funcionamento atual das plataformas CaRINA II e do caminhão autônomo, permitindo assim que se façam testes tanto para a pesquisa e o desenvolvimento de veículos autônomos, quanto para demonstrações públicas.

No seguinte enlace é apresentado um vídeo de uma das demonstrações do caminhão autônomo usando o sistema de detecção de obstáculos com Radar desenvolvido neste trabalho: <<https://www.youtube.com/watch?v=3FiaYBk92C0>>.

6.2 Trabalhos Futuros

O primeiro trabalho futuro seria realizar percepção 360 graus usando vários Radares devidamente calibrados em torno do carro, isso vai proporcionar novas ferramentas para realizar

controle de cruzeiro, fazer mudanças de faixa e ultrapassar outros veículos em cenários complexos. Para conseguir isso, precisamos colocar o Radar nos lugares certos do carro e usar um método de calibração baseado na pose dos Radares. Esses dados poderiam ser fundidos também com dados provenientes do Velodyne e câmeras.

Propomos também o uso de algoritmos de aprendizado de máquina para a classificação dos objetos detectados, usando Deep Learning podemos classificar diferentes objetos, como pedestres, ciclistas, carros etc. Os algoritmos de aprendizado de máquina vão procurar apenas nas regiões da imagem onde foram detectados obstáculos, reduzindo assim o tempo de processamento para a classificação final dos objetos. Os algoritmos de aprendizado de máquina podem também ser treinados usando dados de vários sensores juntos no caso de câmera e LiDAR, fornecendo assim no treinamento do algoritmo uma maior quantidade de características relevantes como a profundidade.

O tracking dos objetos detectados também é um trabalho relevante onde pode ser usado um algoritmo como o filtro de Kalman. Também pode ser feita a segmentação dos objetos individualmente para cada sensor e realizar a uma fusão no nível de Declaração (fusão de alto nível) e usar algoritmos como Dempster Shafer para confirmar a presença de obstáculos baseados na incerteza da detecção em cada sensor.

A calibração de sensores e alinhamento de dados são tarefas importantes que podem ajudar a construir mapas mais precisos usando fusão de sensores como LiDAR, Radar e câmera estéreo, estes mapas podem ser construídos também pela remoção dos objetos que não fazem parte do lugar (ciclistas pessoas, carros), utilizando um classificador para que se saiba onde há uma área navegável para além da rua (onde existem carros estacionados).

Como trabalho futuro propomos usar outros recursos computacionais para processar os dados (nuvens de pontos e imagens) e também para o cálculo do mapa de distância. Também para melhorar o cálculo da distância é proposto o uso de múltiplas câmeras com diferentes linhas de base para alcançar a detecção em diferentes distâncias e o uso de novos métodos para o cálculo da distância e reconstrução 3D são mais rápidos e aplicados em GPU.

Finalmente usando a calibração correta dos sensores pretende-se coletar dados de diferentes cenários principalmente de dados de câmera e de sensores de profundidade e disponibilizar para que outras pessoas possam testar algoritmos de fusão de sensores sem preocupar-se pela calibração dos dados e focar-se no desenvolvimento do algoritmo.

6.3 Publicações geradas

ALENCAR, F. A.; ROSERO, L. A.; FILHO, C. M.; OSÓRIO, F. S.; WOLF, D. F. et al. Fast metric tracking by detection system: Radar blob and camera fusion. In: IEEE. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics

(LARS-SBR). [S.I.], 2015. p. 120–125.

SANTOS, T. C. d.; GÓMEZ, A. E.; FILHO, C. M.; GOMES, D.; PERAFAN, J. C.; WOLF, D. F.; OSÓRIO, F. and ROSERO, L. A. A simulation framework for multi-vehicle communication. In: IEEE. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). [S.I.], 2015. p. 301–308.

ROSERO, L.; and OSÓRIO, F. Detecção de pedestres e obstáculos usando visão computacional e radar em veículos automotivos. In CTDR/WTDR - Workshop on M Sc Dissertation and PhD Thesis in Robotics. SBC, 2016.

SHINZATO, P; DOS SANTOS, T. C.; ROSERO, L. A.; RIDEL, D.; MASSERA, C. A.; ALENCAR, F. A.; HATA, A.; BATISTA, M.; OSÓRIO, F.; WOLF, D.; CaRINA Dataset: An Emerging-Country Urban Scenario Benchmark for Road Detection Systems. 2016 IEEE 19th International Conference on Intelligent Transportation Systems, Rio de Janeiro, 2016, No prelo.

REFERÊNCIAS

- ALENCAR, F. A.; ROSERO, L. A.; FILHO, C. M.; OS, F. S.; WOLF, D. F. et al. Fast metric tracking by detection system: Radar blob and camera fusion. In: IEEE. 2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR). [S.I.], 2015. p. 120–125. Citado na página 34.
- ANSAR, A.; DANILIDIS, K. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 25, n. 5, p. 578–589, 2003. Citado na página 28.
- BADINO, H.; FRANKE, U.; PFEIFFER, D. The stixel world-a compact medium level representation of the 3d-world. In: SPRINGER BERLIN HEIDELBERG. Joint Pattern Recognition Symposium. [S.I.], 2009. p. 51–60. Citado 2 vezes nas páginas 29 e 30.
- BENENSON, R.; MATHIAS, M.; TIMOFTE, R.; GOOL, L. V. Pedestrian detection at 100 frames per second. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. [S.I.: s.n.], 2012. p. 2903–2910. ISSN 1063-6919. Citado 3 vezes nas páginas 30, 60 e 75.
- BENEZETH, Y.; EMILE, B.; LAURENT, H.; ROSENBERGER, C. Vision-based system for human detection and tracking in indoor environment. *International Journal of Social Robotics*, Springer, v. 2, n. 1, p. 41–52, 2010. Citado 2 vezes nas páginas 20 e 34.
- BERTOZZI, M.; BOMBINI, L.; CERRI, P.; MEDICI, P.; ANTONELLO, P.; MIGLIETTA, M. Obstacle detection and classification fusing radar and vision. In: Intelligent Vehicles Symposium, 2008 IEEE. [S.I.: s.n.], 2008. p. 608–613. ISSN 1931-0587. Citado 3 vezes nas páginas 20, 34 e 35.
- BRADSKI, G. Dr. Dobb's Journal of Software Tools, 2000. Citado na página 48.
- CYGANEK, B.; SIEBERT, J. P. An introduction to 3D computer vision techniques and algorithms. [S.I.]: John Wiley & Sons, 2011. Citado na página 29.
- DAVID. Computing oriented minimum bounding boxes in 2D. 2016. Disponível em: <<https://geidav.wordpress.com/2014/01/23/computing-oriented-minimum-bounding-boxes-in-2d/>>. Acesso em: 28 nov. 2016. Citado na página 55.
- DELPHI. Electronically Scanning Radar. 2016. Disponível em: <<http://www.delphi.com/manufacturers/auto/safety/active/electronically-scanning-radar>>. Acesso em: 28 nov. 2016. Citado 2 vezes nas páginas 43 e 44.
- DEMENTHON, D. F.; DAVIS, L. S. Model-based object pose in 25 lines of code. *International journal of computer vision*, Springer, v. 15, n. 1-2, p. 123–141, 1995. Citado na página 28.
- FERNANDES, L. C.; SOUZA, J. R.; PESSIN, G.; SHINZATO, P. Y.; SALES, D.; MENDES, C.; PRADO, M.; KLASER, R.; MAGALHÃES, A. C.; HATA, A. et al. Carina intelligent robotic car: architectural design and applications. *Journal of Systems Architecture*, Elsevier, v. 60, n. 4, p. 372–392, 2014. Citado 2 vezes nas páginas 36 e 48.

FIORE, P. D. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, v. 23, n. 2, p. 140–148, 2001. Citado na página [28](#).

FREEMAN, H.; SHAPIRA, R. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Communications of the ACM*, ACM, v. 18, n. 7, p. 409–413, 1975. Citado na página [55](#).

FRITSCHÉ, P.; KUEPPERS, S.; BRIESE, G.; WAGNER, B. Radar and lidar sensorfusion in low visibility environments. In: ICIINCO - 13th International Conference on Informatics in Control, Automation and Robotics. [S.I.: s.n.], 2016. p. 1–8. Citado na página [34](#).

GAO, X.-S.; HOU, X.-R.; TANG, J.; CHENG, H.-F. Complete solution classification for the perspective-three-point problem. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 25, n. 8, p. 930–943, ago. 2003. ISSN 0162-8828. Citado 2 vezes nas páginas [28](#) e [52](#).

GERONIMO, D.; LOPEZ, A. M.; SAPPA, A. D.; GRAF, T. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 32, n. 7, p. 1239–1258, 2010. Citado 2 vezes nas páginas [20](#) e [34](#).

GOLOVINSKIY, A.; FUNKHOUSER, T. Min-cut based segmentation of point clouds. In: *IEEE Workshop on Search in 3D and Video (S3DV) at ICCV*. [S.I.: s.n.], 2009. Citado 2 vezes nas páginas [57](#) e [58](#).

HALLER, I.; PANTILIE, C.; ONIGA, F.; NEDEVSCHI, S. Real-time semi-global dense stereo solution with improved sub-pixel accuracy. In: *IEEE. Intelligent Vehicles Symposium (IV)*, 2010 IEEE. [S.I.], 2010. p. 369–376. Citado na página [29](#).

HARTLEY, R. I.; ZISSELMAN, A. *Multiple View Geometry in Computer Vision*. Second. [S.I.]: Cambridge University Press, ISBN: 0521540518, 2004. Citado na página [28](#).

HEUEL, S.; ROHLING, H. Pedestrian classification in automotive radar systems. In: *IEEE. 2012 13th International Radar Symposium*. [S.I.], 2012. p. 39–44. Citado na página [34](#).

_____. Pedestrian recognition in automotive radar sensors. In: *IEEE. 2013 14th International Radar Symposium (IRS)*. [S.I.], 2013. v. 2, p. 732–739. Citado 2 vezes nas páginas [34](#) e [35](#).

HIRSCHMULLER, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: *IEEE. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.I.], 2005. v. 2, p. 807–814. Citado na página [29](#).

HORAUD, R.; DORNAIKA, F.; LAMIROY, B. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision*, Springer, v. 22, n. 2, p. 173–189, 1997. Citado na página [28](#).

HORNUNG, A.; WURM, K. M.; BENNEWITZ, M.; STACHNISS, C.; BURGARD, W. Octo-map: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots*, Springer, v. 34, n. 3, p. 189–206, 2013. Citado na página [36](#).

II, M. L.; HALL, D.; LLINAS, J. *Handbook of multisensor data fusion: theory and practice*. CRC press, 2008. Citado na página [31](#).

- IOANNOU, Y.; TAATI, B.; HARRAP, R.; GREENSPAN, M. Difference of normals as a multi-scale operator in unorganized point clouds. In: . [S.I.]: IEEE Computer Society, 2012. p. 501–508. Citado 3 vezes nas páginas 55, 57 e 58.
- ISO. Iso 11898, 1993. Road vehicles—interchange of digital information—Controller Area Network (CAN) for high-speed communication, 1993. Citado na página 45.
- ITSEEZ. Camera Calibration and 3D Reconstruction — OpenCV. 2016. Disponível em: <http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html>. Citado 2 vezes nas páginas 26 e 27.
- _____. Depth Map from Stereo Images. opencv.org/: [s.n.], 2016. Disponível em: <http://docs.opencv.org/3.1.0/dd/d53/tutorial_py_depthmap.html>. Acesso em: 28 nov. 2016. Citado na página 29.
- _____. OpenCV. opencv.org/: [s.n.], 2016. Disponível em: <<http://opencv.org/>>. Acesso em: 28 nov. 2016. Citado 2 vezes nas páginas 27 e 49.
- _____. Real Time pose estimation of a textured object. 2016. Disponível em: <http://docs.opencv.org/3.1.0/dc/d2c/tutorial_real_time_pose.html>. Acesso em: 28 nov. 2016. Citado na página 41.
- _____. rotcalipers.cpp. 2016. Disponível em: <<https://github.com/opencv/opencv/blob/master/modules/imgproc/src/rotcalipers.cpp>>. Acesso em: 28 nov. 2016. Citado na página 55.
- KALMAN, R. E.; BUCY, R. S. New results in linear filtering and prediction theory. *Journal of basic engineering, American Society of Mechanical Engineers*, v. 83, n. 1, p. 95–108, 1961. Citado na página 31.
- KATO, T.; NINOMIYA, Y.; MASAKI, I. An obstacle detection method by fusion of radar and motion stereo. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 3, n. 3, p. 182–188, 2002. Citado na página 34.
- KLASER, R. L. Navegação de veículos autônomos em ambientes externos não estruturados baseada em visão computacional. Dissertação (Mestrado) — Universidade de São Paulo, 2014. Citado 2 vezes nas páginas 36 e 38.
- LANGER, D.; JOCHEM, T. Fusing radar and vision for detecting, classifying and avoiding roadway obstacles. In: IEEE. Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE. [S.I.], 1996. p. 333–338. Citado na página 34.
- LEPETIT, V.; MORENO-NOGUER, F.; FUA, P. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, Springer, v. 81, n. 2, p. 155–166, 2009. Citado 2 vezes nas páginas 28 e 52.
- LINZMEIER, D. T.; SKUTEK, M.; MEKHAIEL, M.; DIETMAYER, K. C. A pedestrian detection system based on thermopile and radar sensor data fusion. In: IEEE. 2005 7th International Conference on Information Fusion. [S.I.], 2005. v. 2, p. 8-pp. Citado 2 vezes nas páginas 20 e 34.
- LU, C.-P.; HAGER, G. D.; MJOLSNESS, E. Fast and globally convergent pose estimation from video images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 22, n. 6, p. 610–622, 2000. Citado na página 28.

MELVIN, W. L.; SCHEER, J. A. Principles of Modern Radar. [S.I.]: SciTech Publishing, 2013. Vol. II: Advanced Techniques. Citado na página 23.

NAWC, N. A. W. C. Electronic Warfare and Radar Systems Engineering Handbook. [S.I.], 1997. Citado 2 vezes nas páginas 24 e 52.

POINTCLOUDS.ORG. Point Cloud Library (PCL) 1.8.0. 2016. Disponível em: <<http://docs.pointclouds.org/trunk/index.html>>. Acesso em: 28 nov. 2016. Citado na página 49.

POINTGREY. Bumblebee2. 2012. Citado na página 63.

_____. Point Grey Bumblebee® XB3. 2014. Citado 2 vezes nas páginas 43 e 47.

QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.; WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE, JAPAN. ICRA workshop on open source software. [S.I.], 2009. v. 3, n. 3.2, p. 5. Citado 2 vezes nas páginas 49 e 71.

RICHARDS, M. A.; HOLM, W. A.; SCHEER, J. A. Principles of Modern Radar. [S.I.]: SciTech Publishing, 2010. Vol. I: Basic Principles. Citado na página 23.

RITTER, H.; ROHLING, H. Pedestrian detection based on automotive radar. In: IET. Radar Systems, 2007 IET International Conference on. [S.I.], 2007. p. 1–4. Citado na página 34.

ROHLING, H.; HEUEL, S.; RITTER, H. Pedestrian detection procedure integrated into an 24 ghz automotive radar. In: IEEE. 2010 IEEE Radar Conference. [S.I.], 2010. p. 1229–1232. Citado na página 34.

RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA). Shanghai, China: [s.n.], 2011. Citado 2 vezes nas páginas 49 e 58.

SHINZATO, P.; WOLF, D.; STILLER, C. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In: Intelligent Vehicles Symposium Proceedings, 2014 IEEE. [S.I.: s.n.], 2014. p. 687–692. Citado 2 vezes nas páginas 36 e 38.

SKOLNIK, M. Introduction to Radar Systems. Third edition. [S.I.]: Tata McGraw-Hills, 2001. Citado 2 vezes nas páginas 24 e 25.

URMSON, C.; ANHALT, J.; BAGNELL, J. A. D.; BAKER, C. R.; BITTNER, R. E.; DOLAN, J. M.; DUGGINS, D.; Ferguson , D.; GALATALI, T.; GEYER, H.; GITTELMAN, M.; HARBAUGH, S.; Hebert , M.; HOWARD, T.; Kelly , A.; KOHANBASH, D.; Likhachev , M.; MILLER, N.; PETERSON, K.; RAJKUMAR, R.; RYBSKI, P.; SALESKY, B.; SCHERER, S.; SEO, Y.-W.; SIMMONS, R.; SINGH, S.; SNIDER, J. M.; Stentz , A. T.; WHITTAKER, W. R. L.; ZIGLAR, J. Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge. <http://archive.darpa.mil/grandchallenge/>, 2007. Citado 2 vezes nas páginas 33 e 34.

UTINO, V.; WOLF, D.; OSORIO, F. Data fusion obtained from multiple images aiming the navigation of autonomous intelligent vehicles in agricultural environment. In: Robotics: SBR-LARS Robotics Symposium and Robocontrol (SBR LARS Robocontrol), 2014 Joint Conference on. [S.I.: s.n.], 2014. p. 157–162. Citado na página 37.

VELODYNE. HDL-32E. 2016. Disponível em: <<http://velodynelidar.com/hdl-32e.html>>. Acesso em: 28 nov. 2016. Citado na página 48.

WANG, X.; XU, L.; SUN, H.; XIN, J.; ZHENG, N. Bionic vision inspired on-road obstacle detection and tracking using radar and visual information. In: Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on. [S.l.: s.n.], 2014. p. 39–44. Citado na página 34.

WHO, W. H. O. Pedestrian safety: a road safety manual for decision-makers and practitioners. World Health Organization, 2013. Citado na página 19.

XIANG, Y.; CHOI, W.; LIN, Y.; SAVARESE, S. Subcategory-aware convolutional neural networks for object proposals and detection. In: . [s.n.], 2016. abs/1604.04693. Disponível em: <<http://arxiv.org/abs/1604.04693>>. Citado na página 34.

ZHANG, Z. A flexible new technique for camera calibration. IEEE Transactions on pattern analysis and machine intelligence, IEEE, v. 22, n. 11, p. 1330–1334, 2000. Citado na página 27.