

---

Detecção e rastreamento de obstáculos em  
ambientes urbanos utilizando visão estéreo

*Daniela Alves Ridel*

---



SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

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

**Daniela Alves Ridel**

## Detecção e rastreamento de obstáculos em ambientes urbanos utilizando visão estéreo

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Denis Fernando Wolf

**USP – São Carlos**  
**Agosto de 2016**

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

R634d      Ridel, Daniela Alves  
            Detecção e rastreamento de obstáculos em  
ambientes urbanos utilizando visão estéreo / Daniela  
Alves Ridel; orientador Denis Fernando Wolf. - São  
Carlos - SP, 2016.  
            88 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, 2016.

            1. Detecção. 2. Rastreamento. 3. Câmeras.  
4. Veículos Inteligentes. I. Wolf, Denis Fernando,  
orient. II. Título.

**Daniela Alves Ridel**

Detection and tracking of obstacles in urban environments  
using stereo vision

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. Denis Fernando Wolf

**USP – São Carlos**  
**August 2016**



*Este trabalho é dedicado aos meus pais, Rosinei da Silva Alves Ridel e João Eduardo Ridel  
e à minha irmã Denise Alves Ridel.*





# AGRADECIMENTOS

---

---

À Deus pelo cuidado comigo e por me mostrar sempre o caminho certo a seguir.

Aos meus pais João Eduardo Ridel e Rosinei da Silva Alves Ridel e toda minha família pela força e apoio para que eu sempre siga rumo aos meus sonhos.

Ao meu orientador Denis Wolf por ter me recebido no Laboratório de Robótica Móvel (LRM) e desde o início ter me dado a oportunidade de expandir meus conhecimentos e fazer parte de atividades tão engrandecedoras.

Ao Patrick Shinzato pelo altruísmo na partilha de conhecimento e por meio de quem eu conheci o LRM e o projeto CaRINA.

A todos os integrantes LRM por manterem um ambiente colaborativo de troca diária de conhecimentos e pelas amizades que vou levar para a vida toda. Alberto, Xicaum, Diego, Kelen, Jeff, Rafael, Thomio, Valéria, Marcos, Andrés. Gostaria de agradecer ao Tiago pela disponibilidade e ajuda com as coletas de ROS *bags files* utilizados neste trabalho, ao Carlos pelas explicações e trocas de ideias sobre vários assuntos e ao Luis por disponibilizar para o grupo as mensagens de visualização de obstáculos e a integração com Rviz.

Aos amigos da pós-graduação que fiz aqui em São Carlos.

Aos amigos dos tempos de graduação na UFMS.

Sou muito grata a todos que participaram dessa jornada comigo.



*“Seu maior sonho não diz respeito apenas a você. Seu sonho é uma parte singular de um sonho muito maior que Deus tem para o mundo. Talvez seja por essa razão que Deus se interesse mais por seus sonhos do que você mesma. Ele a criou não apenas para que você alcance seus sonhos, mas para que enfrente todos os obstáculos em seu caminho e vivencie seus sonhos.”*

*(Você nasceu para isso - Bruce Wilkinson)*



# RESUMO

RIDEL, D. A.. **Detecção e rastreamento de obstáculos em ambientes urbanos utilizando visão estéreo**. 2016. 88 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Segundo relatório disponibilizado pela *World Health Organization* (WHO) (WHO, 2015), 1,3 milhões de pessoas morrem todos os anos no mundo devido à acidentes de trânsito. Veículos inteligentes se mostram como uma proeminente solução para reduzir esse drástico número. Por isso, diversos grupos de pesquisa no mundo têm concentrado esforços para o desenvolvimento de pesquisa que viabilize o desenvolvimento desse tipo de tecnologia.

Diversos são os requisitos necessários para que um veículo possa circular de forma completamente autônoma. Localização, mapeamento, reconhecimento de semáforos e placas de trânsito são apenas alguns dentre tantos. Para que um veículo trafegue nas vias de forma segura, ele precisa saber onde estão os agentes que coabitam o mesmo espaço. Depois que esses agentes são detectados é necessário prever suas movimentações de forma a reduzir os riscos de colisão.

Neste projeto propôs-se a construção de um sistema que visa detectar agentes (obstáculos) e realizar o rastreamento deles para estimar suas velocidades e localizações enquanto estiverem no campo de visão do veículo autônomo, assim possibilitando realizar o cálculo da chance de colisão de cada um desses obstáculos com o veículo autônomo.

O sistema utiliza unicamente a informação provida por uma câmera estereoscópica. Os pontos da cena são agrupados utilizando a informação da 24-vizinhança, disparidade e um valor que corresponde a chance de fazerem parte de um obstáculo. Após o agrupamento, cada grupo é dado como um possível obstáculo, após checar a consistência desses obstáculos por dois frames consecutivos, o grupo, agora considerado um obstáculo passa a ser rastreado utilizando filtro de Kalman (WELCH; BISHOP, 1995) e para checar a correspondência de obstáculos ao longo de toda a sequência é utilizado o algoritmo de Munkres (MUNKRES, 1957).

A detecção e o rastreamento foram avaliados quantitativamente e qualitativamente utilizando dados coletados no Campus II da USP de São Carlos, bem como o conjunto de dados KITTI (GEIGER; LENZ; URTASUN, 2012). Os resultados demonstram a eficiência do algoritmo tanto na detecção dos obstáculos como no rastreamento dos mesmos.

**Palavras-chave:** Detecção, Rastreamento, Câmeras, Veículos Inteligentes.



# ABSTRACT

RIDEL, D. A.. **Detecção e rastreamento de obstáculos em ambientes urbanos utilizando visão estéreo**. 2016. 88 f. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

According to a report provided by the WHO (World Health Organization) in 2015 ([WHO, 2015](#)), 1.3 million people die every year worldwide due to traffic accidents. Intelligent vehicles appear as a prominent solution to reduce this number. Many research groups in the world have been focussing efforts on the development of research in order to enable the development of such technology.

There are several requirements for a vehicle be completely autonomous on the roads. Location, mapping, recognition of traffic lights and traffic signs are just a few among many. For safety the vehicle needs to detect all the other elements that are present in the same environment and to estimate their velocity in order to know where they are planning to go to avoid any kind of collision.

This project proposes a system to detect obstacles and perform their tracking to estimate their speeds and locations enabling the calculation of the chance of collision of each of these obstacles with the autonomous vehicle.

The system only uses the information provided by a stereoscopic camera. The points in the scene are clustered using the 24-neighborhood information, disparity and a value related to the chance of it being part of an obstacle. After the clustering, each cluster is considered a possible obstacle, when the consistence is checked in two frames the cluster becomes an obstacle and starts being tracked using Kalman filter ([WELCH; BISHOP, 1995](#)), to match obstacles being tracked in the whole sequence the Munkres algorithm ([MUNKRES, 1957](#)) is used.

The detection and tracking were evaluated qualitatively and quantitatively using data collected in the Campus II of USP in São Carlos and data from KITTI dataset ([GEIGER; LENZ; URTASUN, 2012](#)). The results show the algorithm's efficiency in obstacle detection and tracking.

**Key-words:** Detection, Tracking, Cameras, Intelligent Vehicles.





# LISTA DE ILUSTRAÇÕES

---

---

Figura 1 – Plataformas de teste do LRM. . . . .	24
Figura 2 – Arquitetura do sistema da plataforma CaRINA 2 . . . . .	37
Figura 3 – Sistema de coordenadas da câmera (x = direita, y = baixo, z = frente) . . . . .	38
Figura 4 – Plataforma CaRINA 2, exemplificação de localização dos sensores . . . . .	38
Figura 5 – Diagrama da proposta. . . . .	40
Figura 6 – Dados das Figuras 6b e 6c foram usados como entrada para o algoritmo de agrupamento (Fig. 6d). . . . .	42
Figura 7 – 24-vizinhança . . . . .	43
Figura 8 – Resultados da detecção para a sequência de imagens do KITTI. A primeira coluna é a imagem da câmera esquerda. A segunda representa cada obstáculo pelo seus conjuntos de pontos. . . . .	51
Figura 9 – Obstáculos muito próximos da câmera não são detectados por causa do parâmetro de disparidade máxima. . . . .	51
Figura 10 – Imagens sem limitação de distância. . . . .	52
Figura 11 – Resultados da detecção para a sequência de imagens do CaRINA. A primeira coluna é a imagem da câmera esquerda. A segunda representa cada obstáculo pelo seus conjuntos de pontos. . . . .	53
Figura 12 – Imagem ilustrando as diferentes sequências utilizadas para avaliação do método. . . . .	56
Figura 13 – Imagens da sequência A. Nessa sequência um ciclista no primeiro momento está à esquerda do veículo autônomo desloca-se até a sua frente, onde permanece até o fim da sequência de imagens. Os pontos coloridos agrupados correspondem à saída da etapa de detecção. O retângulo ao entorno corresponde à saída da etapa de rastreamento . . . . .	57
Figura 14 – Gráficos comparando o resultado da sequência A com o GT . . . . .	58
Figura 15 – Erros na etapa de detecção da sequência A. . . . .	60
Figura 16 – Gráfico comparando o resultado obtido com e sem o uso do filtro de Kalman. . . . .	61
Figura 17 – A detecção e rastreamento são baseadas em um mapa de disparidade esparsa, levando a saltos na etapa de detecção. . . . .	62
Figura 18 – Imagens da sequência B. Nessa sequência duas pessoas estão paradas, uma do lado esquerdo e outra do lado direito da rua. Ambas cruzam na frente do veículo autônomo e atravessam para o lado oposto. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento . . . . .	63

Figura 19 – Gráficos comparando o resultado da sequência B com o <i>Ground Truth</i> . . . .	64
Figura 20 – Imagens da sequência C. Nessa sequência três pessoas atravessam na frente do veículo. Duas pessoas atravessam do lado direito do carro para o lado esquerdo e uma pessoa faz o inverso. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento . . . . .	66
Figura 21 – Gráficos comparando o resultado da sequência C com o <i>Ground Truth</i> . . . .	67
Figura 22 – Imagens da sequência D também com três pessoas. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento . . . . .	68
Figura 23 – Gráficos comparando o resultado da sequência D com o <i>Ground Truth</i> . . . .	70
Figura 24 – Erros na etapa de detecção da sequência D. Nas Figuras 24a, 24b e 24c é exemplificado um caso de erro, onde dois pedestres foram detectados como um único obstáculo. Isso ocorre porque como é possível ver no mapa de custo Figura 24b, a pequena região que separa as duas pessoas teve um valor de custo mais próximo de um obstáculo, do que uma área navegável, portanto no agrupamento os dois obstáculos acabaram sendo agrupados em um mesmo grupo, como pode ser visto na Figura 24c. Já nas Figuras 24d e 24e ocorreu a troca de IDs entre dois tracklets distintos. . . . .	71
Figura 25 – Exemplo da configuração das coordenadas de um objeto detectado em relação ao carro. . . . .	85
Figura 26 – Avaliação dos TTX do carro e do obstáculo na primeira sequência. . . . .	85
Figura 27 – Imagens da primeira sequência, onde um pedestre vem de encontro ao carro autônomo. O retângulo ao redor corresponde à saída da etapa de rastreamento. . . . .	86
Figura 28 – Imagens da segunda sequência, onde um pedestre atravessa do lado esquerdo da calçada para o lado direito. . . . .	87
Figura 29 – Avaliação dos TTX do carro e do obstáculo na segunda sequência. . . . .	88
Figura 30 – Exemplo da estimativa de colisão. . . . .	88

# LISTA DE TABELAS

---

---

Tabela 1 – Resultados quantitativos da avaliação da sequência A. . . . .	59
Tabela 2 – Análise MOTA e MOTP . . . . .	62
Tabela 3 – Resultados quantitativos da avaliação da sequência B. . . . .	63
Tabela 4 – Resultados quantitativos da avaliação da sequência C. . . . .	65
Tabela 5 – Resultados quantitativos da avaliação da sequência D. . . . .	69



# LISTA DE ABREVIATURAS E SIGLAS

---

---

CaRINA ..	Carro Robótico Inteligente para Navegação Autônoma
DBSCAN .	<i>Density-Based Spatial Clustering of Applications with Noise</i>
ELAS ....	<i>Efficient LARge-scale Stereo</i>
F-BE .....	<i>Forward-Backward Error</i>
GP .....	Processo Gaussiano
GT .....	<i>Ground Truth</i>
HOG .....	<i>Histogram of Oriented Gradients</i>
IROS .....	<i>International Conference on Intelligent Robots and Systems</i>
KNN .....	<i>K-Nearest Neighbors</i>
LARS ....	<i>Latin American Robotics Symposium</i>
LIDAR ...	<i>Light Detection and Ranging</i>
LRM .....	Laboratório de Robótica Móvel
MDP .....	<i>Markov Decision Processes</i>
MF .....	<i>Median Flow tracker</i>
ML .....	<i>Mostly Lost</i>
MOTA ...	<i>Multiple Object Tracking Accuracy</i>
MOTP ....	<i>Multiple Object Tracking Precision</i>
MT .....	<i>Mostly Tracked</i>
NCC .....	<i>Normalized Cross Correlation</i>
OpenCV ..	<i>Open Source Computer Vision</i>
PT .....	<i>Partially Tracked</i>
RANSAC .	<i>RANdom SAmple Consensus</i>
ROS .....	<i>Robot Operating System</i>
SIFT .....	<i>Scale-Invariant Feature Transform</i>
SURF ....	<i>Speeded Up Robust Features</i>
SVM .....	<i>Support Vector Machine</i>
TLD .....	<i>Tracking-Learning-Detection</i>
WHO ....	<i>World Health Organization</i>



# SUMÁRIO

---

---

1	INTRODUÇÃO	23
2	TRABALHOS RELACIONADOS	27
2.1	Detecção de obstáculos	28
2.2	Rastreamento de obstáculos	30
2.3	Considerações Finais	31
3	MATERIAIS E MÉTODOS	33
3.1	Câmera Estereoscópica	33
3.2	ROS	35
3.3	OpenCV	36
3.4	Veículo experimental CaRINA 2	36
3.5	Considerações Finais	38
4	DETECÇÃO E RASTREAMENTO	39
4.1	Detecção de Obstáculos	40
4.1.1	<i>Correspondência</i>	41
4.1.2	<i>Seleção de Pontos</i>	41
4.1.3	<i>Computação de custo dos pontos</i>	42
4.1.4	<i>Agrupamento de pontos</i>	43
4.2	Rastreamento	44
4.2.1	<i>Median Flow tracker</i>	44
4.2.1.1	<i>Fluxo óptico</i>	44
4.2.1.2	<i>Forward-Backward Error</i>	45
4.2.2	<i>Filtro de Kalman</i>	45
4.2.3	<i>Algoritmo de Munkres</i>	48
4.3	Considerações Finais	48
5	RESULTADOS	49
5.1	Avaliação da Detecção	49
5.2	Avaliação do Rastreamento	52
5.2.1	<i>Métricas de Avaliação</i>	53
5.2.2	<i>Construção do Ground Truth</i>	55
5.2.3	<i>Avaliação</i>	55

<b>5.2.4</b>	<b><i>Sequência A</i></b> . . . . .	<b>56</b>
5.2.4.1	<i>Análise do Filtro de Kalman</i> . . . . .	61
5.2.4.2	<i>Análise do MOTA e MOTP</i> . . . . .	62
<b>5.2.5</b>	<b><i>Sequência B</i></b> . . . . .	<b>62</b>
<b>5.2.6</b>	<b><i>Sequência C</i></b> . . . . .	<b>65</b>
<b>5.2.7</b>	<b><i>Sequência D</i></b> . . . . .	<b>68</b>
<b>5.3</b>	<b>Considerações Finais</b> . . . . .	<b>69</b>
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>73</b>
<b>REFERÊNCIAS</b> . . . . .		<b>75</b>
<b>APÊNDICE A</b>	<b>PUBLICAÇÕES</b> . . . . .	<b>81</b>
<b>ANEXO A</b>	<b>PREDIÇÃO DE COLISÃO</b> . . . . .	<b>83</b>



---

## INTRODUÇÃO

---

A construção de veículos autônomos poderá ajudar a resolver problemas atualmente enfrentados no sistema de transporte vigente, como a prevenção de acidentes e o melhor uso do fluxo das vias. Essa tecnologia pode também propiciar no futuro o desenvolvimento de sistemas de transporte inteligentes que melhorem o tempo de utilização dos veículos e diminuam a extensa frota automobilística que continua crescendo ano após ano.

As pesquisas em veículos autônomos tiveram uma grande evolução devido à competição promovida pela agência militar norte-americana DARPA (*Defense Advanced Research Projects Agency*), cujo principal propósito era acelerar o desenvolvimento de tecnologias em veículos autônomos visando sua aplicabilidade em questões militares. A primeira edição do DARPA Grand Challenge ocorreu em 2004 na Califórnia/EUA e nenhuma equipe conseguiu completar o desafio com sucesso.

Já na edição seguinte, realizada em 2005, foi proposto o desafio de um veículo autônomo realizar uma travessia de 212 Km em um deserto localizado nos Estados Unidos, 23 times foram selecionados e 5 deles conseguiram completar o desafio, desses, 4 equipes completaram a prova abaixo do tempo limite de 10 horas. A equipe vencedora foi a equipe da Universidade de Stanford com o veículo Stanley ([THRUN \*et al.\*, 2006](#)), completando a prova em 6 horas e 53 minutos.

A última edição da competição ocorreu em 2007 e o desafio era a navegação de veículos autônomos em um ambiente urbano, cerca de 96 Km, no estado da Califórnia, sendo que os veículos deveriam cumprir as regras de trânsito do estado e atuar em um ambiente onde haviam outros carros se locomovendo. O desafio foi vencido pela Universidade Carnegie Mellon com o veículo Boss ([URMSON \*et al.\*, 2009](#)).

Grande parte do conhecimento desenvolvido pela equipe da Universidade de Stanford foi absorvido pela empresa Google, que contratou o coordenador da equipe, bem como parte do time para o desenvolvimento de seu próprio carro autônomo. Até o ano de 2016 os carros da Google já percorreram 2.414.016 quilômetros e estão já nas ruas em Mountain View-CA,

Austin-TX, Kirkland-WA e Metro Phoenix-AZ <sup>1</sup>.

A indústria automotiva também tem empregado esforços para que essa tecnologia chegue cada vez mais perto da realidade do consumidor. Em 2013 a Mercedes Benz percorreu um caminho de 103 km na Alemanha com seu veículo autônomo (ZIEGLER *et al.*, 2014), utilizando radar e câmeras em combinação com mapas digitais precisos (esses mapas informavam a localização de semáforos, número e direção das faixas de trânsito e placas de trânsito). O veículo autônomo desenvolvido pela Audi, apelidado pela equipe de "Jack", viajou cerca de 900 quilômetros de forma parcialmente autônoma (quando o veículo se aproximava de uma cidade, o motorista era requisitado a assumir o volante) da Califórnia até Nevada para a CES 2015 (Consumer Electronics Show) <sup>2</sup>.

O Projeto Carro Robótico Inteligente para Navegação Autônoma (CaRINA) teve início em 2010 com o veículo CaRINA 1, Figura 1a, um carro elétrico que foi automatizado e serviu de plataforma para os trabalhos iniciais do grupo do Laboratório de Robótica Móvel (LRM) do Instituto de Ciências Matemáticas e de Computação (ICMC) da Universidade de São Paulo (USP), no campus de São Carlos. Atualmente o grupo utiliza como plataforma de teste o veículo CaRINA 2, Figura 1b, um Fiat Palio Adventure Dualogic<sup>3</sup>, adquirido em 2012, com diversos resultados práticos em mapeamento, localização e controle testados. Além dos carros, no início de 2016 foi apresentado ao público o caminhão autônomo que foi um projeto em parceria com a Scania, Figura 1c.

Figura 1 – Plataformas de teste do LRM.



Fonte: Elaborada pelo autor.

Diversos são os requisitos necessários para que um veículo autônomo possa circular em ruas de forma segura. Um desses requisitos é a necessidade da percepção do que ocorre à sua volta, outros carros se movimentando, pedestres atravessando a rua, motociclistas. Nesse cenário se encontram os algoritmos de detecção e rastreamento, que permitem ao veículo modelar o comportamento de outros agentes no cenário, e dessa forma realizar ultrapassagens e prevenir

<sup>1</sup> <<https://www.google.com/selfdrivingcar/>>, acessado em Junho de 2016.

<sup>2</sup> <<http://www.audiusa.com/newsroom/news/press-releases/2015/01/550-mile-piloted-drive-from-silicon-valley-to-las-vegas>>, acessado em Janeiro de 2015.

<sup>3</sup> Com esse veículo no ano de 2013 foi realizado o primeiro teste autorizado de um veículo autônomo em vias públicas da América Latina.

qualquer tipo de colisão. A maioria dos algoritmos atuais realiza a detecção de componentes do cenário (pessoas, carros e motociclistas) de forma individual, devido à dificuldade em algoritmos supervisionados de detectar diferentes componentes em um mesmo algoritmo.

No contexto de veículos autônomos, uma das necessidades fundamentais para um veículo circular nas ruas sem um motorista é o reconhecimento de obstáculos à sua volta. Seja para segurança de pedestres atravessando uma rua ou carros se locomovendo no mesmo meio. A partir do instante que esses objetos são detectados é interessante rastreá-los e com isso estimar toda a sua movimentação enquanto permanecerem no campo de atuação do veículo autônomo. Rastrear o deslocamento de obstáculos diminui a possibilidade de colisões, além de permitir ao veículo autônomo fazer ultrapassagens quando necessário. Além disso, depois que a localização, direção e velocidade desses obstáculos são estimadas, é possível prever se existe a chance desses obstáculos colidirem com o veículo autônomo. Caso positivo, o veículo autônomo pode tomar uma ação preventiva.

Este trabalho de mestrado visa auxiliar uma das partes fundamentais do projeto CaRINA criando um sistema de detecção e rastreamento de obstáculos em ambientes urbanos, utilizando unicamente uma câmera estéreo como sensor exteroceptivo. Os objetivos específicos são detectar diferentes tipos de objetos, minimizar o número de suposições sobre o ambiente e não precisar de treinamento prévio.

Para isso foi utilizada uma abordagem baseada em aprendizado não-supervisionado realizando o agrupamento de pontos em obstáculos baseado na 24-vizinhança, no valor de disparidade e na chance de cada ponto fazer parte de um obstáculo. Depois que os pontos são agrupados em diferentes obstáculos, eles são rastreados utilizando Filtro de Kalman, fluxo óptico e algoritmo de Munkres. O resultado esperado é conseguir estimar corretamente a posição e deslocamento de obstáculos em cenários urbanos simples com o intuito de evitar possíveis colisões.

Esta dissertação está organizada da seguinte maneira. No Capítulo 2 são apresentados os trabalhos relacionados. No capítulo 3 são descritos os materiais e métodos utilizados. O projeto de detecção e rastreamento é apresentado no Capítulo 4, explicando o diagrama do algoritmo proposto. Os resultados são mostrados no Capítulo 5, e no Capítulo 6 são apresentadas as conclusões.



---

## TRABALHOS RELACIONADOS

---

Neste capítulo são apresentados os trabalhos relacionados a algoritmos de detecção e rastreamento. Primeiramente são citados alguns trabalhos que englobam os sistemas de detecção e rastreamento de objetos de uma forma mais ampla e nas seções seguintes são discutidas abordagens mais focadas em sensores de visão e em aplicações para veículos inteligentes.

Detecção e rastreamento de obstáculos são alguns dos principais requisitos para uma navegação segura, principalmente em ambientes urbanos. Várias abordagens para resolução desse problema vêm sendo desenvolvidas ao longo dos últimos anos, por meio do uso de sensores ativos como *Light Detection and Ranging* (LIDAR) ((WOJKE; HÄSELICH, 2012) e (HIMMELSBACH; WUENSCH, 2012)) ou radar, sensores passivos como câmeras, ou mesmo a fusão de diferentes tipos desses sensores (PREMEBIDA *et al.*, 2007).

A detecção de obstáculos pode ser feita de uma forma ampla onde os mesmos não passam por alguma espécie de classificação, sendo apenas distinguidos da área navegável. Ou pode ser feita utilizando algoritmos responsáveis por detectar uma classe específica, por exemplo, carros. Nestes, a etapa de reconhecimento é fundamental e a maioria dos algoritmos treinam um classificador capaz de encontrar objetos dessas classes específicas nas imagens.

Em (SIVARAMAN; TRIVEDI, 2013) é realizado um resumo e comparação de métodos de detecção e rastreamento apenas de veículos utilizando câmeras e eles são divididos entre: baseados em características e baseados em movimentação, a maioria dos trabalhos com câmeras monoculares utilizam informações de aparência dos carros, por outro lado a maior parte dos trabalhos com visão estereoscópica utilizam informações de movimentação dos veículos.

A grande maioria dos trabalhos que utilizam câmeras monoculares e usam abordagens baseadas em extração de características, tem utilizado descritores como *Histogram of Oriented Gradients* (HOG) ((SUN; BEBIS; MILLER, 2006), (YUAN *et al.*, 2011), (NIKNEJAD *et al.*, 2012)) e Haar-like features (CHANG; CHO, 2010). Outros trabalhos também usam *Speeded Up Robust Features* (SURF) (LIN *et al.*, 2012), *Scale-Invariant Feature Transform* (SIFT) (LOWE,

2004), Gabor (SUN; BEBIS; MILLER, 2006) além da combinação destes. Em conjunto com esses descritores, os classificadores mais utilizados são *Support Vector Machine* (SVM) ((SUN; BEBIS; MILLER, 2006) e (YUAN *et al.*, 2011)) e Adaboost ((SIVARAMAN; TRIVEDI, 2010)). Os trabalhos que utilizam câmera monocular e são baseados em movimentação em geral utilizam técnicas de fluxo óptico (FRANKE *et al.*, 2005) e (DANESCU; ONIGA; NEDEVSCHI, 2011).

Nos trabalhos que utilizam visão estereoscópica ocorre o inverso. As abordagens baseadas em movimentação são mais comuns do que as baseadas na aparência. A maioria dos estudos utiliza o rastreamento de pontos em 3D e distinção entre objetos estáticos e em movimento. Em ambas as abordagens os filtros mais comumente utilizados para a fase de rastreamento são filtros de Kalman (FRANKE *et al.*, 2005) e filtros de partículas (DANESCU; ONIGA; NEDEVSCHI, 2011). Uma vantagem do uso de sistemas estereoscópicos é a possibilidade tanto do uso de informação 3D como a informação das intensidades dos *pixels* da imagem de uma das câmeras.

Câmeras são sensíveis à luz, condições climáticas e exigem uma computação mais robusta para processar todas as informações captadas do meio (SIVARAMAN; TRIVEDI, 2013), no entanto, as imagens providenciam uma fonte de dados densa e rica, de onde podem ser retiradas informações de contexto que podem ajudar na tarefa de rastreamento (do inglês *tracking*). Além disso, em geral, câmeras costumam ter menor custo do que sensores ativos.

Na seção 2.1 são apresentadas abordagens que utilizam visão estereoscópica para detecção de obstáculos sem classificação, ou seja, detectando qualquer tipo de obstáculo sem assignar uma classe específica. Na seção 2.2 são descritos os métodos mais recentes utilizados pela comunidade de Veículos Inteligentes para realizar o rastreamento de obstáculos.

## 2.1 Detecção de obstáculos

Em (BERNINI *et al.*, 2014) é feito um resumo e comparação das principais técnicas de detecção de obstáculos utilizando visão estereoscópica com foco em aplicações reais para veículos inteligentes em abordagens não-supervisionados. Os métodos são divididos em quatro categorias: mapa de ocupação probabilístico, mapa de elevação digital, fluxo óptico e agrupamento baseado em geometria.

Um método baseado em mapa de elevação digital é proposto em (BICHSEL; BORGES, 2015), onde primeiramente é feita uma estimacão do chão. Em seguida é criado um mapa de ocupação bidimensional baseado na estimacão de um Modelo de Elevacão Digital, onde o componente vertical representa a altura na direçao  $y$ , os espaços ocupados no mapa são agrupados e rastreados ao longo do tempo e uma confiançã é gerada para cada um deles. Agrupamentos são descartados usando uma abordagem Bayesiana,

Em (LENZ *et al.*, 2011) a abordagem de segmentaçao entre *pixels* estáticos e dinâmicos é explorada depois da construçao de um novo método de odometria visual (KITZ; GEIGER;

LATEGAHN, 2010). Nesse método é estimada a movimentação do veículo com base em sequências de pares estereoscópicos ao longo do tempo. As características de canto são correlacionadas entre pares de imagens em tempos consecutivos e estimada a movimentação do veículo por meio da relação entre características que foram correspondidas utilizando três imagens da mesma cena estática. Em seguida é realizado o cálculo das matrizes de rotação e translação entre as características correspondidas. Descobrendo-se a movimentação da câmera por meio dessas correspondências é possível distinguir *pixels* que obtiveram uma movimentação de acordo com o padrão de movimento da câmera, (*pixels* estáticos), e *pixels* que se movimentaram fora desse padrão, (*pixels* dinâmicos).

A abordagem de (GUIZILINI; RAMOS, 2013) utiliza imagens de uma câmera monocular, extratores de características SIFT (LOWE, 2004) e características de cantos (SHI; TOMASI, 1994) e realiza a correspondência de características entre imagens em tempos consecutivos utilizando fluxo óptico. Nele é feita a suposição que a maior parte dessas características correspondem a *pixels* estáticos da imagem. Assim é calculada a matriz fundamental que descreve a movimentação da câmera, ou seja, a movimentação do veículo. Desse modo, todos aqueles *pixels* que não correspondem à movimentação descrita pela câmera são considerados *pixels* dinâmicos. Essa separação é realizada utilizando o algoritmo *RANdom SAmples Consensus* (RANSAC) (FISCHLER; BOLLES, 1981) e Processo Gaussiano (GP) (RASMUSSEN, 2006). A desvantagem desse método está relacionada ao seu tempo de computação que inviabiliza a aplicação em tempo real.

A abordagem de (BEWLEY *et al.*, 2014) estende a abordagem proposta em (GUIZILINI; RAMOS, 2013) para múltiplos obstáculos independentes. E ao invés de utilizar o Processo Gaussiano é utilizado o algoritmo *K-Nearest Neighbors* (KNN), diminuindo assim o tempo de computação necessário.

A abordagem de (BADINO; FRANKE; PFEIFFER, 2009) constrói uma representação da cena por meio de barras retangulares adjacentes denominadas "*stixels*", que levam em consideração o fato de o espaço livre ser limitado na maioria das vezes por objetos com superfícies verticais. A cena pode ser representada utilizando um número consideravelmente inferior de informações utilizando  $\eta$  barras verticais ( $\eta = \frac{\text{largura da imagem}}{\text{largura das barras}}$ ). Cada *stixel* é definido por uma posição 3D relativa à câmera, possuindo uma certa largura e altura. Para encontrar os obstáculos é construída uma grade de ocupação onde os eixos dessa grade correspondem aos valores da coluna da imagem e o valor da disparidade calculado pelo algoritmo *Semi-Global Matching* (HIRSCHMULLER, 2005).

A partir dessa grade uma busca começa da parte de baixo da imagem na direção vertical até o ponto onde uma célula ocupada é encontrada. Pode-se considerar então que o espaço a frente dessa célula é considerado não ocupado, ou livre. A rua é estimada ajustando-se uma superfície B-Spline (WEDEL *et al.*, 2008) nos dados. Comumente pode ocorrer de uma coluna possuir mais de uma célula ocupada. Para lidar com esse problema, uma etapa de subtração de



*background* é realizada. Depois a delimitação da altura dos obstáculos é obtida pela segmentação ótima entre as disparidades do obstáculo e o *background* da imagem. Com o espaço livre e a altura para cada coluna computados, para definir as barras basta fundir as alturas computadas de acordo com a largura predefinida. O passo final calcula a média de disparidade de cada uma dessas barras verticais que descreverão o ambiente.

Outros métodos que utilizam a mesma estrutura podem ser encontrados na literatura (PFEIFFER; FRANKE, 2010), (ERBS *et al.*, 2014) e (PFEIFFER; FRANKE, 2010). Em (PFEIFFER; FRANKE, 2011) é realizada uma abordagem multi-camadas. A maioria desses métodos necessita de um mapa de disparidade denso. A eficiência do método está muito ligada à qualidade do mapa de disparidade utilizado. Em (BENENSON *et al.*, 2012) não existe essa necessidade. No entanto, são feitas suposições sobre planaridade do ambiente.

A maioria das técnicas de detecção de obstáculos são dependentes de mapas de disparidade densos. A abordagem proposta em (KRAMM; BENSRAHAI, 2012) utiliza mapas de disparidade esparsos e propõe uma técnica de detecção que consegue utilizar v-disparidade mesmo em ambientes complexos, ocasiões onde técnicas de v-disparidade falham (KRAMM; BENSRAHAI, 2012). Técnicas que utilizam v-disparidade mapeiam pontos  $(u, v, d)$ , sendo  $(u, v)$  as coordenadas da imagem e  $d$  o valor de disparidade para um espaço  $(d, v)$ . Nesse novo espaço, o plano do chão aparecerá como uma diagonal e os obstáculos aparecerão como linhas verticais, facilitando a distinção entre o chão e obstáculos. Em seguida, são construídos histogramas de v-disparidade para identificar valores de profundidade relevantes na cena. Para cada um desses valores relevantes é extraído um conjunto de pontos que são agrupados utilizando o algoritmo de agrupamento por densidade *Density-Based Spatial Clustering of Applications with Noise* (DBSCAN), (ESTER *et al.*, 1996). Esse algoritmo não consegue modelar muito bem os objetos detectados tendo problemas no reconhecimento do formato dos obstáculos.

Segundo (PINGGERA; FRANKE; MESTER, 2015), um dos principais problemas dos métodos baseados em Stixels (BADINO; FRANKE; PFEIFFER, 2009), (PFEIFFER; FRANKE, 2011) e Mapas de Elevação Digital (ONIGA; NEDEVSCHI, 2010) é que eles dependem de mapas de disparidade pré-computados.

## 2.2 Rastreamento de obstáculos

Em (CHOI, 2015) é proposto um algoritmo de rastreamento de vários alvos que integra modelos de movimentação e aparência de longo termo. Assim como utiliza um novo descritor de fluxo local agregado que codifica o padrão de movimentação relativo entre um par de detecções (as detecções são representadas por caixas delimitadoras, do inglês *Bounding Box*). O descritor codifica como os pontos de trajetória de interesse dentro de uma detecção se movem com respeito a outra detecção.

Em (LENZ; GEIGER; URTASUN, 2015) é proposta a utilização de rastreamento por



meio de detecção (do inglês *tracking-by-detection*) e é formulado como um problema de custo mínimo usando o algoritmo de menor caminho com aplicações em carros autônomos e cenários reais.

No trabalho de (XIANG; ALAHI; SAVARESE, 2015) o rastreamento de múltiplos objetos é formulado como um problema de decisão utilizando Processos de Decisão Markoviana (do inglês *Markov Decision Processes* (MDP)), onde o tempo de vida de um objeto é modelado com um MDP e vários MDPs são agregados para um rastreamento de multi-objetos.

A abordagem de rastreamento de veículos descrita em (OHN-BAR; SIVARAMAN; TRIVEDI, 2013) realiza a detecção de veículos utilizando uma única câmera e técnicas de extração de características de veículos. Na sequência, o fluxo óptico é calculado para achar padrões de movimento que se diferem da movimentação do veículo autônomo. Em seguida, esse fluxo é agrupado tanto no fluxo óptico como no mapa de profundidades para produzir as detecções de veículos que são projetadas no mundo 3D e rastreadas utilizando filtro de Kalman.

## 2.3 Considerações Finais

Neste Capítulo foram apresentados diversos métodos que realizam a detecção e rastreamento tanto de obstáculos específicos (e.g. pessoas, carros) como trabalhos que detectam obstáculos de forma genérica. A grande maioria deles com aplicação em veículos autônomos. Vários dos trabalhos que utilizam visão estereoscópica necessitam de mapas de disparidade densos. E os trabalhos que se baseiam em características normalmente necessitam de uma etapa prévia de treinamento. Diferentemente das abordagens citadas anteriormente, o método proposto neste trabalho não requer a utilização de mapas densos, detecta obstáculos de forma genérica e não precisa de nenhuma etapa de aprendizado ou suposições sobre o ambiente.



---

## MATERIAIS E MÉTODOS

---

Neste capítulo são apresentados os materiais e métodos utilizados ao longo do desenvolvimento desse projeto. A seção 3.1 descreve o sensor utilizado para capturar os dados. As seções 3.2 e 3.3 apresentam, respectivamente, as bibliotecas ROS e OpenCV, que foram amplamente utilizadas no desenvolvimento desse trabalho. Por fim, a seção 3.4 apresenta a plataforma robótica CaRINA 2, que foi utilizada na coleta dos dados experimentais utilizados na avaliação dos algoritmos desenvolvidos.

### 3.1 Câmera Estereoscópica

Um sistema estéreo é uma forma bastante conhecida de modelar o sistema de visão que estima profundidade. Esse tipo de sistema é composto por duas câmeras, e a partir delas duas imagens são obtidas. O processo de estimação da profundidade da cena pode ser dividido em quatro etapas, segundo (BRADSKI; KAEHLER, 2008):

1. Remover matematicamente distorções da lente;
2. Retificação do par estéreo. Ajustar os ângulos e distâncias entre câmeras;
3. Correspondência entre as imagens. Encontrar as mesmas características nas imagens esquerda e direita;
4. Reprojeção dos pontos. Se sabemos o arranjo geométrico das câmeras, podemos tornar o mapa de disparidade em distâncias por triangulação.

Primeiramente, é necessário realizar a calibração das câmeras, pois câmeras possuem distorções internas e muitas vezes as lentes distorcem a localização de *pixels* próximos às bordas da imagem. Se o par de câmeras não for calibrado, dificilmente irá existir uma restrição para estabelecer o emparelhamento dos pontos das imagens esquerda e direita, sendo assim necessária

a realização de uma calibração prévia das câmeras por meio de software. O gabarito de calibração *chessboard* é comumente conhecido e muito utilizado em calibração de câmeras estéreo devido ao fato de seus cantos serem de fácil localização. Com esse gabarito posicionado em frente às duas câmeras, uma série de fotos são gravadas e são identificados os cantos de todos os quadrados do tabuleiro, de modo que sabendo-se o posicionamento desses pontos em ambas as imagens é possível descobrir a diferença de alinhamento entre os *pixels*. Com base nessas coordenadas e como resultado do processo de calibração são obtidos o modelo de geometria da câmera e o modelo de distorção das lentes que definem os parâmetros intrínsecos que são usados para corrigir a distorção das lentes. Segundo (RUSSELL; NORVIG, 2004) "A ideia da visão estereoscópica: As diferentes posições da câmera resultam em visões bidimensionais ligeiramente distintas da mesma cena tridimensional".

Depois que os parâmetros de correção são aplicados às câmeras é possível mensurar a profundidade da cena na qual a câmera está presente. Isto somente é possível devido à paralaxe de visualização, ou seja, a disparidade entre os *pixels* das duas imagens alinhadas.

A saída da terceira etapa é o mapa de disparidade. Dado um *pixel* que descreve alguma característica da cena na imagem da esquerda, é possível encontrar o *pixel* respectivo na imagem direita. Esse passo é conhecido como correspondência. Ao realizar o alinhamento das duas imagens e mensurar a distância entre os pontos respectivos, é obtida a disparidade entre os dois pontos de vista. As medidas de distância e disparidade são inversamente proporcionais, ou seja, quanto menor a distância, maior a disparidade entre os *pixels*. Com base nesse princípio, um mapa de disparidade é construído para cada frame capturado pelo par de câmeras, de modo que cada mapa demonstra em escala de cinza o nível de proximidade dos objetos na cena, onde objetos mais próximos geralmente são representados pela cor branca e os mais afastados pela cor preta. Objetos próximos ao observador apresentam maior disparidade, enquanto objetos distantes apresentam menor disparidade.

Apesar da construção do mapa de disparidade ser um problema bastante estudado e de fácil compreensão, existem diversas dificuldades que o permeiam como oclusão parcial de objetos, distorções, presença de ruídos e variações de iluminação (RZIZA *et al.*, 2000), que fazem com que ele ainda seja um problema em aberto.

Vários algoritmos que realizam a etapa de correspondência podem ser encontrados na literatura (MüHLMANN *et al.*, 2002), (MATTHIES; KANADE; SZELISKI, 1989), (KONOLIGE, 1998), (HUANG; ZHUANG, 1995) e (GEIGER; ROSER; URTASUN, 2010). O algoritmo de correspondência utilizado no processo de construção do mapa de disparidade foi o *Efficient LArge-scale Stereo* (ELAS) (GEIGER; ROSER; URTASUN, 2010). A escolha desse método se dá pelo fato dele realizar a correspondência de forma rápida em imagens de alta resolução. Além disso, ele lida bem com a presença de superfícies inclinadas e com pouca textura, ocasiões onde a maioria dos outros algoritmos falha. Esse método utiliza um conjunto de pontos de suporte que formam correspondências mais robustas.

Pontos de suporte são *pixels* que podem ser robustamente correspondidos devido à sua textura e singularidade. A correspondência de pontos de suporte é realizada em uma grade regular usando a distância  $l_1$  entre vetores formados pela concatenação dos filtros de Sobel horizontal e vertical.

As correspondências são consideradas somente se elas podem ser correspondidas da esquerda para a direita e da direita para a esquerda. Para se livrar de correspondências ambíguas, são eliminados todos os pontos cuja taxa entre a melhor e a segunda melhor correspondência excedem um limiar fixo de  $\tau = 0,9$ . Correspondências falsas são removidas eliminando todos os pontos cujo valor de disparidade não é igual aos pontos de suporte que os rodeiam. Para cobrir a imagem inteira, pontos de suporte adicionais são criados nos cantos da imagem e as disparidades são estabelecidas igual à de seus vizinhos mais próximos.

A partir da informação de disparidade é possível recuperar a informação 3D por uma etapa de reconstrução.

$$Z = \frac{fT}{d}, \quad (3.1)$$

onde;  $Z$  é o valor de profundidade de um *pixel* cuja disparidade é  $d$ ;  $f$  é a distância focal da câmera; e  $T$  é a distância (*baseline*) entre as duas câmeras. Segundo (BRADSKI; KAEHLER, 2008) com um par de câmeras perfeitamente alinhado e sabendo-se a correspondência, a profundidade em  $Z$  pode ser calculada por similaridade de triângulos.

## 3.2 ROS

O *Robot Operating System* (ROS) é um *framework* que foi construído para encorajar o desenvolvimento de softwares relacionados a robótica de uma maneira colaborativa no mundo todo<sup>1</sup>. O envio e recebimento de diferentes tipos de informações são padronizados, fornecendo uma troca de mensagens mais transparente entre os diversos processos. Além disso, ele provê uma série de ferramentas e bibliotecas que facilitam o compartilhamento e teste de soluções de vários grupos de pesquisa.

Neste trabalho, o ROS se mostrou uma ferramenta fundamental, pois, por existirem módulos separados, é de suma importância que exista uma comunicação entre eles para a troca de informações. Além disso, os dados coletados já estão padronizados com o tipo de mensagens disponibilizadas no framework, tanto para o transporte de imagens como outras informações do veículo. O ROS possui um pacote chamado CvBridge que faz a conversão entre mensagens de imagens do ROS para imagens OpenCV. Os visualizadores também auxiliam bastante para avaliação qualitativa dos dados.

---

<sup>1</sup> <[www.ros.org](http://www.ros.org)>

Não só neste trabalho mas no projeto CaRINA como um todo, a ferramenta ROS tem sido amplamente utilizada. Vários são os algoritmos necessários para fazer um veículo navegar de forma autônoma. O ROS torna mais simples a integração de vários códigos por conectar diferentes módulos por meio de mensagens e uma interface onde os módulos assinam e publicam mensagens. Assim, diferentes módulos podem assinar a mesma mensagem publicada em outro módulo, processar essas informações e publicar uma nova mensagem com alguma nova informação.

### 3.3 OpenCV

Visão Computacional é a área responsável pela forma como um computador extrai informações do meio a sua volta, com base nas imagens capturadas por câmeras de vídeo. A biblioteca *Open Source Computer Vision* (OpenCV)<sup>2</sup> (BRADSKI; KAEHLER, 2008) providencia uma grande gama de funções que podem ser usadas para extração de características, transformações, segmentação e rastreamento em imagens, calibração de câmeras, além de alguns algoritmos de aprendizado de máquina.

A implementação mais conhecida de um sistema estéreo de código aberto está disponível nessa biblioteca. Ela agrega vários métodos necessários para realização do mapeamento tridimensional e providencia uma infraestrutura em visão computacional com grande foco em aplicações de tempo real. Além disso, possui várias funções básicas para manipulação de imagens como leitura, escrita e conversão entre sistemas de cores.

Apesar da biblioteca OpenCV prover várias funções para realizar o mapeamento tridimensional, as funções utilizadas para este fim neste trabalho provieram de pacotes disponibilizados no *framework* ROS. No entanto, algumas funções providas por esta biblioteca foram utilizadas. Desde funções mais simples para conversão de imagens em BGR para escala em cinza, até funções mais complexas como cálculo de fluxo óptico.

### 3.4 Veículo experimental CaRINA 2

A plataforma CaRINA 2 é um Fiat Palio Adventure Dualogic que foi modificado possibilitando o controle completo de suas funções (aceleração, esterçamento e freio) por meio de um computador. O computador acoplado ao veículo possui um processador Intel(R) Core(TM) i7-3770 CPU @ 3.40GHz e executa o sistema operacional Linux Ubuntu. Todos os softwares utilizados no carro são baseados na plataforma ROS. Na Figura 2 é possível ver a arquitetura do sistema do CaRINA. Os principais módulos são localização, controle e percepção. É possível notar na figura que todos os módulos são interconectados, vindo daí a necessidade do uso de

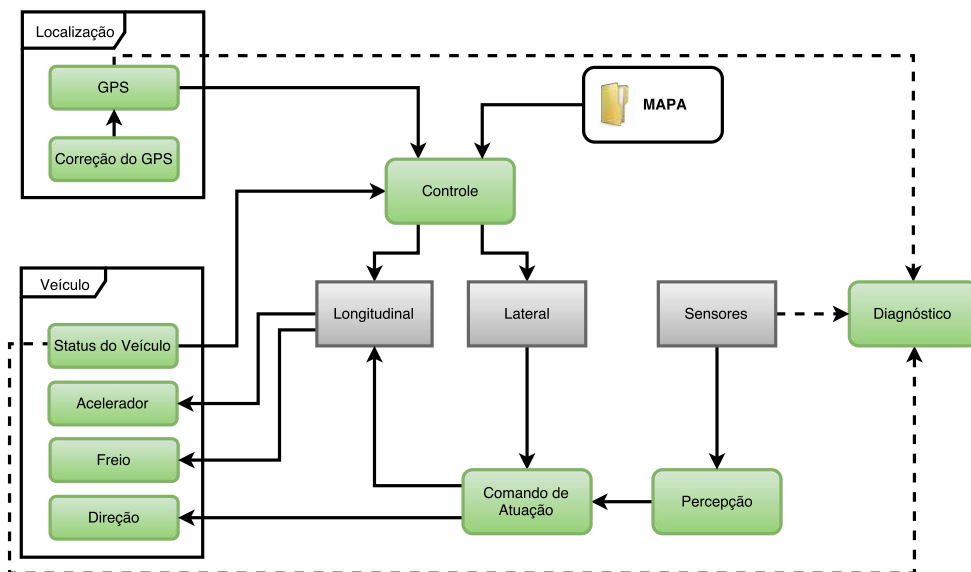
---

<sup>2</sup> <[www.opencv.org](http://www.opencv.org)>

um framework (ROS) que consiga lidar com todas essas ligações entre os módulos. A grande maioria dos algoritmos que são executados no carro são implementados em C++ e Python.

Atualmente, a prevenção de colisão com obstáculos utilizada na plataforma é feita utilizando o sensor Velodyne. Essa prevenção está contida no módulo de Percepção, que manda o comando de atuação para o carro frear caso detecte algum obstáculo. A abordagem proposta neste trabalho visa fornecer um sistema baseado em visão para atuar nesse módulo de percepção substituindo o sensor Velodyne.

Figura 2 – Arquitetura do sistema da plataforma CaRINA 2

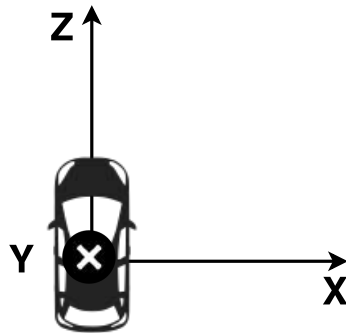


Fonte: Santos (2015).

As configurações de sensores no carro mudam dependendo do experimento a ser realizado. Na maioria dos experimentos tem sido utilizado um sensor LIDAR Velodyne HDL-32E 10 Hz que possui 32 feixes com 360 graus de campo de visão horizontal e 20 graus de campo de visão vertical, 1 câmera estéreo PointGrey Bumblebee XB3 Firewire 16 fps, capaz de adquirir imagens de  $1280 \times 960$  com 3.8 mm de comprimento focal (66 graus de campo de visão horizontal), 1 GPS Septentrio AsteRx2eH operando a 10 Hz com correção RTK e duas antenas, 1 IMU Xsens MTi e 1 radar MMW 20 Hz com  $\pm 45$  graus de campo de visão, podendo detectar até 60 m. Apesar dos vários tipos de sensores disponíveis, a abordagem deste trabalho foca somente em sensores visuais, principalmente porque as frequências da grande maioria dos sensores são diferentes, então um etapa de sincronização dessas informações seria necessária antes de usar os dados de forma conjunta.

Para realizar a calibração das câmeras foi utilizado o pacote `camera_calibration`, disponível no ROS. As câmeras foram calibradas com gabaritos de  $8 \times 6$ , com quadrados de 10cm. Para exemplificação, o sistema de coordenadas da câmera pode ser visto na Figura 3. A câmera está localizada no topo do veículo autônomo logo abaixo do sensor Velodyne 3D, conforme ilustrado na Figura 4.

Figura 3 – Sistema de coordenadas da câmera (x = direita, y = baixo, z = frente)



Fonte: Elaborada pelo autor.

Figura 4 – Plataforma CaRINA 2, exemplificação de localização dos sensores



Fonte: [Fernandes et al. \(2014\)](#).

### 3.5 Considerações Finais

Esse capítulo descreve a plataforma e as ferramentas utilizadas no desenvolvimento deste trabalho. Todos os códigos foram desenvolvidos no framework ROS, principalmente utilizando as trocas de mensagens e os programas de visualização de dados. O uso do ROS facilita a utilização por parte do grupo dos dados resultantes em cada uma das etapas desse trabalho e também facilita o trabalho de integração do código no carro. Os dados para os experimentos foram coletados através de uma câmera PointGrey Bumblebee XB3, utilizando o *baseline* estendido, acoplada em uma estrutura no topo da plataforma CaRINA 2. Para manipular as imagens foi utilizada principalmente a biblioteca OpenCV.



---

## DETECÇÃO E RASTREAMENTO

---

Neste capítulo é apresentada a proposta do trabalho. A proposta é utilizar uma câmera estereoscópica e construir um sistema de detecção e rastreamento que possibilite ao veículo autônomo reagir em situações de perigo e prever essas situações com o maior tempo de antecedência possível, como por exemplo, prever que um pedestre caminhando na calçada em direção à rua pretende atravessá-la, e ao mesmo tempo saber que um pedestre que está saindo da rua e subindo para a calçada não deve apresentar uma situação de risco para o veículo autônomo.

O sistema pode ser dividido em três etapas. A primeira e a segunda correspondem à detecção dos obstáculos e ao rastreamento deles enquanto estiverem no campo de visão do veículo autônomo. A terceira etapa deve usar essas informações para prever se haverá uma colisão. A Figura 5 exemplifica cada uma dessas etapas, onde cada módulo é representado por caixas retangulares na cor cinza.

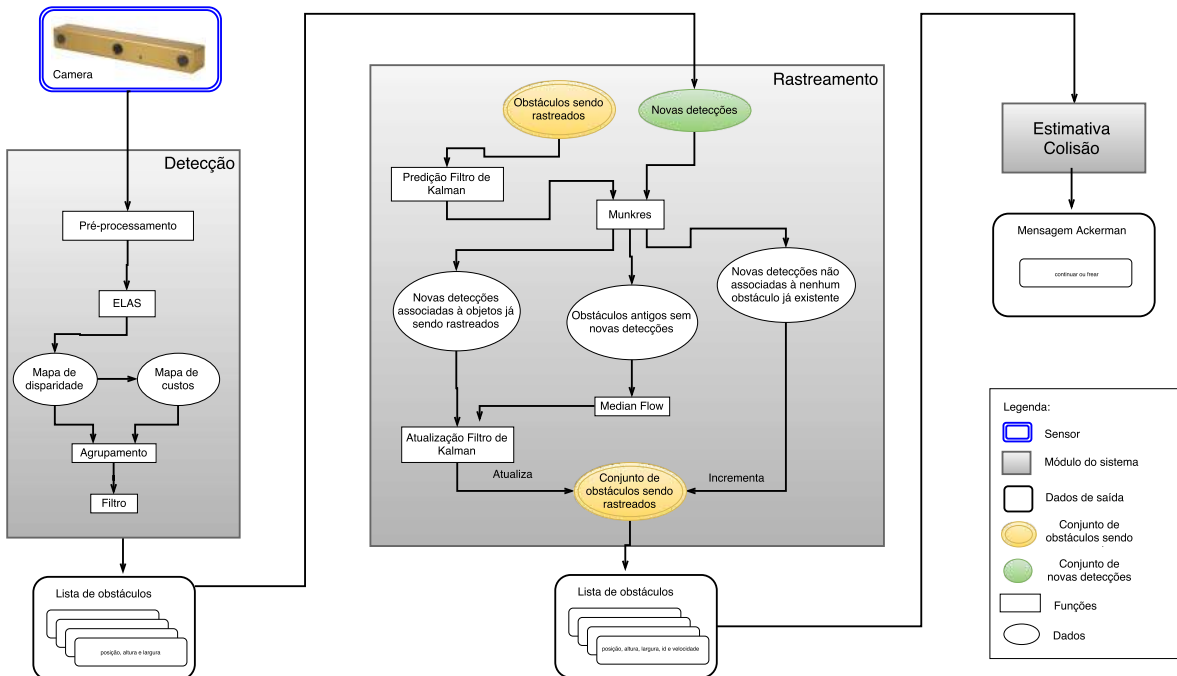
A entrada para a fase de detecção são as imagens de uma câmera estereoscópica que já foi previamente calibrada. A fase de detecção então recebe essas imagens, processa-as (conforme a seção 4.1) e retorna os obstáculos detectados, ou seja, a localização  $z$  e  $x$  do obstáculo em relação às câmeras e a altura e largura dos obstáculos.

A etapa de rastreamento (seção 4.2) recebe as informações da etapa de detecção como entrada e a partir do instante que um obstáculo é detectado e associado por dois frames consecutivos é instanciado um filtro de Kalman (seção 4.2.2) para ele. A checagem dessa consistência se torna necessária para eliminar ruídos detectados como obstáculos esporadicamente nas imagens.

A cada novo instante de tempo é realizada a predição do filtro de Kalman e em seguida a associação de obstáculos que já estão sendo rastreados com novas detecções, para isso é utilizando o algoritmo de Munkres (MUNKRES, 1957) (também conhecido como algoritmo húngaro). Esse é um algoritmo bastante utilizado para resolver o problema de associação, fazendo a melhor associação possível, de forma a obter o menor custo. O custo utilizado nessa associação é a distância euclidiana no plano  $Z \times X$  (plano paralelo ao chão) dos obstáculos, os

valores  $Z$  e  $X$  dos novos obstáculos são passados para realizar a etapa de correção do filtro de Kalman para os objetos que foram associados a uma nova detecção.

Figura 5 – Diagrama da proposta.



Fonte: Elaborada pelo autor.

Já os obstáculos que não foram associados a nenhuma nova detecção têm seu tempo de vida decrescido e sua nova posição  $Z$  e  $X$  é estimada utilizando o algoritmo de *Median Flow tracker* (MF) (KALAL; MIKOLAJCZYK; MATAS, 2010), que estima o deslocamento de uma janela de *pixels* na imagem. A partir desse deslocamento é calculada a mediana de  $Z$  e  $X$  dos pontos mais centrais dessa janela. A partir dessa estimativa os valores  $X$  e  $Z$  são utilizados na etapa de correção do filtro de Kalman. Em seguida, todos os obstáculos que estão sendo rastreados passam por uma checagem e são eliminados caso tenham tempo de vida igual a zero, ou estejam saindo da imagem. Objetos que não foram associados a nenhum obstáculo já sendo rastreado passam a ser novos obstáculos a serem rastreados somente se não possuírem uma área de intersecção no plano da imagem com nenhum outro obstáculo.

As informações de cada obstáculo, bem como as informações de velocidade do veículo são passadas para a etapa de predição de colisão (Anexo A) que informará se existe alguma possibilidade de colisão. A saída dessa etapa é passada para o veículo que deve executar alguma ação preventiva, caso constatada a chance de colisão.

## 4.1 Detecção de Obstáculos

A abordagem de detecção de obstáculos aqui apresentada não faz distinção entre classes de obstáculos. Ao contrário, os obstáculos aqui são detectados de uma forma genérica, onde a

única diferença se dá entre área navegável e obstáculo. Nos parágrafos seguintes estão descritos os processos utilizados para a diferenciação entre área navegável e não navegável, e o posterior agrupamento desses pontos que fazem parte de áreas não-navegáveis em obstáculos.

Uma amostragem de *pixels* é retirada de um mapa de disparidade e é calculado um custo para cada *pixel*, de acordo com (SHINZATO; GOMES; WOLF, 2014) e (SHINZATO, 2015). Em seguida esses *pixels* são agrupados baseado nesse custo e em sua 24-vizinhança, de forma que cada grupo de *pixels* corresponda à um obstáculo.

O sistema segue um processo de quatro etapas. A primeira (4.1.1) consiste em computar o mapa de disparidade. Na segunda (4.1.2), é realizada uma amostragem dos *pixels* para construir um grafo de nós  $p(x, y, z, u, v, d)$ , onde  $x, y, z$  são as coordenadas 3D obtidas por um algoritmo de reconstrução,  $u, v$  são coordenadas na imagem e  $d$  é o valor de disparidade. Na terceira etapa (4.1.3), um custo que representa a chance de um *pixel* fazer parte de um obstáculo é computado. A última etapa (4.1.4) corresponde ao agrupamento, onde pontos são agrupados e cada grupo representará um obstáculo. A Figura 6 mostra os resultados intermediários desse processo e detalhes de como computar os valores de custo estão disponíveis em (SHINZATO, 2015).

### 4.1.1 Correspondência

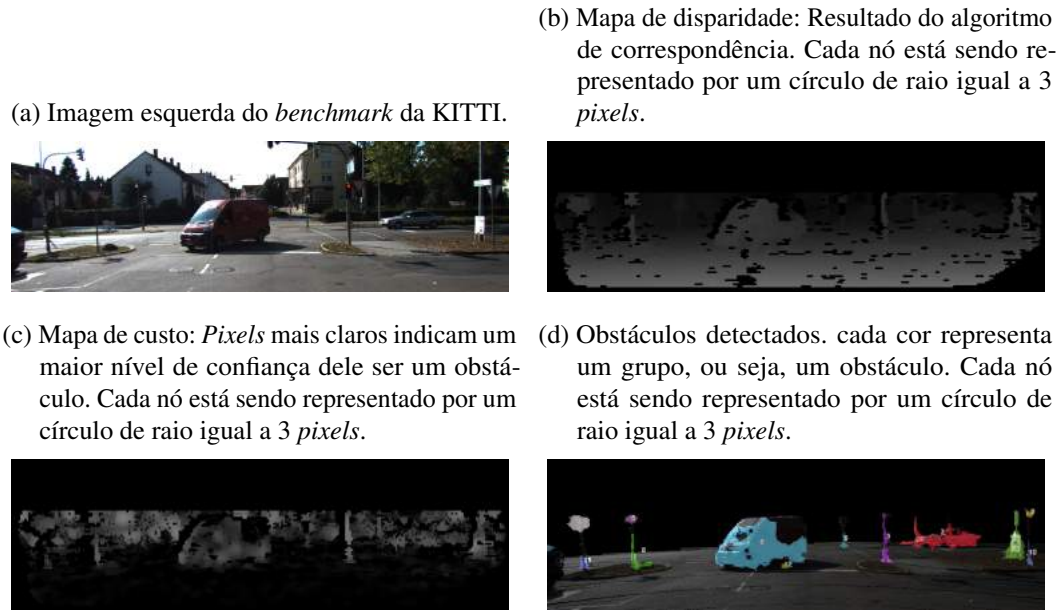
Antes de realizar a correspondência entre as imagens (esquerda e direita), é aplicado a elas uma equalização de histograma para suavizar o ruído causado por sombras. Somente a disparidade das linhas com espaçamento igual a  $\lambda$  é computada para aumento na velocidade de computação. O mapa de disparidade foi construído utilizando uma versão modificada do ELAS (GEIGER; URTASUN, 2010), que não utiliza os filtros de pós processamento. O mapa de disparidade resultante (Fig. 6b) foi então usado na etapa de seleção de pontos para a construção de um grafo. Outros métodos que calculam mapas de disparidade também podem ser usados.

### 4.1.2 Seleção de Pontos

Um grafo não-direcionado  $G = \{P, E\}$  é criado nessa etapa.  $P$  é o conjunto de nós que representam pontos esparsos  $(x, y, z, u, v, d)$  e  $E$  é o conjunto de todas as arestas que garantem que não existe nenhum outro ponto entre dois pontos conectados por uma aresta. Diferentemente de trabalhos anteriores, as arestas de  $G$  não são geradas pelo método de triangulação planar de Delaunay (LEE; SCHACHTER, 1980), mas de acordo com a 24-vizinhança válida de cada *pixel* (Fig. 7). Essa modificação possibilita ao sistema responder a maiores taxas de quadros por causa do tempo de acesso constante.

$P$  deve conter pontos igualmente distribuídos na imagem (distância para qualquer vizinho maior ou igual a  $\lambda$ ) com um valor de disparidade válido (maior que zero) para a extração de um conjunto esparsos de pontos.

Figura 6 – Dados das Figuras 6b e 6c foram usados como entrada para o algoritmo de agrupamento (Fig. 6d).



Fonte: Elaborada pelo autor.

### 4.1.3 Computação de custo dos pontos

Nessa etapa, um valor de custo  $C$  é calculado para cada nó  $p$  do grafo  $G$ , onde  $0.0 < C \leq 1.0$  e representa um nível de confiança do nó  $p$  ser um obstáculo. Usando o grafo  $G$  da etapa anterior, a abordagem calcula  $C$  pela Equação 4.1 (mais detalhes em (SHINZATO; GOMES; WOLF, 2014) (SHINZATO, 2015)), que depende somente na condição local representada pela Equação 4.2 para cada par de nós conectados por uma aresta. A imagem resultante (Fig. 6c) é usada na etapa de agrupamento de pontos. A Equação 4.1 é definida como:

$$C(p) = \frac{1}{\|Q(p)\|} \sum_{n \in Q(p)} \text{sigmoid} \left( \frac{|p_y - n_y|}{\|p - n\|}, \cos(\theta), \sigma \right), \quad (4.1)$$

onde  $|p_y - n_y|$  corresponde à diferença absoluta dos valores de altura entre os dois pontos,  $\|p - n\|$  é a distância entre os pontos  $p$  e  $n$ ,  $\theta$  e  $\sigma$  são constantes,  $Q(p)$  é o conjunto de todos os 24 vizinhos de  $p$ , e  $\text{sigmoid}()$  é uma função definida na Equação 4.2:

$$\text{sigmoid}(x, \mu, \sigma) = 0.5 + \frac{0.5(x - \mu)\sigma}{\sqrt{1 + (x - \mu)^2\sigma^2}}, \quad (4.2)$$

onde  $\mu$  especifica o limiar efetivo e  $\sigma$  é um parâmetro de escala que influencia no limiar tangente. A intuição nesse método foi obtida de (SHINZATO; GOMES; WOLF, 2014), na qual a estimação dos obstáculos era binária.

#### 4.1.4 Agrupamento de pontos

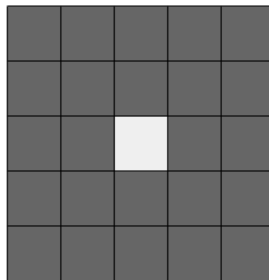
Os valores de disparidade e custo foram usados como entrada para o agrupamento de pontos em obstáculos. O primeiro ponto,  $p$ , é selecionado e, se ele não faz parte de um grupo, um novo rótulo de grupo é designado a ele. Todos os pontos que pertencem a vizinhança de  $p$  são recuperados e para cada vizinho  $q$  ( $q \in N$ , onde  $N$  é o conjunto de vizinhos de  $p$ ). Para ser considerado parte do mesmo grupo de  $p$ , as três seguintes condições devem ser satisfeitas:

1.  $q$  não faz parte de um grupo,
2.  $|(p_d - q_d)| \leq \eta$ ,
3. custo de  $q > \alpha$ ,

onde  $p_d$  e  $q_d$  correspondem aos valores de disparidade dos pontos. Se o ponto  $q$  satisfaz as condições, ele se torna parte do mesmo grupo de  $p$ . O processo continua para cada vizinho que se tornou parte do mesmo grupo até que todos os vizinhos tenham sido processados. Depois de toda a vizinhança de  $p$  ser analisada, outro ponto é escolhido e o mesmo processo é repetido.

O algoritmo se diferencia do conhecido algoritmo DBSCAN porque ele não usa os conceitos de pontos de borda ou núcleo e checa a condição do número mínimo de pontos somente para a validação de um grupo já formado. Se um grupo tem pontos suficientes, ele é considerado um obstáculo, caso contrário o grupo inteiro é considerado ruído. A condição que leva em consideração a chance de um ponto ser parte de um obstáculo é usada para separar pontos em obstáculos de pontos que fazem parte da área navegável. A condição de disparidade agrupa pontos de uma certa uniformidade e mudanças bruscas no mapa de disparidade não vão agrupar pontos vizinhos ao mesmo obstáculo. Os valores usados para  $\eta$  e  $\alpha$  foram 1 e 0,1, respectivamente. No final do algoritmo, um certo número de grupos que correspondem a obstáculos são detectados (Fig. 6d).

Figura 7 – 24-vizinhança



Fonte: Elaborada pelo autor.

## 4.2 Rastreamento

O rastreamento tem como objetivo conseguir dizer onde um objeto previamente detectado está ao longo de uma sequência de frames, caso ele ainda esteja presente na imagem. Várias são as possíveis formas de se realizar um rastreamento, a grande maioria dos algoritmos utiliza um filtro de estimação, um método de associação de dados e, em alguns, fluxo óptico.

Nos próximos parágrafos são explicados o Filtro de Kalman (WELCH; BISHOP, 1995), o Algoritmo de Fluxo Óptico Lucas Kanade Piramidal (HORN; SCHUNCK, 1981), o Forward-Backward Error (KALAL; MIKOLAJCZYK; MATAS, 2010) e o algoritmo de Munkres (MUNKRES, 1957).

### 4.2.1 Median Flow tracker

O algoritmo *Tracking-Learning-Detection* (TLD) (KALAL; MIKOLAJCZYK; MATAS, 2012) é um algoritmo de rastreamento que em sua versão inicial realizava o rastreamento de um único obstáculo. E quando proposto, demonstrou bons resultados em diferentes cenários, inclusive em casos onde o objeto rastreado mudava de aparência ao longo da sequência de frames. A base geral do algoritmo se concentra em que o rastreamento deve seguir o objeto de imagem a imagem. A Detecção deve detectar todas as aparências e corrigir o rastreamento quando necessário e o Aprendizado estimar os erros e realizar atualizações para evitar futuros erros. Com isso o algoritmo consegue aprender diversas aparências de um mesmo objeto e continuar seguindo-o, mesmo se ele mudar de orientação em relação à câmera.

A etapa de rastreamento do TLD utiliza o algoritmo de MF, as entradas para o algoritmo são duas imagens em instantes de tempo consecutivos  $I_{t-1}$  e  $I_t$  e uma caixa delimitadora ( $b_{t-1}$ ) indicando onde está localizado o obstáculo de interesse na imagem  $I_{t-1}$ . O algoritmo gera uma grade de  $10 \times 10$ , pegando uniformemente 100 *pixels* dessa imagem delimitada por  $b_{t-1}$ . O fluxo desses *pixels* é calculado usando o algoritmo Lucas-Kanade Pyramidal, seção 4.2.1.1, e para cada *pixel* é calculado o *Forward-Backward Error* (F-BE), seção 4.2.1.2, e o *Normalized Cross Correlation* (NCC), (LEWIS, 1995). Somente os 50% dos *pixels* cuja trajetória for de menor erro são utilizados para gerar uma nova caixa delimitadora como resposta da localização do objeto no instante de tempo atual  $b_t$ .

#### 4.2.1.1 Fluxo óptico

Segundo Berthold K.P. Horn e Brian G. Schunck, fluxo óptico é a distribuição de velocidades aparentes do movimento de padrões de brilho em uma imagem (HORN; SCHUNCK, 1981). Ele pode surgir do movimento relativo de objetos e do espectador. Basicamente, o fluxo óptico consegue estimar movimento entre os *pixels* de imagens consecutivas. Os *pixels* são monitorados ao longo do tempo e o seu movimento é convertido em vetores de velocidade. Em outras palavras, são encontrados os *pixels* correspondentes entre a imagem anterior e a atual.

Com isso, é possível obter o deslocamento de cada *pixel* e em seguida associar à cada *pixel* uma velocidade de deslocamento.

O algoritmo Lucas Kanade é um método de fluxo óptico, no entanto, nele grandes movimentos podem mover a previsão dos pontos para fora da janela local. E isso levou ao desenvolvimento do Lucas Kanade Piramidal, onde se começa a busca pelos níveis mais altos da pirâmide e depois sobre os mais baixos.

O algoritmo Lucas Kanade Piramidal tem como objetivo, utilizando o algoritmo Lucas Kanade, detectar grandes movimentos. Inicialmente utilizando apenas uma parte das imagens  $I_t$  e  $I_{t+1}$  é aplicado o Lucas Kanade. A estimativa de movimento do nível anterior é tomada como o ponto de partida para a estimativa de movimento para a próxima camada. Dessa forma, o processo é repetido, porém, com uma parte maior das imagens  $I_t$  e  $I_{t+1}$ , até chegar nas imagens originais.

#### 4.2.1.2 Forward-Backward Error

A ideia do F-BE (KALAL; MIKOLAJCZYK; MATAS, 2010) é estimar se a trajetória de um *pixel* estimada entre duas imagens em tempos consecutivos é confiável. Para isso, as trajetórias são calculadas tanto na ida, imagem  $I_{t-1}$  para imagem  $I_t$ , quanto na volta, imagem  $I_t$  para  $I_{t-1}$ . Ou seja, na primeira imagem  $I_{t-1}$  são escolhidos *pixels*  $p_{t-1}$  que fazem parte de um objeto cujo deslocamento deseja-se estimar para a segunda imagem  $I_t$  (no instante de tempo seguinte). Para isso, ambas as imagens, bem como as coordenadas dos *pixels* previamente escolhidos, são passados para um algoritmo que estima onde estes *pixels*,  $p_t^f$ , estão na segunda imagem  $I_t$ . O mesmo processo se repete com inversão das entradas. Os *pixels* resultantes  $p_t^f$  juntamente com a segunda imagem  $I_t$  e a primeira  $I_{t-1}$ , assim a saída depois da nova estimação será  $p_{t-1}^b$ . Assim, comparando os pontos iniciais,  $p_t^f$ , na imagem  $I_{t-1}$  com os pontos resultantes volta,  $p_{t-1}^b$ , é possível descobrir os pontos com trajetórias inconsistentes e o erro é dado pela distância euclidiana entre esses dois grupos de pontos, ou seja, se o mesmo fluxo ocorreu na ida e na volta ele é considerado correto, caso contrário, essa trajetória é descartada.

## 4.2.2 Filtro de Kalman

O filtro de Kalman é um dos algoritmos mais utilizados para eliminar ruído e inferir valores para variáveis não-observáveis. O algoritmo é capaz de estimar o estado de um sistema através de observações parciais baseado em um ciclo de iterações entre predição e correção (WELCH; BISHOP, 1995).

A primeira etapa do filtro é a predição, nela a estimativa e a covariância do erro são previstas baseadas nas variáveis de modelo do sistema. Em seguida, o ganho de Kalman é calculado e ele visa compensar a diferença entre a predição e a correção pela atualização das variáveis dada nova leitura. A covariância do erro indica o quão precisa a estimativa é.

Utilizando o filtro de Kalman é possível, por exemplo, inferir a velocidade de um objeto a partir de múltiplas localizações dele, mesmo sem nunca medir diretamente a velocidade, e a partir dessa estimação poder prever onde o objeto estará no instante de tempo seguinte. Isso é possível porque essas variáveis observáveis e não observáveis interagem entre si, são correlacionadas.

Em espaços multi-dimensionais é assignado uma média para cada uma das dimensões e a covariância (incerteza) será uma matriz quadrática de tamanho igual ao número de dimensões. O filtro segue um ciclo de predição e correção:

<b>Predição:</b>	<b>Correção:</b>
$X' = Fx + u$	$Y = Z - Hx$
$P' = FPF^T + Q$	$S = HPH^T + R$
	$K = PH^T S^{-1}$
	$X' = X + (Ky)$
	$P' = (I - KH)P$

Neste trabalho, usamos filtro de Kalman para modelar o comportamento dos obstáculos, então como não sabemos sua real movimentação, não há a necessidade da variável  $u$ .

$$X = \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} \quad (4.3)$$

A primeira etapa tenta prever, essa predição também conhecida como estimativa *a priori*, o novo estado  $X'$  e a nova matriz de covariância  $P'$ . A qualidade da predição é determinada pelo quão próximo o modelo do sistema está do sistema atual. A matriz  $F$  é a matriz de transição que deve ser construída de acordo com o modelo que se deseja estimar. Para um sistema onde pretende-se estimar posição  $x$ ,  $y$  e velocidade  $v_x$ ,  $v_y$ , a matriz  $F$  pode ser representada por:

$$F = \begin{pmatrix} 1 & 0 & \Delta_t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (4.4)$$

Essa matriz descreve a física de movimentação para atualização das variáveis.

$$\begin{pmatrix} x' \\ y' \\ vx' \\ vy' \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta_t & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta_t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ vx \\ vy \end{pmatrix} \quad (4.5)$$



onde a posição nova de  $x'$  depende da posição antiga  $x$  e da velocidade  $vx$ , a nova velocidade de  $vx'$  depende somente da velocidade antiga  $vx$ , e o mesmo para as demais variáveis.

$$x' = 1x + 0y + 0z + \Delta_t vx + 0vy + 0vz = x + \Delta_t vx \quad (4.6)$$

A etapa de predição (movimentação) seleciona a melhor estimativa  $x$  e multiplica pela matriz de transição de estados  $F$ , com isso será obtida uma nova localização para o agente.

$$x' = Fx \quad (4.7)$$

A matriz de covariância  $P$  que mensura a incerteza, é atualizada da seguinte forma, onde  $Q$  é a matriz de ruído na movimentação:

$$P' = FPF^T + Q \quad (4.8)$$

A etapa de atualização leva em consideração as leituras ( $Z$ ) dos sensores, no caso temos a leitura de posição, que são mensuradas pela etapa de detecção:

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.9)$$

Em seguida, é calculado o erro de predição da leitura, ou seja, a diferença entre a leitura atual e a leitura predita.

$$Y = Z - Hx, \quad (4.10)$$

onde  $Hx$  é a predição de leitura.

O erro é mapeado em uma matriz  $S$ , que é obtida pela incerteza no sistema no espaço de medidas, mais uma matriz  $R$  que caracteriza o ruído das leituras, que em seguida é mapeado na variável  $K$ , que é usada como um fator que determina o quanto a predição deve ser calibrada.

$$S = HPH^T + R \quad (4.11)$$

$$K = PH^T S^{-1} \quad (4.12)$$

E finalmente a estimativa e incerteza são atualizadas. A estimativa final é estimada pela calibração da predição com o erro de predição da leitura.

$$X' = X + (Ky) \quad (4.13)$$

A matriz de covariância ( $P'$ ) indica o grau de precisão da estimativa.

$$P' = (I - KH)P, \quad (4.14)$$

onde  $I$  é a matriz de identidade.

### 4.2.3 Algoritmo de Munkres

Uma das necessidades fundamentais para realizar o rastreamento de objetos é a necessidade de um algoritmo de associação de dados. Dadas dois conjuntos de objetos detectados em instantes de tempos consecutivos, o algoritmo de associação de dados deve fazer a correspondência entre os objetos dos dois conjuntos da melhor forma, de modo que seja possível saber se um mesmo objeto está presente nos dois conjuntos e essa informação possa ser usada no rastreamento.

O Algoritmo de Munkres (MUNKRES, 1957) é um método de associação de dados, sendo uma variação da abordagem proposta em (KUHN, 1955). Dados dois conjuntos de dados deseja-se fazer a melhor correspondência entre eles. Testar todas as possíveis combinações de correspondências possíveis se torna inviável conforme o número de entradas cresce, por isso existe a necessidade de um algoritmo que consiga fazer uma associação eficiente sem a necessidade de testar todos os possíveis casos. Esse algoritmo opera sobre uma matriz de custos,  $N \times M$ , onde  $N$  e  $M$  são os tamanhos dos dois conjuntos de dados, cada item  $k_{ij}$  dessa matriz é a distância entre o  $i$ -ésimo objeto do conjunto  $N$  e o  $j$ -ésimo elemento do conjunto  $M$ . O algoritmo então busca nessa matriz as melhores correspondências de forma que o custo total das distâncias entre os objetos seja o menor possível.

## 4.3 Considerações Finais

Este Capítulo apresenta a proposta de detecção e rastreamento de obstáculos. A etapa de detecção agrupa os pontos em obstáculos usando as informações do mapa de disparidade, mapa de custos e informação de 24-vizinhança. Em seguida, é realizado o rastreamento de obstáculos-alvo estimando suas localizações e velocidades com filtro de Kalman no plano paralelo ao chão. A associação de dados é feita usando o algoritmo Munkres que acha as melhores associações com o intuito de minimizar os custos. O algoritmo de Median-Flow é usado quando ocorre uma falha na detecção e nenhuma nova detecção é associada a um *tracklet* no instante de tempo atual.

As principais vantagens da abordagem proposta são: minimizar o número de suposições sobre o ambiente, não necessitar de mapas de disparidade densos, não precisar de nenhuma fase de treinamento, detectar tanto obstáculos estáticos quanto dinâmicos. Além disso, existe a contribuição para o projeto CaRINA, com a inclusão de um novo sistema no veículo autônomo que aumentará a possibilidade de aplicações a serem desenvolvidas no veículo (ultrapassagens de carros e desvio de obstáculos).

---

## RESULTADOS

---

Este Capítulo apresenta o processo de avaliação e os resultados obtidos. Na seção 5.1 são descritos os resultados da etapa de detecção. E na seção 5.2 o processo completo é avaliado, sendo descritos o processo de construção do *Ground Truth* (GT) e os experimentos realizados.

### 5.1 Avaliação da Detecção

Para avaliar a abordagem de detecção de obstáculos proposta nessa dissertação foram utilizadas 200 imagens. Cem imagens do dataset KITTI (GEIGER; LENZ; URTASUN, 2012) e 100 coletadas com a plataforma CaRINA. As imagens contêm diferentes tipos de obstáculos interagindo com o veículo, implicando-o em fazer alguma ação como frear ou ultrapassá-los. Todos os obstáculos visíveis foram manualmente selecionados para a construção do GT. Eles são identificados baseado em condições espaciais na cena. Por exemplo, se existe um poste entre a câmera e um carro passando (que na visão da câmera, divide o carro ao meio, o poste e o carro vão ser considerados três obstáculos, a parte esquerda do carro, o poste, e a parte direita do carro). Além disso, se as bordas de mais de um obstáculo não são claras eles são marcadas como um mesmo obstáculo. Obstáculos distantes da calçada ou rua são considerados *background* e agrupados em um único obstáculo. Só foram considerados *pixels* cujas distâncias fossem menor que 30 m, porque os valores de disparidade não são muito precisos depois dessa distância.

O único parâmetro diferente no algoritmo de detecção entre os dois conjuntos de imagens foi a variável de amostragem da imagem ( $\gamma$ ). No experimento com os dados do KITTI foi utilizado  $\gamma = 5$  e nos experimentos com os dados do CaRINA foi utilizado  $\gamma = 10$ . Isso porque as câmeras utilizadas nas capturas das imagens possuem distâncias focais e tamanhos de sensor CCD diferentes, assim um ajuste teve de ser feito para compensar essa diferença. Devido a essa diferença entre os dois sensores, a largura em *pixels* de uma pessoa detectada era menor nas imagens do KITTI do que nos dados do CaRINA.

Para construção do GT os obstáculos foram manualmente selecionados em todas as imagens. Nas imagens manualmente segmentadas, quando as bordas dos obstáculos não eram claras, elas foram rotuladas como um único obstáculo. Assim, quando algumas pessoas são detectadas como um único obstáculo, isso pode ser considerado um acerto, dependendo da visibilidade de suas bordas. Por outro lado, quando elas têm a borda visível, elas são agrupadas separadamente. Para os obstáculos detectados pela abordagem proposta foram construídos polígonos conforme (SKLANSKY, 1982). Cada polígono foi associado ao polígono do GT que possuía a maior área de intersecção. Para cada polígono do GT foi calculada a área de intersecção com cada polígono associado (mais de um polígono detectado poderiam ser associados com o mesmo polígono do GT) decrescido desse valor a área de intersecção entre os polígonos detectados. A área resultante foi dividida pela área do polígono do GT. Um acerto foi considerado quando a área resultante fosse maior que 0,4. Foram calculadas as métricas de *precision* e *recall* para cada imagem.

As imagens do KITTI possuem  $1242 \times 375$  (465750) *pixels* e o tempo necessário para processar uma imagem é 55 ms, as médias de *precision* e *recall* para todas as imagens foram 0,58 e 0,68. Uma alta taxa de *recall* é desejada em sistemas de veículos autônomos, porque todos os obstáculos perto do veículo devem ser detectados. Os obstáculos foram corretamente detectados na maior parte das imagens, no entanto, alguns não foram identificados devido a falhas no mapa de disparidade e principalmente porque esses mapas perdem precisão em função da distância. Quando a câmera se aproxima dos obstáculos, a chance dele ser detectado aumenta. Em alguns casos, obstáculos visíveis podem não ser segmentados pela condição de disparidade, fazendo com que sejam agrupados em um único obstáculo. Os obstáculos foram corretamente segmentados na maioria das imagens, por exemplo, pedestres caminhando (Figura 8a (10, 11, 12, 15, 17, 19 e 24)), (Figura 8c (7 e 8)), carros (Figura 8b (4, 5, 6 e 7)), (Figura 8d (7)), ciclistas (Figura 8d (6 e 10)), postes (Figura 8a (22, 25 e 28)), (Figura 8b (1, 8 e 9)) e (Figura 8d (11, 12, 16, 18, 20, 21, 24, 25 e 26)) e árvores (Figura 8c (2 e 5)).

Em alguns casos, os obstáculos não foram identificados por causa de erros no mapa de disparidade ou por causa do agrupamento de grupos de obstáculos em um mesmo obstáculo pelo método (pessoas caminhando lado a lado) e agrupamento individual pelo GT (Figura 8a obstáculo 14)). Como o parâmetro de disparidade máxima no algoritmo ELAS utilizado no teste foi 100, alguns obstáculos muito próximos à câmera não foram detectados (Figura 9).

Apesar de termos limitado a distância para 30 metros para avaliar a detecção, a abordagem proposta pode alcançar obstáculos mais distantes de 30 metros, como mostrado na Figura 10. Outra razão para a baixa precisão são as guias que não são consideradas obstáculos no GT pelo fato de terem diferentes alturas e formatos na visão perspectiva, o que dificulta a sua detecção como um único obstáculo.

Também foram avaliados dados coletados com a plataforma CaRINA. As imagens foram coletadas usando o *baseline* estendido de uma Bumblebee XB3 (BBX3-13S2C-38) Point Grey.

Figura 8 – Resultados da detecção para a sequência de imagens do KITTI. A primeira coluna é a imagem da câmera esquerda. A segunda representa cada obstáculo pelo seus conjuntos de pontos.

(a) Grupo de pessoas (14), 7 pedestres (10, 11, 12, 15, 17 e 19) e postes (22, 25 e 28).



(b) Quatro carros (4, 5, 6 e 7), postes (1, 8 e 9) e vegetação (10, 11 e 12).



(c) Dois pedestres (7 e 8), um poste (5) e vegetação e parede (1, 3 e 4).



(d) Dois ciclistas (6 e 10), postess (11, 12, 14, 16, 17 18, 19, 20, 21, 22, 23, 24, 25 e 26), um carro (7) e parede (4).



(e) Dois carros (6 e 7), um tronco de árvore (1) e vários postes.



Figura 9 – Obstáculos muito próximos da câmera não são detectados por causa do parâmetro de disparidade máxima.



Fonte: Elaborada pelo autor.

Figura 10 – Imagens sem limitação de distância.

(a) Imagem do dataset KITTI. Quatro dos cinco carros visíveis foram detectados (2, 3, 4 and 5). Um carro foi erroneamente agrupado com o *guardrail*.



(b) Imagem do dataset KITTI. Dois carros distantes (4 e 5) detectados.



Fonte: Elaborada pelo autor.

O processo de avaliação foi o mesmo comentado anteriormente. Pessoas foram corretamente detectadas nas Figuras 11a (3, 4 e 5) 11b (5, 6 e 7) e 11c (5 e 6), e carros nas Figuras 11a (7 e 8), 11b (8 e 9) e 11c (7, 8 e 9). O precision e recall alcançados foram 0,63 and 0,76, respectivamente. As imagens possuem  $1280 \times 960$  (um total de 1228800) *pixels* e o tempo de processamento foi 114 ms, incluindo retificação, construção do mapa de disparidade, computação dos custos e agrupamento. Todas as imagens estão disponíveis para avaliação qualitativa em <https://youtu.be/CD6X99RacgM>.

## 5.2 Avaliação do Rastreamento

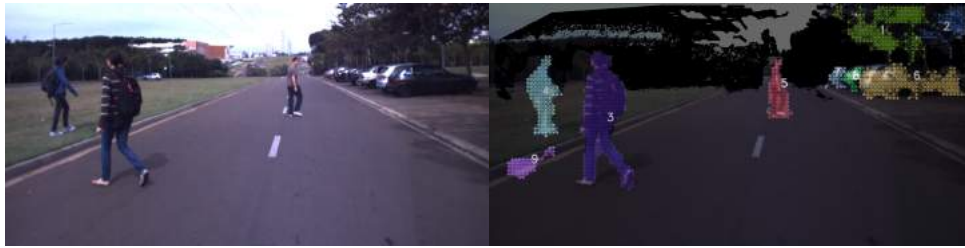
Para avaliar quantitativamente a abordagem proposta foram utilizadas várias métricas que visam mensurar a qualidade do algoritmo de rastreamento. Foram utilizadas, dentre outras, para avaliação as métricas MT, ML, PT (ZHANG; LI; NEVATIA, 2008) e MOTA e MOTP (BERNARDIN; STIEFELHAGEN, 2008), assim como *Precision*, *Recall*, *F1-measure*. Para calcular todas essas métricas o sistema de avaliação do KITTI-DATASET (GEIGER; LENZ; URTASUN, 2012) foi adaptado, usando na etapa de associação de objetos a distância euclidiana entre as estimativas em coordenadas no plano  $Z \times X$  (plano paralelo ao chão) do método e do GT, ao invés de somente determinar a área de sobreposição entre as caixas delimitadoras na imagem esquerda. Isso foi feito para evitar o *bias* de uma abordagem na perspectiva da imagem. Com o intuito de remover o *bias* do algoritmo de disparidade, a posição 3D da cada objeto no GT é a média de todos os pontos manualmente selecionados no objeto. É importante ressaltar que esses pontos são gerados por reconstrução usando o mapa de disparidade. Visto que nosso algoritmo de correspondência é o mesmo para gerar o GT, no entanto, neste, para aumentar a precisão, o mapa é calculado de forma densa.

Na seção 5.2.1 são explicadas as métricas utilizadas na avaliação e o processo de construção do GT.



Figura 11 – Resultados da detecção para a sequência de imagens do CaRINA. A primeira coluna é a imagem da câmera esquerda. A segunda representa cada obstáculo pelo seus conjuntos de pontos.

(a) Três pedestres (3, 4 e 5), e três dos quatro carros visíveis (6, 7 e 8).



(b) Dois carros (8 e 9) e pedestres (5, 6 e 7).



(c) Três dos quatro carros visíveis (7, 8 e 9) e pedestres (5 e 6).



(d) Três pessoas corretamente detectadas (5, 6 e 7), dois carros estacionados (10 e 11) e dois carros foram agrupados como um único obstáculo (8).



Fonte: Elaborada pelo autor.

### 5.2.1 Métricas de Avaliação

Para avaliar a qualidade da detecção foram utilizadas as métricas:

- *Recall:*

$$\frac{\text{Objetos corretamente detectados}}{\text{Total de objetos do GT}}$$

- *Precision:*

$$\frac{\text{Objetos corretamente detectados}}{\text{Total de objetos detectados}}$$

- F1:

$$\frac{2 * (Precision \times Recall)}{(Precision + Recall)}$$

- FA/Frm: taxa de alarmes falsos (quanto menor melhor).
- VP (Verdadeiro Positivos): obstáculos corretamente detectados.
- FP (Falso positivos): obstáculos erroneamente detectados.
- Alvos perdidos: obstáculos que deveriam ter sido detectados.

As métricas utilizadas de (ZHANG; LI; NEVATIA, 2008) foram:

- *Mostly Tracked* (MT): porcentagem de trajetórias do GT que são cobertas pela saída do rastreamento por mais de 80% .
- *Mostly Lost* (ML): porcentagem das trajetórias do GT que são cobertas pelo rastreamento por menos do que 20%.
- *Partially Tracked* (PT): 1.0-MT-ML. Porcentagem das trajetórias do GT que são cobertas pelo rastreamento de 20% a 80%.
- Fragmentos: número total de vezes que uma trajetória do GT é interrompida no resultado do rastreamento, quanto menor melhor.
- Troca de IDs: o número total de vezes que uma trajetória rastreada muda o ID correspondido ao GT, (quanto menor melhor).

As métricas utilizadas de (BERNARDIN; STIEFELHAGEN, 2008) foram *Multiple Object Tracking Accuracy* (MOTA) e *Multiple Object Tracking Precision* (MOTP), que medem, respectivamente, a qualidade do rastreamento de objetos e a precisão da estimativa de localização.

O MOTA mede o quão bem o método consegue detectar objetos e manter suas trajetórias, independente de quão preciso são as estimativas de localização.

$$MOTA = 1 - \frac{FN + FP + \text{Troca de IDs}}{nGT}$$

,

onde *FN* é o número de falso negativos, ou seja, objetos que deveriam ter sido detectados e não foram, *FP* corresponde ao número de falso positivos, ou seja, objetos erroneamente detectados e *nGT* é o número total de objetos no GT.

O MOTP mede a habilidade do método de estimar as posições dos objetos independente dele manter trajetórias consistentes. Ou seja, ele mede as posições estimadas para pares correspondidos entre o método e o GT e divide pelo número total de correspondências certas.



$$\frac{\sum_{i,t} 1 - d_t^i}{tp}$$

onde  $d_t^i$  é a distância euclidiana entre cada um dos pares de hipóteses e objeto associados, onde os pares eram correspondidos caso a distância fosse menor que 0.5m e  $tp$  é o número de objetos que foram associados com o GT.

### 5.2.2 Construção do Ground Truth

Considerando as imagens coletadas utilizando a plataforma CaRINA, foi necessário construir o GT, que nada mais é do que a resposta verdadeira para a detecção e rastreamento. Para se construir o GT foram selecionados na imagem todos os obstáculos entre 1,0m e 2,0m de altura. Essa seleção foi feita utilizando uma caixa delimitadora que englobava toda a extensão do obstáculo e também uma caixa delimitadora menor que retirava uma amostra do objeto. Foi garantido que essa amostra compreendia somente *pixels* que faziam parte do objeto. Com essa amostra e um método de cálculo de disparidade denso foi possível fazer uma média da estimativa 3D desses pontos. Essa média foi utilizada como a posição 3D de cada um dos objetos. Juntamente com a posição 3D foi armazenado um número que identificava unicamente aquele obstáculo ao longo de toda a sequência de frames. Com essa informação foi possível avaliar a qualidade da resposta do método proposto usando métricas apresentadas em (BERNARDIN; STEFELHAGEN, 2008) e (ZHANG; LI; NEVATIA, 2008), que medem a eficiência do rastreamento e da detecção comparadas ao GT.

### 5.2.3 Avaliação

Para realizar a avaliação, foram coletados dados no Campus II da USP, ilustrando diversas situações onde obstáculos atravessam a área na qual o veículo autônomo circulava. A avaliação quantitativa de um sistema genérico de detecção e rastreamento não é uma tarefa fácil. Visto que até mesmo para seres humanos pode ser difícil segmentar obstáculos em cenários complexos. Foram coletados dados em situações específicas em um cenário com poucos prédios e vegetações.

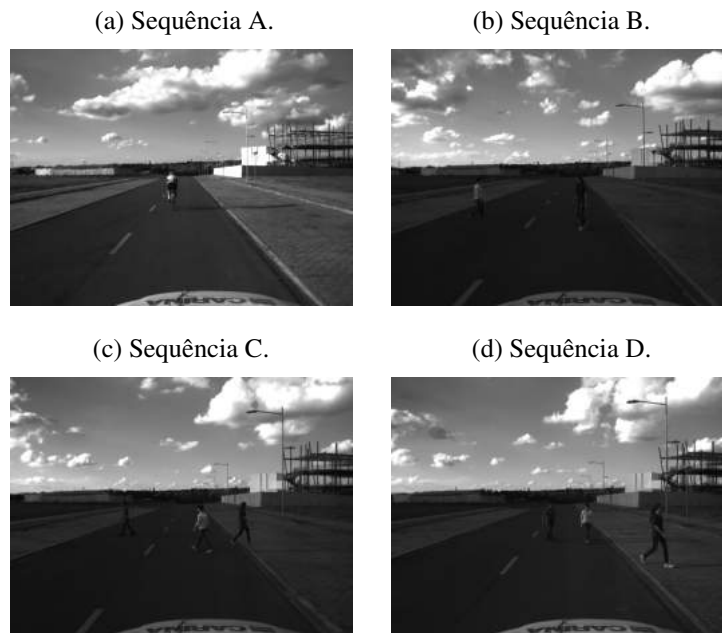
Na primeira sequência (A), Figura 12a, um ciclista pedala em frente ao carro autônomo. Na segunda sequência (B), Figura 12b, dois pedestres atravessam em frente ao carro saindo de uma calçada em direção ao lado oposto, cruzando-se à frente do veículo. Na terceira sequência (C), Figura 12c, dois pedestres atravessam do lado direito da rua para o lado esquerdo e um terceiro pedestre faz a travessia ao contrário. A quarta sequência (D), Figura 12d, três pedestres se deslocam na frente do veículo autônomo com trajetórias distintas.

Entre as etapas de detecção e rastreamento uma filtragem baseada em aspectos geométricos e informação 3D foi feita em todos os objetos, definindo limiares de altura mínima e máxima para eliminar grande parte dos objetos não-alvo. Como o método não realiza em

nenhum momento a diferenciação entre tipos de obstáculos, obstáculos só são considerados se eles possuem um altura mínima de 1m e uma altura máxima de 2 metros.

Nas próximas seções são apresentados os resultados de detecção e rastreamento para quatro ambientes controlados, nomeados ciclista (Sequência A), seção 5.2.4, 2 pedestres (Sequência B), seção 5.2.5, 3 pedestres (Sequência C), seção 5.2.6 e 3 pedestres desordenados (Sequência D), seção 5.2.7.

Figura 12 – Imagem ilustrando as diferentes sequências utilizadas para avaliação do método.



Fonte: Elaborada pelo autor.

### 5.2.4 Sequência A

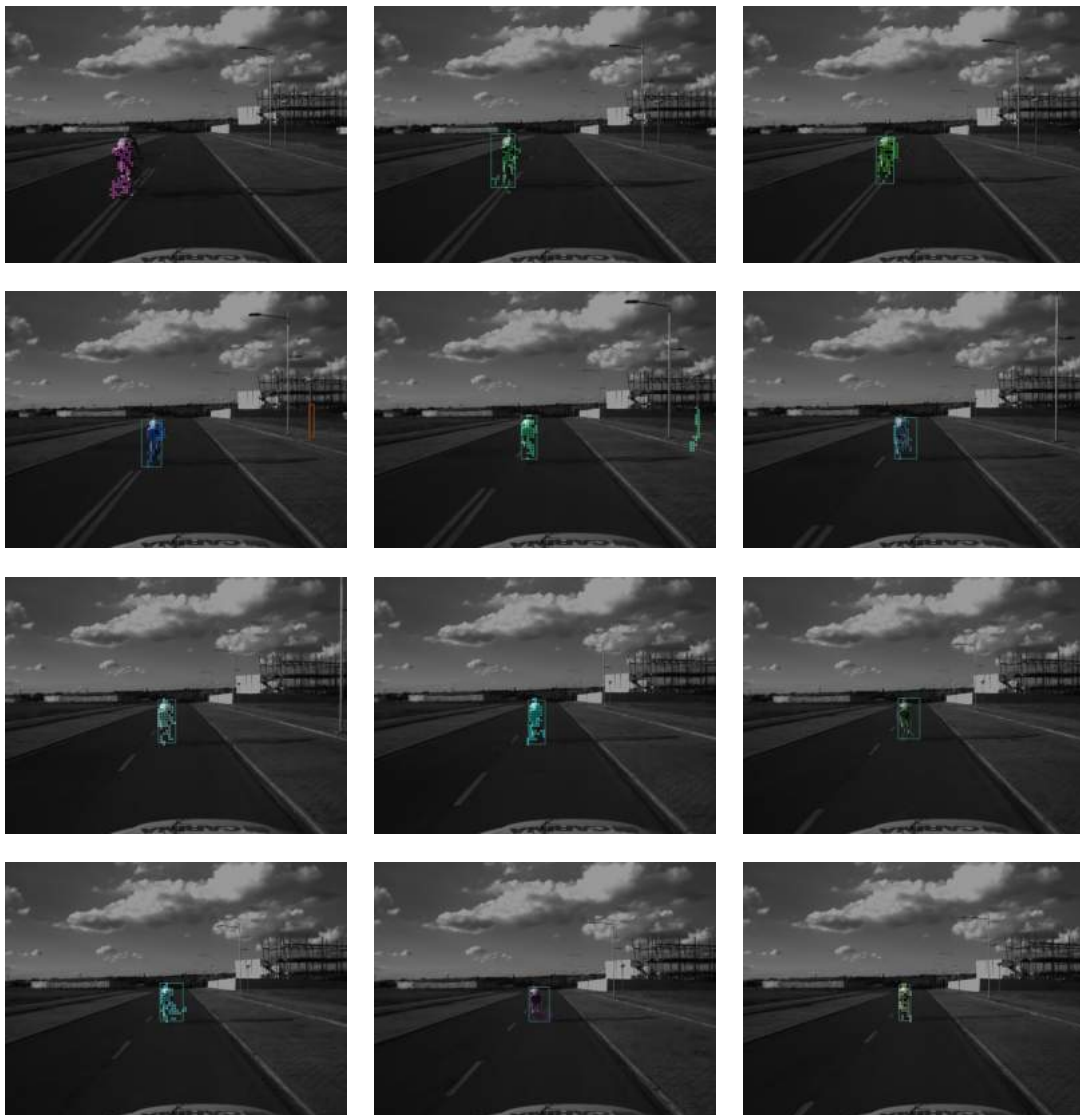
A primeira sequência de imagens utilizadas no teste consiste em 266 frames onde um ciclista ultrapassa o veículo autônomo. O ciclista no primeiro momento está à esquerda do veículo autônomo e desloca-se até a sua frente, onde permanece até o fim da sequência de imagens, como pode ser visualizado na Figura 13.

Na Figura 14 e na Tabela 1 é possível ver os resultados positivos na etapa de detecção. No entanto, é interessante ver como as estimativas do nosso sistema se diferem do GT em coordenadas 3D, mais especificamente no plano  $Z \times X$  (plano paralelo ao chão). Na Figura 14d é possível notar que exceto pelos obstáculos que não existem no GT (hipótese 2 e 3). O trajeto gerado pelo sistema de rastreamento (hipótese 1) chega muito perto do caminho do GT (Objeto 1).

O ciclista foi rastreado corretamente na maior parte das imagens. Ele primeiramente é detectado a uma distância de 7 metros do veículo no segundo frame da sequência. A última detecção ocorre à distância de 31 metros no frame de número 255. A partir do frame 263 o

ciclista não é mais detectado (distância maior que 31m). O número de trajetórias estimadas e trajetórias do GT não coincide porque o sistema detectou alguns postes ao lado direito do carro e realizou o rastreamento deles por alguns frames. Os valores de MOTA e MOTP refletiram esse erro causado pelo detector. É possível notar que na Figura 14d a posição de todos os objetos detectados em relação ao carro. Isso explica porque objetos estáticos estão se movendo em direção ao carro, enquanto o veículo está fixo na posição (0,0). O fato de as leituras serem relativas ao carro não é um problema desde que o GT adote o mesmo sistema de coordenadas.

Figura 13 – Imagens da sequência A. Nessa sequência um ciclista no primeiro momento está à esquerda do veículo autônomo desloca-se até a sua frente, onde permanece até o fim da sequência de imagens. Os pontos coloridos agrupados correspondem à saída da etapa de detecção. O retângulo ao entorno corresponde à saída da etapa de rastreamento



Fonte: Elaborada pelo autor.

Na Figura 15 é possível ver um dos erros mais comuns em todas as sequências. Nela alguns *pixels* que fazem parte do chão são agrupados junto ao ciclista. Esse erro ocorre devido a falhas na computação do mapa de custos e causa erros no cálculo das dimensões dos obstáculos.

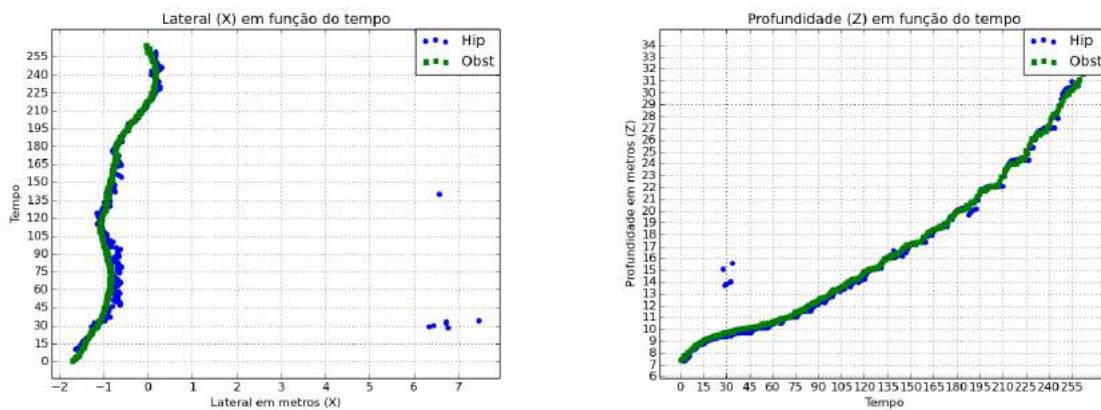
Além disso, devido a falhas no mapa de disparidade, a estimação de altura de alguns postes ao lado da rua foi prejudicada e por isso eles foram considerados como obstáculos alvo.

Devido a erros na computação do mapa de custos, alguns pontos podem ser agrupados como parte do obstáculo, apesar de fazerem parte da área navegável, como pode ser visto na Figura 15.

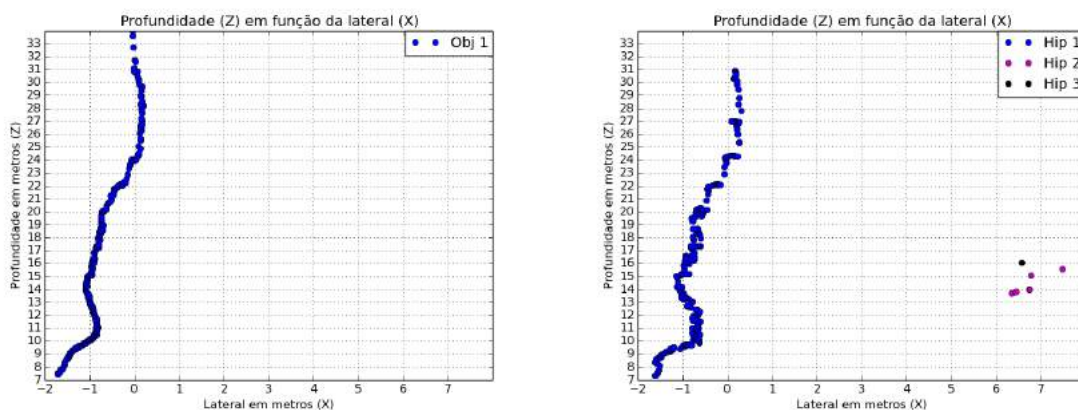
Para melhor visualização imprimimos as coordenadas X em função do tempo (Figura 14a), Z em função do tempo (Figura 14b) e as coordenadas em *birds-eye-view* do deslocamento do ciclista (Figura 14d). Todas as coordenadas estão em relação à câmera acoplada ao carro. É possível ver nas figuras que alguns erros na detecção dos obstáculos são captados, representados pelos pontos no canto inferior direito na Figura 14d.

Figura 14 – Gráficos comparando o resultado da sequência A com o GT

- (a) Gráfico comparando a variação de X ao longo dos frames (b) Gráfico comparando a variação de Z ao longo dos frames



- (c) Gráfico comparando a variação de X e Z no GT (d) Gráfico comparando a variação de X e Z no método



Fonte: Elaborada pelo autor.

Além de avaliar quantitativamente o rastreamento e a detecção, essa sequência foi usada para fazer duas análises. A primeira a respeito do uso do filtro de Kalman na seção 5.2.4.1, e a segunda sobre o que alguns parâmetros refletem nas saídas das métricas MOTA e MOTP na

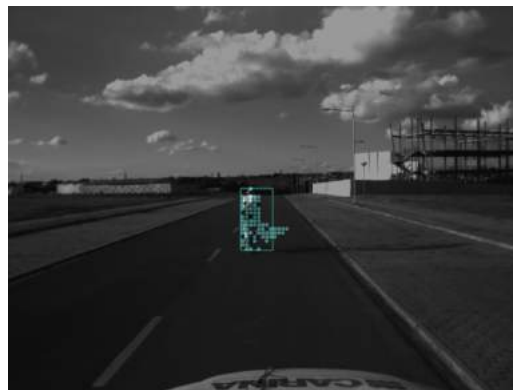
seção 5.2.4.2.

Tabela 1 – Resultados quantitativos da avaliação da sequência A.

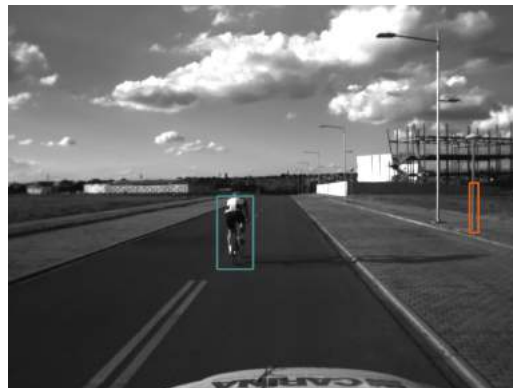
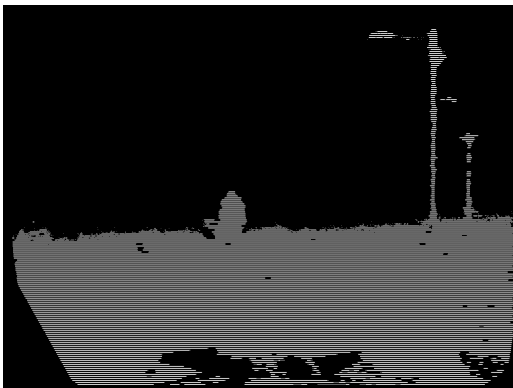
Métricas Rastreamento		Métricas Detecção	
MOTA	0,703008	Recall	0,849624
MOTP	0,753267	Precision	0,852830
MT	1,000000	F1	0,851224
PT	0,000000	FA	0,146067
ML	0,000000	Número de Verdadeiro Positivos	226
Trocas de Ids	0	Número de Falso Positivos	39
Fragmentações	16	Alvos perdidos	40
Número de trajetórias no GT	1	Número de objetos no GT	266
Número de trajetórias do método	3	Número de objetos detectados pelo método	265

Figura 15 – Erros na etapa de detecção da sequência A.

- (a) Mapa de custo exemplificando porque a detecção agrupou pontos do chão juntamente com o (b) *Pixels* no chão agrupados juntamente com o ciclista.



- (c) É possível notar que o poste menor localizado à direita na imagem teve uma segmentação em sua extensão, e por isso ele foi considerado um obtáculo-alvo. Foi aplicado um threshold no mapa de disparidade para facilitar a visualização. (d) Poste à direita (caixa delimitadora laranja) detectado erroneamente



Fonte: Elaborada pelo autor.

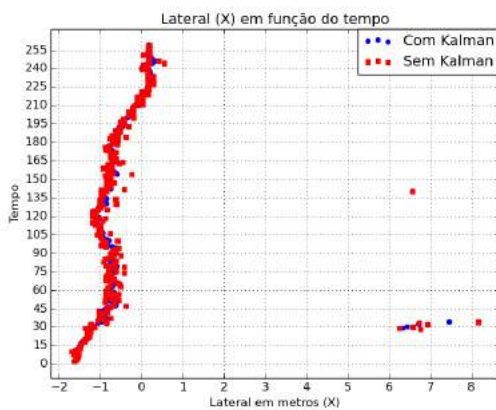


## 5.2.4.1 Análise do Filtro de Kalman

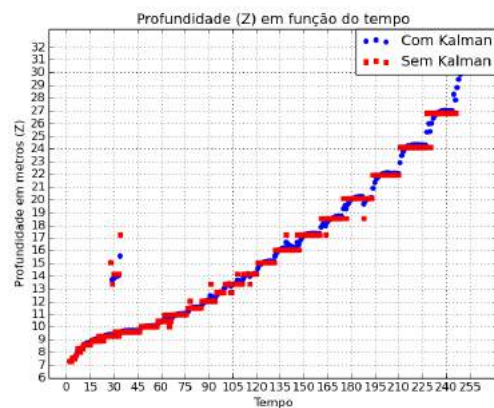
Também foi avaliada a influência do filtro de Kalman na saída desta sequência. A Figura 16 mostra que sem o Filtro de Kalman o obstáculo é detectado várias vezes ao longo do eixo X e introduz saltos ao longo do eixo Z que aumentam com a distância para o veículo. A razão para esses saltos é que o mapa de disparidade usado é esparso e não denso, ou seja, existem várias linhas na imagem que não são monitoradas. Considerando a Figura 17 e assumindo que a disparidade só é computada nas linhas verdes, existem saltos entre 10ms, 20ms e 30ms. Desse modo, um objeto é detectado na mesma linha monitorada durante vários frames. Conseqüentemente, a movimentação é detectada quando o objeto para de ocupar a linha monitorada de baixo e começa a ser detectado na linha monitorada de cima. E esse fenômeno é corrigido pelo uso de filtro de Kalman, que leva a uma estimativa de trajetória contínua.

Figura 16 – Gráfico comparando o resultado obtido com e sem o uso do filtro de Kalman.

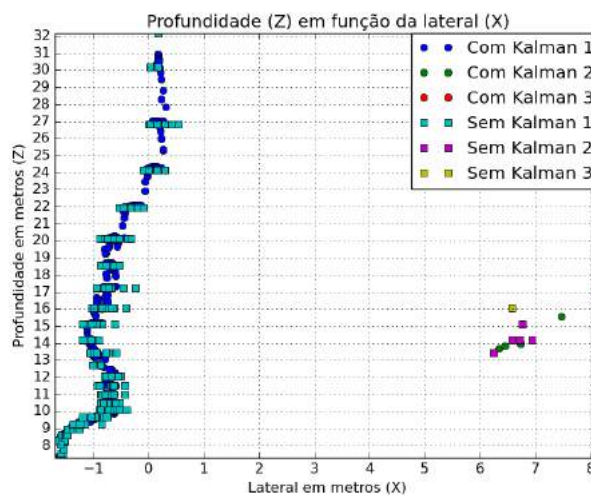
(a) Variação de X (com e sem Filtro de Kalman)



(b) Variação de Z (com e sem Filtro de Kalman)

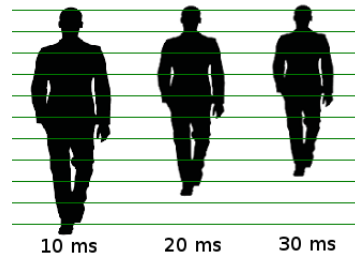


(c) Variação de X e Z (com e sem Filtro de Kalman)



Fonte: Elaborada pelo autor.

Figura 17 – A detecção e rastreamento são baseadas em um mapa de disparidade esparso, levando a saltos na etapa de detecção.



Fonte: Elaborada pelo autor.

#### 5.2.4.2 Análise do MOTA e MOTP

Como discutido na seção 5.2, as métricas de MOTA e MOTP servem para mensurar a qualidade do rastreamento. É possível observar que através da variação do critério de associação de dados ( $T$ ) que quanto mais permissivo ele for, melhor será o valor do MOTA e menor será o valor do MOTP. Isso porque o MOTP mede quão preciso é o método. Por isso se for aceito um raio muito grande para fazer a associação desses dados, esse valor será refletido em um MOTP menor.

Tabela 2 – Análise MOTA e MOTP

Métricas	$T = 0.4$	$T = 0.5$	$T = 0.6$
MOTA	0,462406	0,703008	0,778195
MOTP	0,786123	0,753267	0,740799

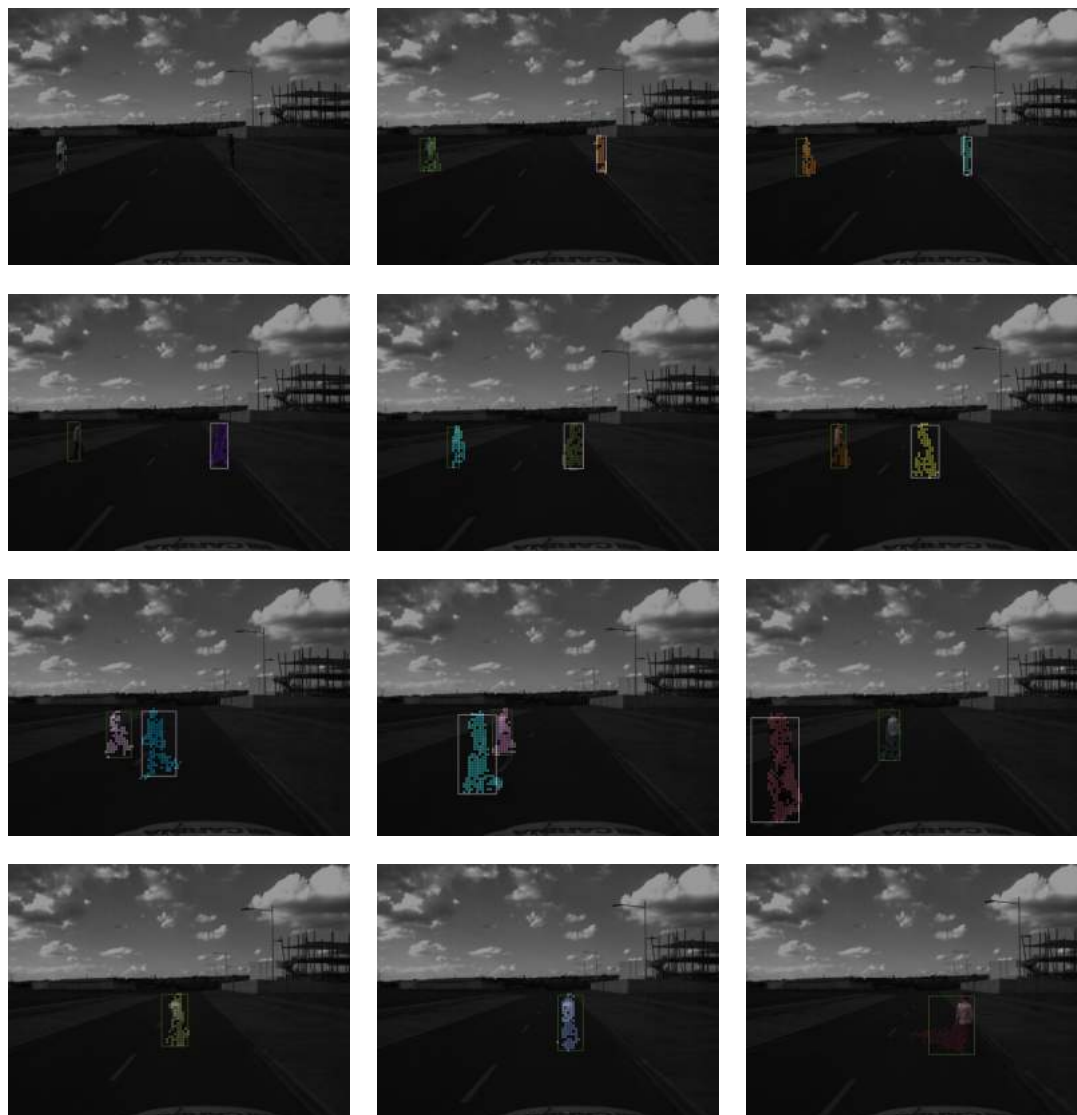
#### 5.2.5 Sequência B

A sequência B (Figura 18) possui 163 imagens, duas pessoas estão paradas, uma do lado esquerdo e outra do lado direito da rua. Ambas cruzam na frente do veículo autônomo e atravessam para o outro lado. Essa foi a sequência que alcançou os maiores valores de MOTA e MOTP. Em nenhum momento os IDs são trocados e o rastreamento é mantido pela maioria das imagens. Apesar de ocorrerem alguns falso positivos, eles não chegam a ser rastreados pelo fato de não manterem consistência entre dois frames. Assim, as únicas trajetórias mantidas são das duas pessoas presentes na cena.

Similar ao caso do ciclista, a Figura 18 mostra os resultados de detecção e rastreamento na perspectiva da imagem, permitindo ao leitor confirmar resultados precisos. A Figura 19 ilustra o caminho completo de cada objeto rastreado relativo ao veículo no plano  $Z \times X$ . É possível ver que as trajetórias estimadas correspondem precisamente ao GT. Isso é confirmado pelos resultados quantitativos listados na Tabela 3.



Figura 18 – Imagens da sequência B. Nessa sequência duas pessoas estão paradas, uma do lado esquerdo e outra do lado direito da rua. Ambas cruzam na frente do veículo autônomo e atravessam para o lado oposto. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento



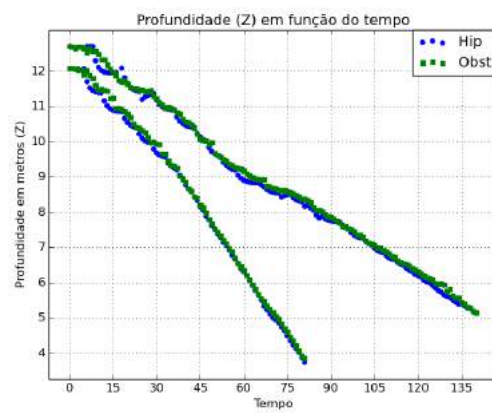
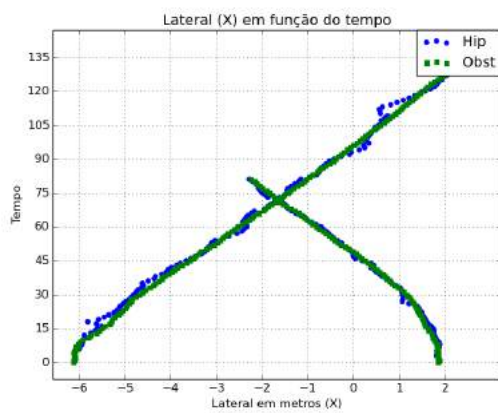
Fonte: Elaborada pelo autor.

Tabela 3 – Resultados quantitativos da avaliação da sequência B.

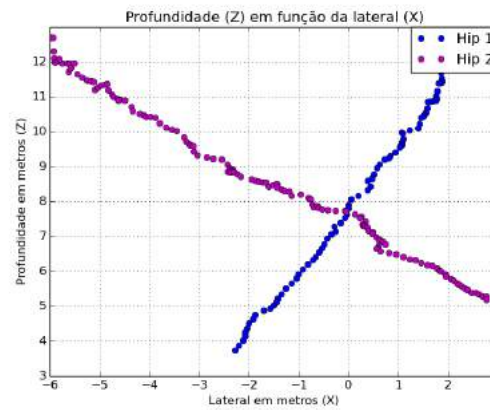
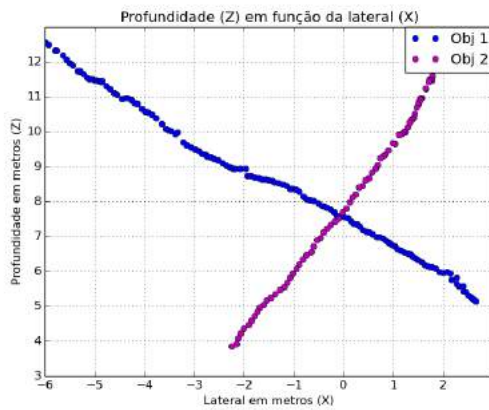
Métricas Rastreamento		Métricas Detecção	
MOTA	0,941704	Recall	0,946188
MOTP	0,837021	Precision	0,995283
MT	1,000000	F1	0,963134
PT	0,000000	FA	0,970115
ML	0,000000	Número de verdadeiro positivos	211
Trocas de id	0	Número de falso positivos	1
Fragmentações	2	Alvos perdidos	12
Número de trajetórias no GT	2	Número de objetos no GT	223
Número de trajetórias do método	2	Número de objetos detectados pelo método	212

Figura 19 – Gráficos comparando o resultado da sequência B com o *Ground Truth*

- (a) Gráfico comparando a variação de X ao longo dos frames (b) Gráfico comparando a variação de Z ao longo dos frames



- (c) Gráfico comparando a variação de X e Z no GT (d) Gráfico comparando a variação de X e Z no método



Fonte: Elaborada pelo autor.

### 5.2.6 Sequência C

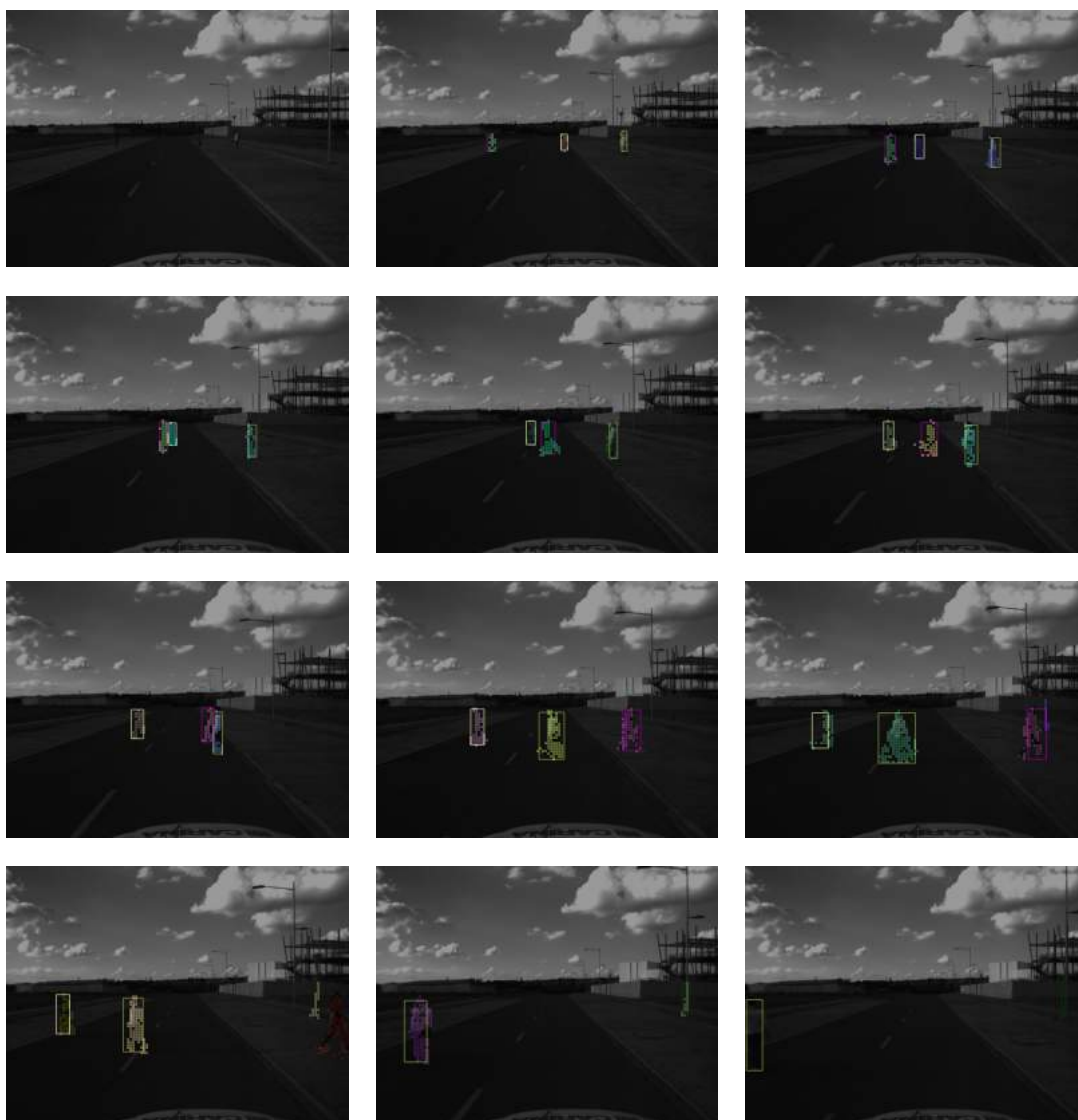
A terceira sequência, visualizada na Figura 20, possui 224 frames, onde três pessoas atravessam na frente do veículo. Duas pessoas atravessam do lado direito do carro para o lado esquerdo e uma pessoa faz o inverso. A dificuldade nesse cenário é o número maior de trajetórias que o método tem que lidar. Apesar de ocorrerem oclusões, o método consegue manter as três trajetórias sem se perder.

Nessa sequência, o rastreador começa a rastrear a pessoa da direita, mas devido a falhas na detecção nos frames seguintes o tracklet acaba sendo eliminado. O mesmo acontece com a segunda pessoa a ser detectada. Quando o veículo autônomo se aproxima mais, as pessoas são novamente detectadas e uma nova instância do rastreamento é assignada para elas. Por isso, na Figura 21d é possível ver que o número de hipóteses (8) de obstáculos, é muito maior que o número real (3), Figura 21c. No entanto, o comprimento dessas trajetórias errôneas é pequeno, visto que a maior parte do tempo os obstáculos são corretamente rastreados pelo mesmo *tracklet*. No final da sequência, foi criado um *tracklet* erroneamente para um poste ao lado direito do carro autônomo. Os resultados quantitativos podem ser vistos na Tabela 4.

Tabela 4 – Resultados quantitativos da avaliação da sequência C.

Métricas Rastreamento		Métricas Detecção	
MOTA	0,603738	Recall	0,717757
MOTP	0,831195	Precision	0,862921
MT	0,333333	F1	0,783673
PT	0,666667	FA	0,271111
ML	0,000000	Número de verdadeiro positivos	384
Trocas de id	0	Número de falso positivos	61
Fragmentações	17	Alvos perdidos	151
Número de trajetórias no GT	3	Número de objetos no GT	535
Número de trajetórias do método	8	Número de objetos detectados pelo método	445

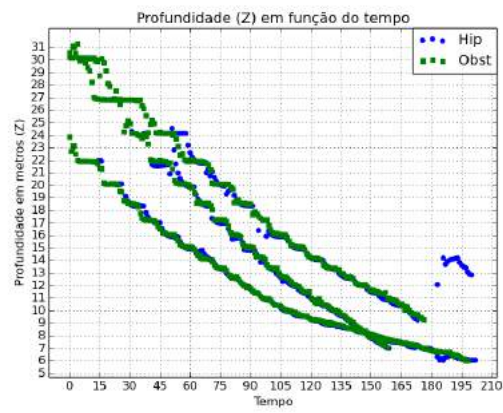
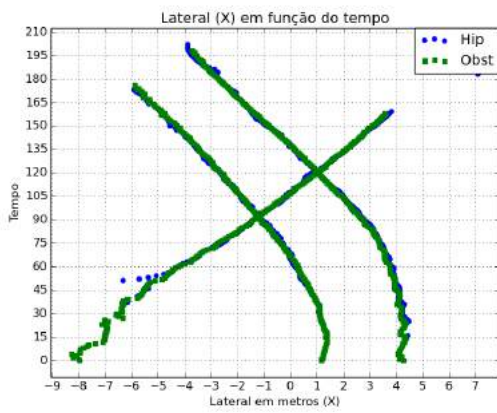
Figura 20 – Imagens da sequência C. Nessa sequência três pessoas atravessam na frente do veículo. Duas pessoas atravessam do lado direito do carro para o lado esquerdo e uma pessoa faz o inverso. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento



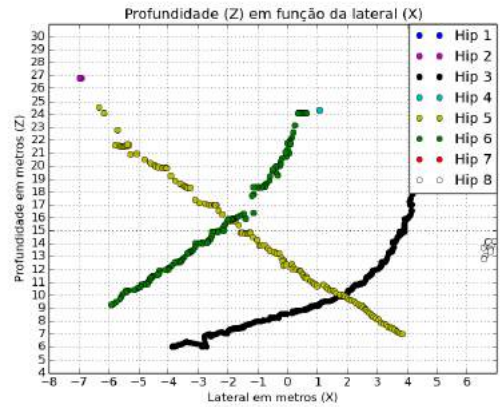
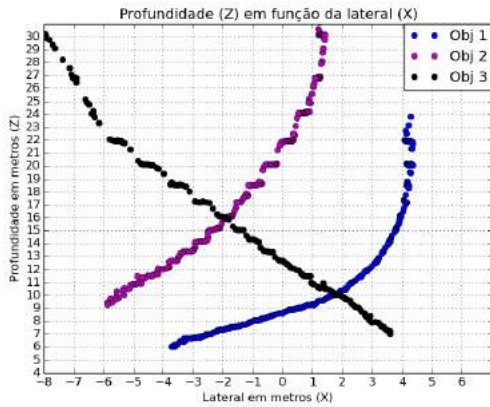
Fonte: Elaborada pelo autor.

Figura 21 – Gráficos comparando o resultado da sequência C com o *Ground Truth*

- (a) Gráfico comparando a variação de X ao longo dos frames  
 (b) Gráfico comparando a variação de Z ao longo dos frames



- (c) Gráfico comparando a variação de X e Z no GT  
 (d) Gráfico comparando a variação de X e Z no método

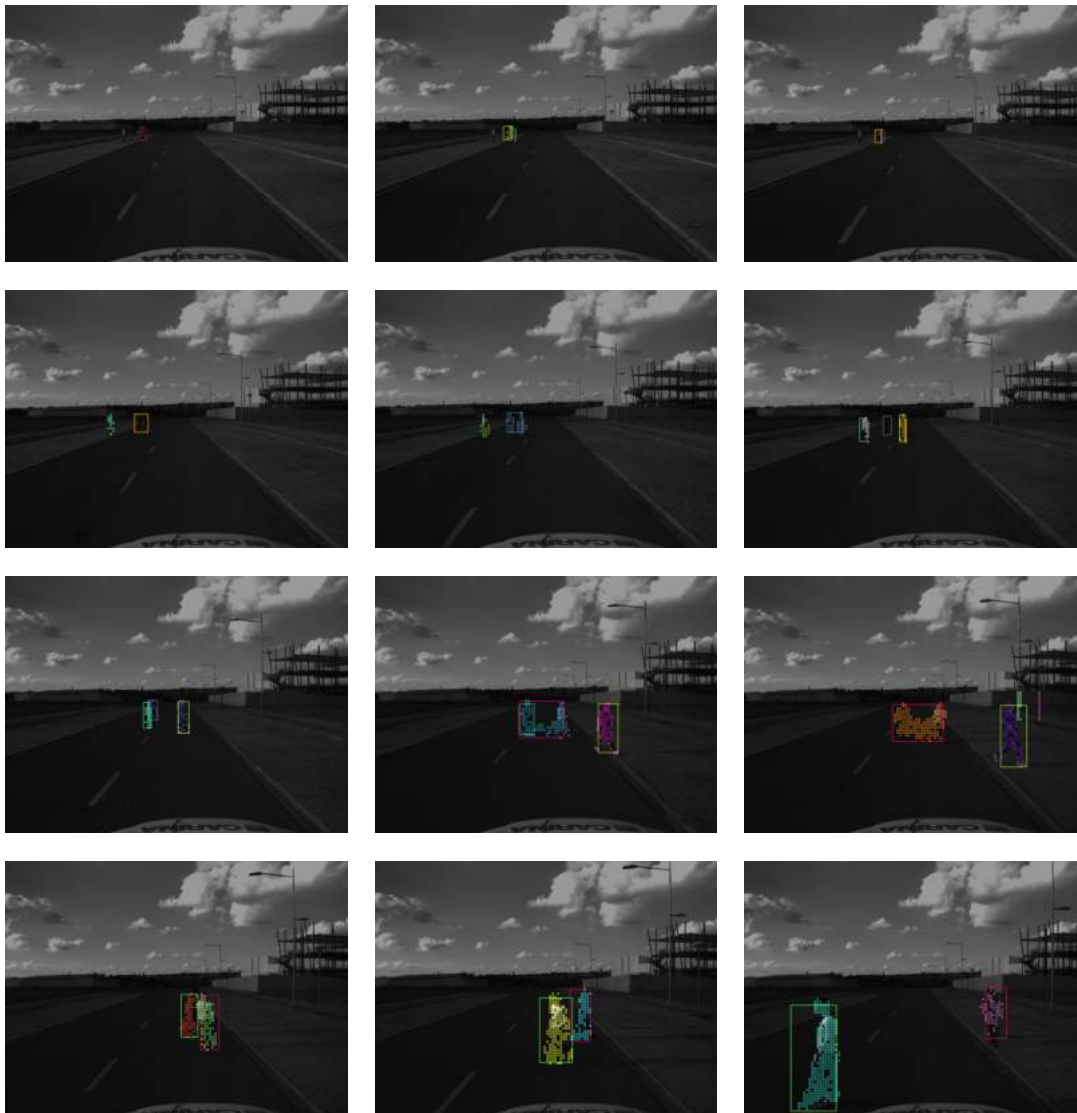


Fonte: Elaborada pelo autor.

### 5.2.7 Sequência D

A sequência D, vista na Figura 22, corresponde a 333 frames, onde três pessoas atravessam na frente do veículo autônomo. Essa foi a sequência com os piores resultados. À medida que os obstáculos estão distantes do carro a precisão da disparidade é menor e isso acaba afetando muito o rastreamento. Uma vez que este depende muito da qualidade da detecção e quando ela falha isso acaba afetando o sistema como um todo.

Figura 22 – Imagens da sequência D também com três pessoas. Os pontos coloridos agrupados correspondem à saída da etapa de detecção, o retângulo ao redor corresponde à saída da etapa de rastreamento



Fonte: Elaborada pelo autor.

Nessa sequência, em vários frames ocorreu falha na detecção, onde duas pessoas foram agrupadas em um único objeto, isso fez com que vários *tracklets* se perdessem. Um exemplo desse erro pode ser visto na Figura 24c. É possível ver que a fonte desse erro é o mapa de custos. Vale ressaltar aqui que a imagem é amostrada reduzindo-a em 10 vezes. Apesar da configuração da disposição dos objetos ser clara na imagem completa, na imagem amostrada essa diferença é



menor. No entanto, como o intuito principal do projeto é a segurança dos obstáculos. Nesse caso apesar de ser detectado como um objeto maior, mesmo assim o carro identificou o obstáculo, a dimensão do objeto compreende as duas pessoas e dessa forma o veículo não colidiria. Outro erro que ocorreu nessa sequência foi a troca de ID entre dois *tracklets*, que pode ser visto nas Figuras 24d e 24e. Os resultados quantitativos estão disponíveis na Tabela 5.

Os resultados qualitativos da detecção e rastreamento podem ser vistos nas imagens disponibilizadas nesse trabalho e também por meio de todas as imagens de todas sequências disponíveis em <[https://youtu.be/Np-sH2t\\_2yU](https://youtu.be/Np-sH2t_2yU)>.

Tabela 5 – Resultados quantitativos da avaliação da sequência D.

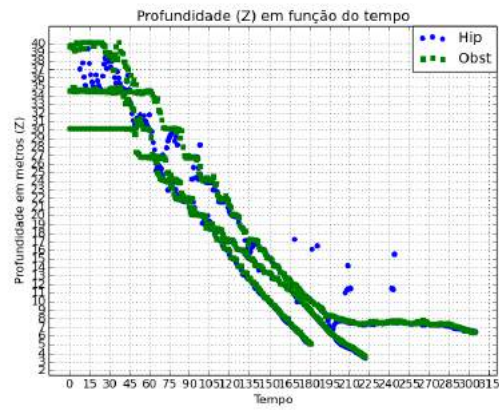
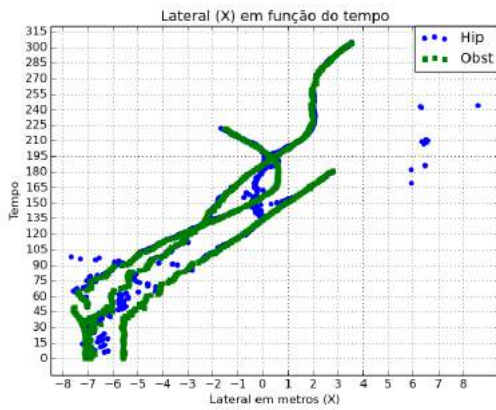
Métricas Rastreamento		Métricas Detecção	
MOTA	0,252113	Recall	0,497183
MOTP	0,799258	Precision	0,671103
MT	0,000000	F1	0,571197
PT	1,000000	FA	0,517964
ML	0,000000	Número de verdadeiro positivos	353
Troca de Ids	1	Número de falso positivos	173
Fragmentações	28	Alvos perdidos	357
Número de trajetórias no GT	3	Número de objetos no GT	710
Número de trajetórias pelo método	13	Número de objetos detectados pelo método	526

### 5.3 Considerações Finais

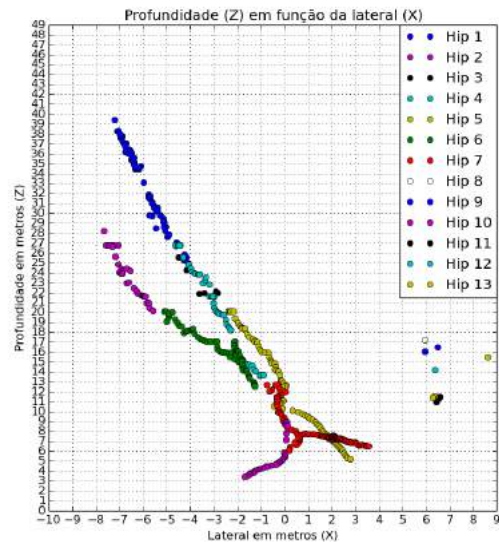
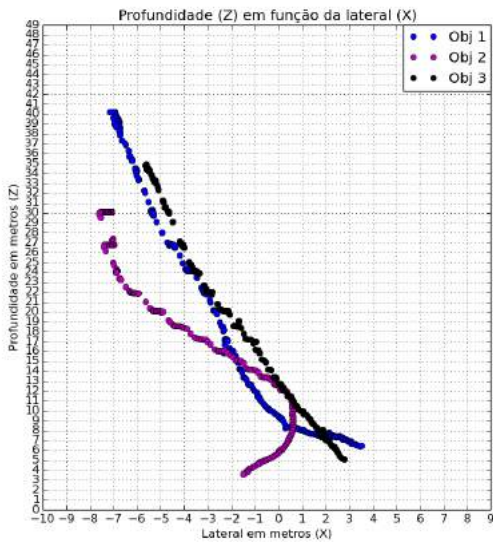
Este Capítulo apresenta os resultados obtidos pela abordagem proposta. Foi observado que o método de rastreamento depende muito da qualidade do módulo de detecção e quando este erra o sistema como um todo é prejudicado. Apesar de ser uma abordagem simples, o método proposto mostrou bons resultados na maioria das sequências. Foram apresentados dados quantitativos e qualitativos dos resultados dos experimentos feitos, onde é possível ver o método detectando e rastreando os obstáculos. A etapa de rastreamento só considera um obstáculo a partir do instante que ele é detectado pela segunda vez. Isso é feito para eliminar ruídos que esporadicamente aparecem nas imagens. Essa condição poderia ser eliminada com a construção de um filtro mais inteligente entre as etapas de detecção e rastreamento, que entregasse somente objetos alvo para a etapa de rastreamento, eliminando assim os postes esporadicamente detectados nas imagens e tornando possível o teste do método em ambientes mais complexos.

Figura 23 – Gráficos comparando o resultado da sequência D com o *Ground Truth*

- (a) Gráfico comparando a variação de X ao longo dos frames (b) Gráfico comparando a variação de Z ao longo dos frames



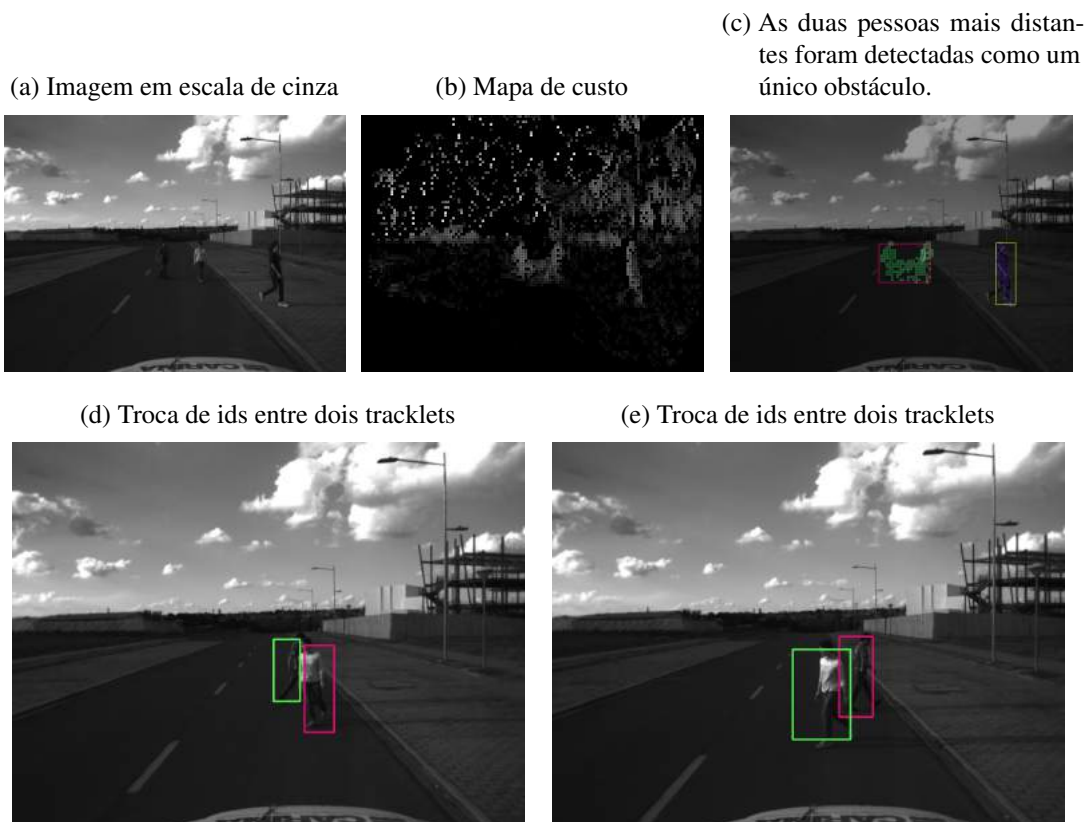
- (c) Gráfico comparando a variação de X e Z no GT (d) Gráfico comparando a variação de X e Z no método



Fonte: Elaborada pelo autor.



Figura 24 – Erros na etapa de detecção da sequência D. Nas Figuras 24a, 24b e 24c é exemplificado um caso de erro, onde dois pedestres foram detectados como um único obstáculo. Isso ocorre porque como é possível ver no mapa de custo Figura 24b, a pequena região que separa as duas pessoas teve um valor de custo mais próximo de um obstáculo, do que uma área navegável, portanto no agrupamento os dois obstáculos acabaram sendo agrupados em um mesmo grupo, como pode ser visto na Figura 24c. Já nas Figuras 24d e 24e ocorreu a troca de IDs entre dois tracklets distintos.



Fonte: Elaborada pelo autor.



---

## CONCLUSÃO

---

Veículos autônomos podem prover mobilidade a pessoas com necessidades especiais, diminuir congestionamentos, melhorar o uso de combustível, diminuir o número de acidentes e com isso salvar vidas. Para que um veículo autônomo possa trafegar nas ruas juntamente com outros carros, pessoas, ciclistas ou qualquer outro tipo de objeto, faz-se necessário o desenvolvimento de algoritmos que detectem esses objetos e que rastreiem suas trajetórias de forma a evitar qualquer tipo de colisão.

O presente trabalho apresentou o desenvolvimento de um sistema de detecção e rastreamento de obstáculos genéricos utilizando mapas de disparidade esparsos. Dentre as suas aplicações, destaca-se sua utilidade no projeto CaRINA, de forma a garantir a segurança de outros agentes operando no mesmo meio que o veículo autônomo. Para isso foi utilizado um sistema estereoscópico como meio de percepção do ambiente. A escolha desse sistema se deu pelo fato dos custos de câmeras serem menores quando comparados a outros sensores, como os sensores LIDAR Velodyne HDL-64E e HDL-32E, e pelo fato de câmeras fornecerem informações mais completas e detalhadas do ambiente. Embora ainda existam alguns problemas com cenários complexos, os resultados obtidos são bastante promissores, dada a simplicidade da abordagem, em comparação a métodos que usam aprendizado supervisionado ou requerem informações do ambiente.

A contribuição principal desse trabalho é um *framework* que utiliza visão estéreo e que combina detecção baseada em disparidade esparsa e rastreamento em 3D. Todo o sistema foi desenvolvido de modo que a substituição de qualquer um dos módulos é trivial, facilitando assim o teste de diferentes algoritmos em conjunto com os outros módulos. As vantagens principais desse método são: uso de um mapa de disparidade esparsa, reduzindo o tempo de computação, assim permitindo um pipeline completo executar eficientemente. Foi utilizada exclusivamente informação visual de uma câmera estéreo. Finalmente, essa abordagem não requer nenhum treinamento prévio e minimiza suposições sobre o ambiente, como a planaridade

das superfícies.

Os resultados obtidos são promissores e possibilitam a realização de vários trabalhos futuros, como por exemplo o desenvolvimento de algoritmos de predição de colisão mais inteligentes que regulem a velocidade ótima do veículo de forma a evitar colisões, desenvolvimento de algoritmos para realizar a ultrapassagem de obstáculos e realizar a análise semântica de cenas. As estimativas aqui feitas assumem que o veículo e os obstáculos possuem trajetórias lineares. Como trabalho futuro também está a transição do sistema para utilizar trajetórias não-lineares. Além disso, como possíveis melhorias podemos citar o uso de detecção de espaço livre com o intuito de rastrear somente obstáculos que estão na rua ou muito próximos a ela e realizar testes com o sistema *online* para prever colisões.

## REFERÊNCIAS

---

- BADINO, H.; FRANKE, U.; PFEIFFER, D. The stixel world - a compact medium level representation of the 3d-world. In: DENZLER, J.; NOTNI, G.; SüßE, H. (Ed.). **Pattern Recognition**. Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5748). p. 51–60. ISBN 978-3-642-03797-9. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-03798-6\\_6](http://dx.doi.org/10.1007/978-3-642-03798-6_6)>. 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: **Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on**. [S.l.: s.n.], 2012. p. 2903–2910. ISSN 1063-6919. Citado na página 30.
- BERNARDIN, K.; STIEFELHAGEN, R. Evaluating multiple object tracking performance: The clear mot metrics. **EURASIP Journal on Image and Video Processing**, v. 2008, n. 1, p. 1–10, 2008. ISSN 1687-5281. Disponível em: <<http://dx.doi.org/10.1155/2008/246309>>. Citado 3 vezes nas páginas 52, 54 e 55.
- BERNINI, N.; BERTOZZI, M.; CASTANGIA, L.; PATANDER, M.; SABBATELLI, M. Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In: **Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on**. [S.l.: s.n.], 2014. p. 873–878. Citado na página 28.
- BEWLEY, A.; GUIZILINI, V.; RAMOS, F.; UPCROFT, B. Online self-supervised multi-instance segmentation of dynamic objects. In: **Robotics and Automation (ICRA), 2014 IEEE International Conference on**. [S.l.: s.n.], 2014. p. 1296–1303. Citado na página 29.
- BICHSEL, R.; BORGES, P. Discrete-continuous clustering for obstacle detection using stereo vision. In: **Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on**. [S.l.: s.n.], 2015. p. 538–545. Citado na página 28.
- BRADSKI, G. R.; KAEHLER, A. Learning opencv - computer vision with the opencv library: software that sees. In: . [S.l.: s.n.], 2008. p. I–XVII, 1–555. Citado 3 vezes nas páginas 33, 35 e 36.
- CHANG, W.-C.; CHO, C.-W. Online boosting for vehicle detection. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on**, v. 40, n. 3, p. 892–902, June 2010. ISSN 1083-4419. Citado na página 27.
- CHOI, W. Near-online multi-target tracking with aggregated local flow descriptor. **ICCV**, 2015. Citado na página 30.
- DANESCU, R.; ONIGA, F.; NEDEVSCI, S. Modeling and tracking the driving environment with a particle-based occupancy grid. **Intelligent Transportation Systems, IEEE Transactions on**, v. 12, n. 4, p. 1331–1342, Dec 2011. ISSN 1524-9050. Citado na página 28.
- ERBS, F.; WITTE, A.; SCHARWAECHTER, T.; MESTER, R.; FRANKE, U. Spider-based stixel object segmentation. In: **Intelligent Vehicles Symposium Proceedings, 2014 IEEE**. [S.l.: s.n.], 2014. p. 906–911. Citado na página 30.

ESTER, M.; KRIEGEL, H.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, Oregon, USA.** [s.n.], 1996. p. 226–231. Disponível em: <<http://www.aaai.org/Library/KDD/1996/kdd96-037.php>>. Citado na página 30.

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**, North-Holland, v. 60, n. 4, p. 372–392, 2014. Citado na página 38.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. **Commun. ACM**, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/358669.358692>>. Citado na página 29.

FRANKE, U.; RABE, C.; BADINO, H.; GEHRIG, S. 6d-vision: Fusion of stereo and motion for robust environment perception. In: KROPATSCH, W.; SABLATNIG, R.; HANBURY, A. (Ed.). **Pattern Recognition**. Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3663). p. 216–223. ISBN 978-3-540-28703-2. Disponível em: <[http://dx.doi.org/10.1007/11550518\\_27](http://dx.doi.org/10.1007/11550518_27)>. Citado na página 28.

GEIGER, A.; LENZ, P.; URTASUN, R. Are we ready for autonomous driving? the kitti vision benchmark suite. In: **Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2012. Citado 4 vezes nas páginas 11, 13, 49 e 52.

GEIGER, A.; ROSER, M.; URTASUN, R. Efficient large-scale stereo matching. In: **Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision, Queenstown, New Zealand, November 8-12, 2010, Revised Selected Papers, Part I**. [s.n.], 2010. p. 25–38. Disponível em: <[http://dx.doi.org/10.1007/978-3-642-19315-6\\_3](http://dx.doi.org/10.1007/978-3-642-19315-6_3)>. Citado na página 34.

GEIGER, A.; ZIEGLER, J.; STILLER, C. Stereoscan: Dense 3d reconstruction in real-time. In: **Intelligent Vehicles Symposium (IV)**. [S.l.: s.n.], 2011. Citado na página 85.

GEIGER, M. R. A.; URTASUN, R. Efficient large-scale stereo matching. In: **Computer Vision - ACCV 2010 - 10th Asian Conference on Computer Vision**. [S.l.: s.n.], 2010. p. 25–38. Citado na página 41.

GUIZILINI, V.; RAMOS, F. Online self-supervised segmentation of dynamic objects. In: **2013 IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, May 6-10, 2013**. [s.n.], 2013. p. 4720–4727. Disponível em: <<http://dx.doi.org/10.1109/ICRA.2013.6631249>>. Citado na página 29.

HIMMELSBACH, M.; WUENSCH, H.-J. Tracking and classification of arbitrary objects with bottom-up/top-down detection. In: **Intelligent Vehicles Symposium (IV), 2012 IEEE**. [S.l.: s.n.], 2012. p. 577–582. ISSN 1931-0587. Citado na página 27.

HIRSCHMULLER, H. Accurate and efficient stereo processing by semi-global matching and mutual information. In: **Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on**. [S.l.: s.n.], 2005. v. 2, p. 807–814 vol. 2. ISSN 1063-6919. Citado na página 29.

- HORN, B. K. P.; SCHUNCK, B. G. Determining optical flow. **Artif. Intell.**, v. 17, n. 1-3, p. 185–203, 1981. Citado na página 44.
- HUANG, Y.; ZHUANG, S. Motion-partitioned adaptive block matching for video compression. In: **Image Processing, 1995. Proceedings., International Conference on**. [S.l.: s.n.], 1995. v. 1, p. 554–557 vol.1. Citado na página 34.
- KALAL, Z.; MIKOLAJCZYK, K.; MATAS, J. Forward-backward error: Automatic detection of tracking failures. In: **ICPR**. [S.l.: s.n.], 2010. p. 2756–2759. Citado 3 vezes nas páginas 40, 44 e 45.
- \_\_\_\_\_. Tracking-learning-detection. **IEEE Trans. Pattern Anal. Mach. Intell.**, v. 34, n. 7, p. 1409–1422, 2012. Citado na página 44.
- KITT, B.; GEIGER, A.; LATEGAHN, H. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In: **Intelligent Vehicles Symposium (IV), 2010 IEEE**. [S.l.: s.n.], 2010. p. 486–492. ISSN 1931-0587. Citado na página 29.
- KONOLIGE, K. Small vision systems: Hardware and implementation. In: SHIRAI, Y.; HIROSE, S. (Ed.). **Robotics Research**. Springer London, 1998. p. 203–212. ISBN 978-1-4471-1582-3. Disponível em: <[http://dx.doi.org/10.1007/978-1-4471-1580-9\\_19](http://dx.doi.org/10.1007/978-1-4471-1580-9_19)>. Citado na página 34.
- KRAMM, S.; BENSRAHAI, A. Obstacle detection using sparse stereovision and clustering techniques. In: **Intelligent Vehicles Symposium (IV), 2012 IEEE**. [S.l.: s.n.], 2012. p. 760–765. ISSN 1931-0587. Citado na página 30.
- KUHN, H. W. The hungarian method for the assignment problem. **Naval Research Logistics Quarterly**, v. 2, p. 83–97, 1955. Citado na página 48.
- LEE, D.; SCHACHTER, B. Two algorithms for constructing a delaunay triangulation. **International Journal of Computer and Information Sciences**, Kluwer Academic Publishers-Plenum Publishers, v. 9, n. 3, p. 219–242, 1980. ISSN 0091-7036. Disponível em: <<http://dx.doi.org/10.1007/BF00977785>>. Citado na página 41.
- LEFÈVRE, D. V. S.; LAUGIER, C. A survey on motion prediction and risk assessment for intelligent vehicles. In: **ROBOMECH Journal, Springer**. [S.l.: s.n.], 2014. p. 1. Citado na página 83.
- LENZ, P.; GEIGER, A.; URTASUN, R. Followme: Efficient online min-cost flow tracking with bounded memory and computation. In: **International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. Citado na página 30.
- LENZ, P.; ZIEGLER, J.; GEIGER, A.; ROSER, M. Sparse scene flow segmentation for moving object detection in urban environments. In: **IEEE Intelligent Vehicles Symposium (IV), 2011, Baden-Baden, Germany, June 5-9, 2011**. [s.n.], 2011. p. 926–932. Disponível em: <<http://dx.doi.org/10.1109/IVS.2011.5940558>>. Citado na página 28.
- LEWIS, J. Fast normalized cross-correlation. In: **Vision interface**. [S.l.: s.n.], 1995. v. 10, n. 1, p. 120–123. Citado na página 44.
- LIN, B.-F.; CHAN, Y.-M.; FU, L.-C.; HSIAO, P.-Y.; CHUANG, L.-A.; HUANG, S.-S.; LO, M.-F. Integrating appearance and edge features for sedan vehicle detection in the blind-spot area. **Intelligent Transportation Systems, IEEE Transactions on**, v. 13, n. 2, p. 737–747, June 2012. ISSN 1524-9050. Citado na página 27.

LOWE, D. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 60, n. 2, p. 91–110, 2004. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1023/B%3AVISI.0000029664.99615.94>>. Citado 2 vezes nas páginas 28 e 29.

MATTHIES, L.; KANADE, T.; SZELISKI, R. Kalman filter-based algorithms for estimating depth from image sequences. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 3, n. 3, p. 209–238, 1989. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1007/BF00133032>>. Citado na página 34.

MILLER, R.; HUANG, Q. An adaptive peer-to-peer collision warning system. In: **Proceedings of the 55th IEEE Vehicular Technology Conference, VTC Spring 2002, May 6-9, 2002, Birmingham, Alabama, USA**. [s.n.], 2002. p. 317–321. Disponível em: <<http://dx.doi.org/10.1109/VTC.2002.1002718>>. Citado 2 vezes nas páginas 83 e 84.

MUNKRES, J. Algorithms for the assignment and transportation problems. **Journal of the Society of Industrial and Applied Mathematics**, v. 5, n. 1, p. 32–38, March 1957. Citado 5 vezes nas páginas 11, 13, 39, 44 e 48.

MüHLMANN, K.; MAIER, D.; HESSER, J.; MÄNNER, R. Calculating dense disparity maps from color stereo images, an efficient implementation. **International Journal of Computer Vision**, Kluwer Academic Publishers, v. 47, n. 1-3, p. 79–88, 2002. ISSN 0920-5691. Disponível em: <<http://dx.doi.org/10.1023/A%3A1014581421794>>. Citado na página 34.

NIKNEJAD, H.; TAKEUCHI, A.; MITA, S.; MCALLESTER, D. On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation. **Intelligent Transportation Systems, IEEE Transactions on**, v. 13, n. 2, p. 748–758, June 2012. ISSN 1524-9050. Citado na página 27.

OHN-BAR, E.; SIVARAMAN, S.; TRIVEDI, M. Partially occluded vehicle recognition and tracking in 3d. In: **Intelligent Vehicles Symposium (IV), 2013 IEEE**. [S.l.: s.n.], 2013. p. 1350–1355. ISSN 1931-0587. Citado na página 31.

ONIGA, F.; NEDEVSKI, S. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. **IEEE Transactions on Vehicular Technology**, v. 59, n. 3, p. 1172–1182, March 2010. ISSN 0018-9545. Citado na página 30.

PFEIFFER, D.; FRANKE, U. Efficient representation of traffic scenes by means of dynamic stixels. In: **IEEE Intelligent Vehicles Symposium (IV), 2010, La Jolla, CA, USA, June 21-24, 2010**. [s.n.], 2010. p. 217–224. Disponível em: <<http://dx.doi.org/10.1109/IVS.2010.5548114>>. Citado na página 30.

\_\_\_\_\_. Towards a global optimal multi-layer stixel representation of dense 3d data. In: **British Machine Vision Conference, BMVC 2011, Dundee, UK, August 29 - September 2, 2011. Proceedings**. [s.n.], 2011. p. 1–12. Disponível em: <<http://dx.doi.org/10.5244/C.25.51>>. Citado na página 30.

PINGGERA, P.; FRANKE, U.; MESTER, R. High-performance long range obstacle detection using stereo vision. In: **Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on**. [S.l.: s.n.], 2015. p. 1308–1313. Citado na página 30.



PREMEBIDA, C.; MONTEIRO, G.; NUNES, U.; PEIXOTO, P. A lidar and vision-based approach for pedestrian and vehicle detection and tracking. In: **2007 IEEE Intelligent Transportation Systems Conference**. [S.l.: s.n.], 2007. p. 1044–1049. ISSN 2153-0009. Citado na página 27.

RASMUSSEN, C. E. Gaussian processes for machine learning. In: . [S.l.]: MIT Press, 2006. Citado na página 29.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach (2nd edition)**. [S.l.]: Prentice Hall, 2004. Citado na página 34.

RZIZA, M.; TAMTAOUI, A.; MORIN, L.; ABOUTAJDINE, D. Estimation and segmentation of a dense disparity map for 3d reconstruction. In: **IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2000, 5-9 June, 2000, Hilton Hotel and Convention Center, Istanbul, Turkey**. [s.n.], 2000. p. 2219–2222. Disponível em: <<http://dx.doi.org/10.1109/ICASSP.2000.859279>>. Citado na página 34.

SANTOS, T. C. Uma proposta de arquitetura de software para a simulação e experimentação de veículos autônomos, dissertação de mestrado, instituto de ciências da computação e matemática (icmc), universidade de são paulo, são carlos. In: . [S.l.: s.n.], 2015. Citado na página 37.

SHI, J.; TOMASI, C. Good features to track. In: **Computer Vision and Pattern Recognition, 1994. Proceedings CVPR '94., 1994 IEEE Computer Society Conference on**. [S.l.: s.n.], 1994. p. 593–600. ISSN 1063-6919. Citado na página 29.

SHINZATO, P. Y. Estimation of obstacles and road area with sparse 3d points, doctoral thesis, institute of mathematics and computer science (icmc), university of sao paulo, sao carlos. In: . [S.l.: s.n.], 2015. Citado 2 vezes nas páginas 41 e 42.

SHINZATO, P. Y.; GOMES, D.; WOLF, D. F. Road estimation with sparse 3d points from stereo data. In: **Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2014. p. 1688–1693. Citado 2 vezes nas páginas 41 e 42.

SIVARAMAN, S.; TRIVEDI, M. A general active-learning framework for on-road vehicle recognition and tracking. **Intelligent Transportation Systems, IEEE Transactions on**, v. 11, n. 2, p. 267–276, June 2010. ISSN 1524-9050. Citado na página 28.

SIVARAMAN, S.; TRIVEDI, M. M. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. **IEEE Transactions on Intelligent Transportation Systems**, v. 14, n. 4, p. 1773–1795, 2013. Disponível em: <<http://dx.doi.org/10.1109/TITS.2013.2266661>>. Citado 2 vezes nas páginas 27 e 28.

SKLANSKY, J. Finding the convex hull of a simple polygon. **Pattern Recogn. Lett.**, Elsevier Science Inc., New York, NY, USA, v. 1, n. 2, p. 79–83, dez. 1982. ISSN 0167-8655. Disponível em: <[http://dx.doi.org/10.1016/0167-8655\(82\)90016-2](http://dx.doi.org/10.1016/0167-8655(82)90016-2)>. Citado na página 50.

SUN, Z.; BEBIS, G.; MILLER, R. Monocular precrash vehicle detection: features and classifiers. **Image Processing, IEEE Transactions on**, v. 15, n. 7, p. 2019–2034, July 2006. ISSN 1057-7149. Citado 2 vezes nas páginas 27 e 28.

THRUN, S.; MONTEMERLO, M.; DAHLKAMP, H.; STAVENS, D.; ARON, A.; DIEBEL, J.; FONG, P.; GALE, J.; HALPENNY, M.; HOFFMANN, G.; LAU, K.; OAKLEY, C. M.; PALATUCCI, M.; PRATT, V.; STANG, P.; STROHBAND, S.; DUPONT, C.; JENDROSSEK,

L.-E.; KOELEN, C.; MARKEY, C.; RUMMEL, C.; NIEKERK, J. van; JENSEN, E.; ALESSANDRINI, P.; BRADSKI, G. R.; DAVIES, B.; ETTINGER, S.; KAEHLER, A.; NEFIAN, A. V.; MAHONEY, P. Stanley: The robot that won the darpa grand challenge. **J. Field Robotics**, v. 23, n. 9, p. 661–692, 2006. Citado na página 23.

URMSON, C.; ANHALT, J.; BAGNELL, D.; BAKER, C. R.; BITTNER, R.; CLARK, M. N.; DOLAN, J. M.; DUGGINS, D.; GALATALI, T.; GEYER, C.; GITTLEMAN, M.; HARBAUGH, S.; HEBERT, M.; HOWARD, T. M.; KOLSKI, S.; KELLY, A.; LIKHACHEV, M.; MCNAUGHTON, M.; MILLER, N.; PETERSON, K. M.; PILNICK, B.; RAJKUMAR, R.; RYBSKI, P. E.; SALESKY, B.; SEO, Y.-W.; SINGH, S.; SNIDER, J. M.; STENTZ, A.; WHITTAKER, W.; WOLKOWICKI, Z.; ZIGLAR, J.; BAE, H.; BROWN, T.; DEMITRISH, D.; LITKOUHI, B.; NICKOLAOU, J.; SADEKAR, V.; ZHANG, W.; STRUBLE, J.; TAYLOR, M.; DARMS, M.; FERGUSON, D. Autonomous driving in urban environments: Boss and the urban challenge. In: **The DARPA Urban Challenge**. [S.l.: s.n.], 2009. p. 1–59. Citado na página 23.

WEDEL, A.; FRANKE, U.; BADINO, H.; CREMERS, D. B-spline modeling of road surfaces for freespace estimation. In: **Intelligent Vehicles Symposium, 2008 IEEE**. [S.l.: s.n.], 2008. p. 828–833. ISSN 1931-0587. Citado na página 29.

WELCH, G.; BISHOP, G. **An Introduction to the Kalman Filter**. 1995. Citado 4 vezes nas páginas 11, 13, 44 e 45.

WHO. Global status report on road safety 2015. In: . [S.l.: s.n.], 2015. p. 323. Citado 2 vezes nas páginas 11 e 13.

WOJKE, N.; HÄSELICH, M. Moving vehicle detection and tracking in unstructured environments. In: **ICRA**. [S.l.: s.n.], 2012. p. 3082–3087. Citado na página 27.

XIANG, Y.; ALAHI, A.; SAVARESE, S. Learning to track: Online multi-object tracking by decision making. In: **The IEEE International Conference on Computer Vision (ICCV)**. [S.l.: s.n.], 2015. Citado na página 31.

YUAN, Q.; THANGALI, A.; ABLAVSKY, V.; SCLAROFF, S. Learning a family of detectors via multiplicative kernels. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 33, n. 3, p. 514–530, March 2011. ISSN 0162-8828. Citado 2 vezes nas páginas 27 e 28.

ZHANG, L.; LI, Y.; NEVATIA, R. Global data association for multi-object tracking using network flows. In: **2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA**. [s.n.], 2008. Disponível em: <<http://dx.doi.org/10.1109/CVPR.2008.4587584>>. Citado 3 vezes nas páginas 52, 54 e 55.

ZIEGLER, J.; BENDER, P.; SCHREIBER, M.; LATEGAHN, H.; STRAUSS, T.; STILLER, C.; DANG, T.; FRANKE, U.; APPENRODT, N.; KELLER, C. G.; KAUS, E.; HERRTWICH, R. G.; RABE, C.; PFEIFFER, D.; LINDNER, F.; STEIN, F.; ERBS, F.; ENZWEILER, M.; KNÖPPEL, C.; HIPPE, J.; HAUEIS, M.; TREPTE, M.; BRENK, C.; TAMKE, A.; GHANAAT, M.; BRAUN, M.; JOOS, A.; FRITZ, H.; MOCK, H.; HEIN, M.; ZEEB, E. Making bertha drive - an autonomous journey on a historic route. **IEEE Intell. Transport. Syst. Mag.**, v. 6, n. 2, p. 8–20, 2014. Disponível em: <<http://dx.doi.org/10.1109/MITS.2014.2306552>>. Citado na página 24.

---

## PUBLICAÇÕES

---

RIDEL, D. A. ; SHINZATO, P. Y. ; WOLF, D. F. . Obstacle segmentation with low-density disparity maps. In: *International Conference on Intelligent Robots and Systems (IROS)*, 2015, Hamburgo. 7th Workshop on Planning, Perception and Navigation for Intelligent Vehicles, 2015.

RIDEL, D. A. ; SHINZATO, P. Y. ; WOLF, D. F. . Clustering-Based Obstacle Segmentation Approach for Urban Environments. In: *Latin American Robotics Symposium (LARS)*, 2015, Uberlândia. Latin American Robotics Symposium, 2015.

ALENCAR, F. A. R. ; MASSERA FILHO, C. ; RIDEL, D. A. ; WOLF, D. F. . Fast Metric Multi-Object Vehicle Tracking for Dynamical Environment Comprehension. In: *Latin American Robotics Symposium (LARS)*, 2015, Uberlândia. Latin American Robotics Symposium, 2015.



---

## PREDIÇÃO DE COLISÃO

---

Um dos maiores desafios em sistemas de veículos inteligentes é a previsão de possíveis colisões futuras. Essa estimativa visa prover informações para o veículo autônomo de forma que ele consiga executar uma ação, seja ela parar ou desviar de um obstáculo, evitando qualquer tipo de acidente. Para evitar uma colisão é necessário realizar a estimação de um modelo de movimento para conseguir prever onde o carro e o obstáculo estarão no tempo seguinte, e com esse modelo estimar a chance de uma possível colisão. As abordagens de modelagem de movimento e predição podem ser divididos em três grupos, conforme proposto em (LEFÈVRE; LAUGIER, 2014):

- Movimentação baseada na física;
- Movimentação baseada em manobras e
- Movimentação baseada na interação consciente.

Na primeira são considerados modelos mais simples, onde a movimentação dos veículos só dependem das leis da física. Na segunda é considerado que a movimentação futura do veículo também depende da manobra que o motorista pretende fazer. E a terceira leva em consideração as interdependências entre as manobras dos veículos.

Modelos de movimentação baseado na física são os que permanecem como sendo geralmente os mais usados para predição de trajetória (LEFÈVRE; LAUGIER, 2014) e estimativa de risco de colisão. A movimentação futura é predita usando modelos dinâmicos e cinemáticos ligados a alguns controles de entradas, propriedades do carro e condições externas.

Nos próximos parágrafos será apresentada a abordagem proposta por (MILLER; HUANG, 2002), onde um ponto de colisão  $(x_+, y_+)$  é estimado baseado nas posições e orientações de dois objetos e em seguida é calculado o tempo (TTX) de cada um desses objetos atingirem esse ponto de colisão. Baseado nesses tempos de chegada um alerta é emitido para que alguma

medida seja tomada no intuito de evitar a colisão. O ponto de colisão entre os obstáculos (que podem ser o carro autônomo e outro obstáculo) é dado por:

$$x_+ = \frac{(y_{obst} - y_{car}) - (x_{obst} \tan \theta_{obst} - x_{car} \tan \theta_{car})}{\tan \theta_{car} - \tan \theta_{obst}} \quad (A.1)$$

$$y_+ = \frac{(x_{obst} - x_{car}) - (y_{obst} \cot \theta_{obst} - y_{car} \cot \theta_{car})}{\cot \theta_{car} - \cot \theta_{obst}} \quad (A.2)$$

onde  $x$  e  $y$  correspondem a posição e  $\theta_{car}$  e  $\theta_{obst}$  correspondem às orientações. No entanto, apesar de existir um ponto de colisão isso não quer dizer que o obstáculo e o carro irão realmente se chocar, pois isso depende do instante de tempo que ambos chegarão nesse ponto de intersecção. Se eles chegarem no mesmo instante de tempo, isso indica que uma colisão entre os dois ocorrerá, sendo necessário, portanto, estimar o tempo de chegada do veículo autônomo no ponto de intersecção, ou seja,  $TTX_{car}$  e o tempo para o obstáculo chegar no ponto de intersecção, ou seja,  $TTX_{obst}$ .

$$TTX_{car} = \frac{|\vec{r}_+ - \vec{r}_{car}|}{|\vec{v}_{car}|} \text{sign}((\vec{r}_+ - \vec{r}_{car}) \cdot \vec{v}_{car}) \quad (A.3)$$

$$TTX_{obst} = \frac{|\vec{r}_+ - \vec{r}_{obst}|}{|\vec{v}_{obst}|} \text{sign}((\vec{r}_+ - \vec{r}_{obst}) \cdot \vec{v}_{obst}) \quad (A.4)$$

onde  $\vec{v}_{car}$  e  $\vec{v}_{obst}$  correspondem às velocidades do carro autônomo e obstáculo, respectivamente, e  $\vec{r}_{car}$  e  $\vec{r}_{obst}$  é a representação em vetor das coordenadas  $(x,y)$ . Se  $TTX_{car} = TTX_{obst}$ , isso quer dizer que ambos chegarão ao mesmo tempo no ponto  $(x_+, y_+)$ , o que indica que uma colisão ocorrerá. Como o veículo e o obstáculo estão sendo representados por pontos é necessário utilizar um limiar que garanta que a colisão não ocorre em nenhuma parte da extensão desses agentes.

Para o caso onde o obstáculo vem de encontro ao carro, as seguintes fórmulas são propostas:

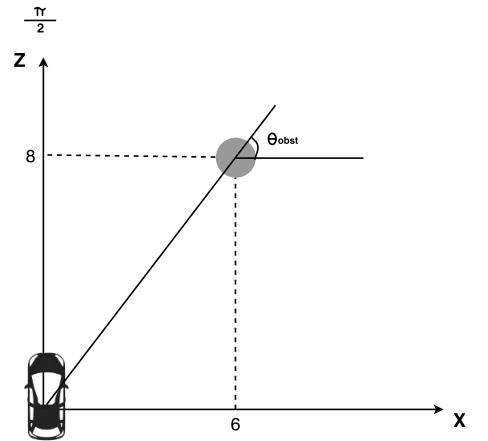
$$x_+ = x_{car} + \frac{|\vec{v}_{car}|}{|\vec{v}_{car} - \vec{v}_{obst}|} d \cos \theta_{car} \quad (A.5)$$

$$y_+ = y_{car} + \frac{|\vec{v}_{car}|}{|\vec{v}_{car} - \vec{v}_{obst}|} d \sin \theta_{car} \quad (A.6)$$

Com o intuito de validar nosso rastreamento de obstáculos, ter uma aplicação prática do sistema de prevenção de colisão em funcionamento, foi implementada a abordagem de (MILLER; HUANG, 2002) para calcular o ponto de colisão entre o carro autônomo e um obstáculo. A cada instante de tempo novas informações são calculadas pelos algoritmos de detecção e rastreamento, e o cálculo para estimar o tempo para ambos chegarem a um ponto de colisão é estimado. A Figura 25 exemplifica o sistema de coordenadas onde o carro sempre está na posição  $(0,0)$ . No entanto, como o algoritmo assume posições globais foi considerada a velocidade global do carro

(usando libviso (GEIGER; ZIEGLER; STILLER, 2011)) e a velocidade global do obstáculo (obtida através da adição da sua velocidade relativa ao carro com a velocidade do carro).

Figura 25 – Exemplo da configuração das coordenadas de um objeto detectado em relação ao carro.

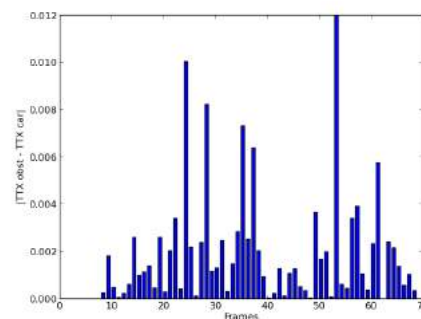
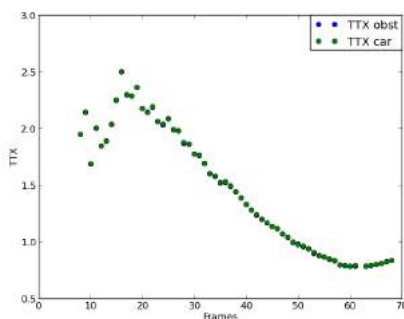


Fonte: Elaborada pelo autor.

Avaliamos a abordagem em duas sequências simples com um único obstáculo. Na primeira sequência um pedestre está na frente do carro e vem de encontro ao carro e na segunda o pedestre está a direita do carro e atravessa para a esquerda. Na primeira sequência (Figura 27), podemos ver pela Figura 26 que o TTX do carro e do obstáculo estão muito próximos, e esse tempo decresce a medida que o pedestre se aproxima do veículo.

Figura 26 – Avaliação dos TTX do carro e do obstáculo na primeira sequência.

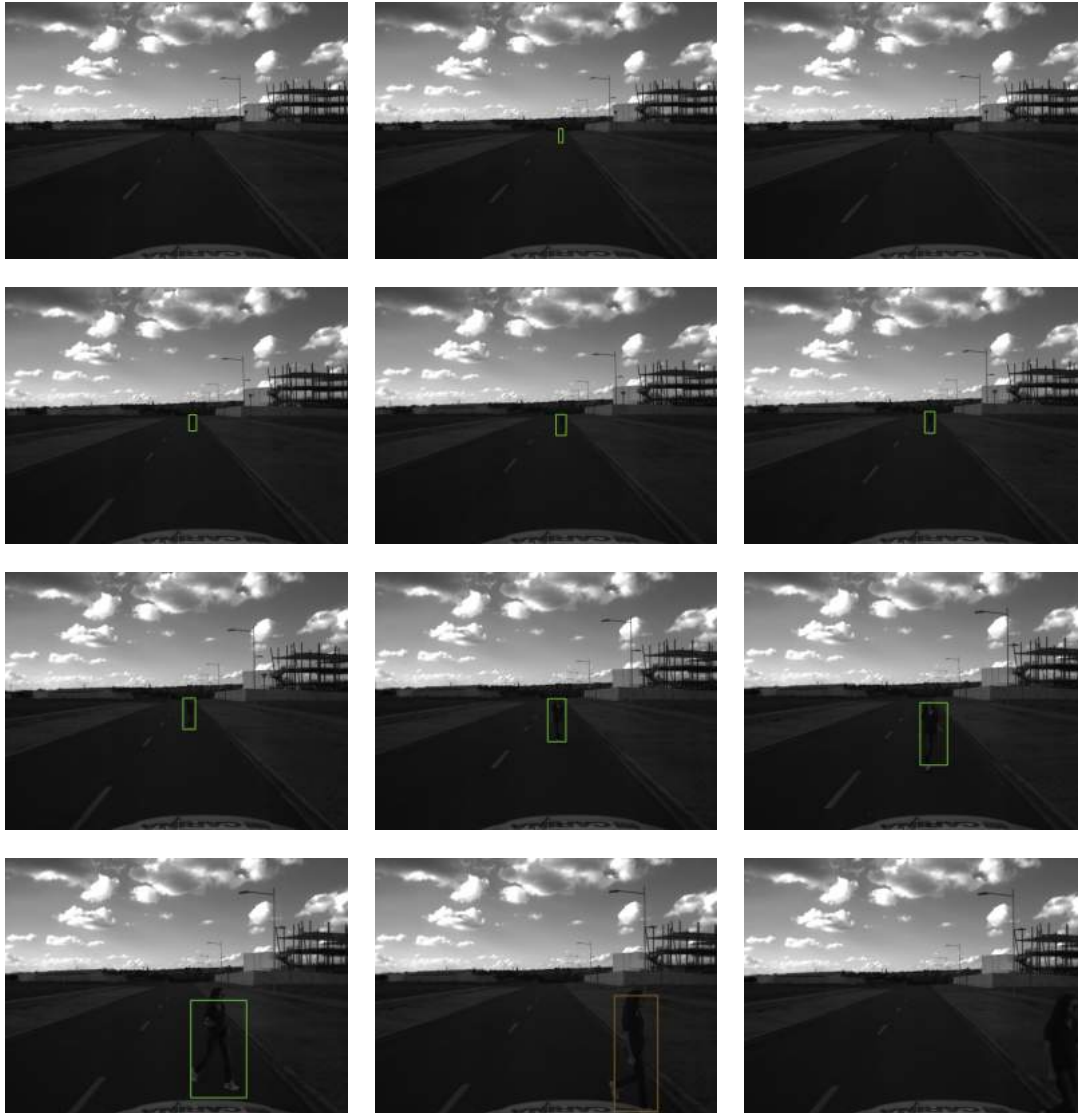
- (a) Podemos notar que o tempo para colisão do pedestre e do carro são muito parecidos para esse caso.
- (b) Gráfico representando as diferenças entre os TTX do carro e do obstáculo. A diferença entre os tempos para colisão são muito pequenas, a maior diferença é de 0,012 segundos.



Fonte: Elaborada pelo autor.

Na segunda sequência o pedestre está do lado esquerdo da calçada e atravessa para o lado direito. A sequência já começa com o pedestre andando e termina quando o obstáculo some à direita da imagem. Um exemplo dessa sequência pode ser visto na Figura 28.

Figura 27 – Imagens da primeira sequência, onde um pedestre vem de encontro ao carro autônomo. O retângulo ao redor corresponde à saída da etapa de rastreamento.

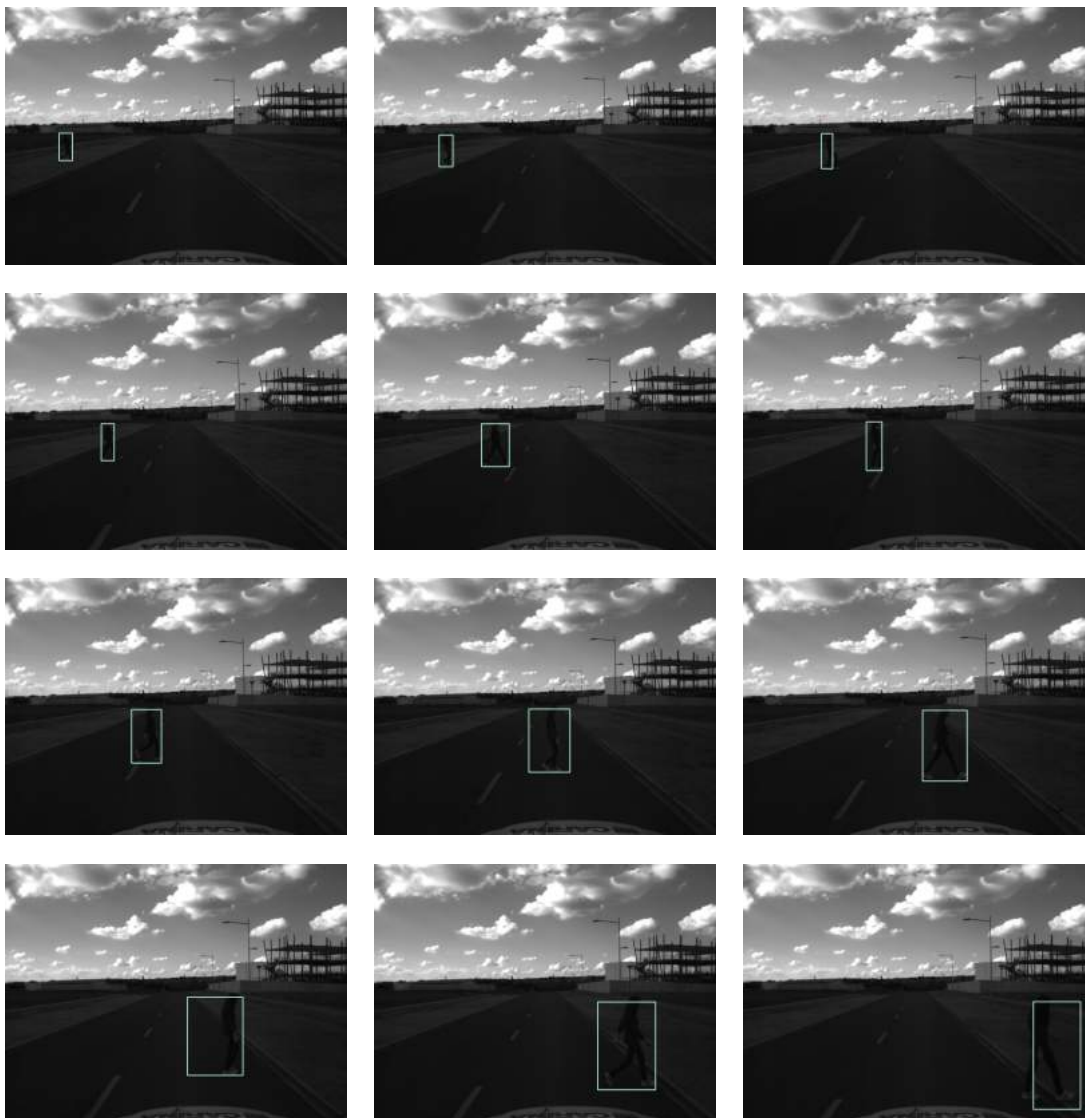


Fonte: Elaborada pelo autor.

Além da discussão ao redor dos valores de TTX, também foram feitos vídeos (Figura 30) ilustrando a sequência, o ponto de interseção e os valores de TTX para o carro e o obstáculo <<https://youtu.be/8uF2iawbKkU>>.



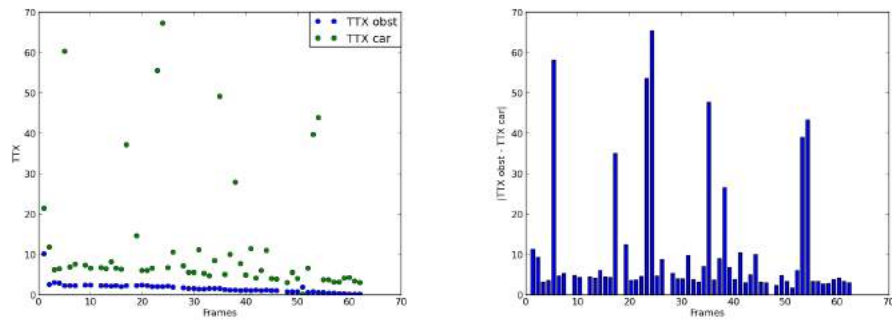
Figura 28 – Imagens da segunda seqüência, onde um pedestre atravessa do lado esquerdo da calçada para o lado direito.



Fonte: Elaborada pelo autor.

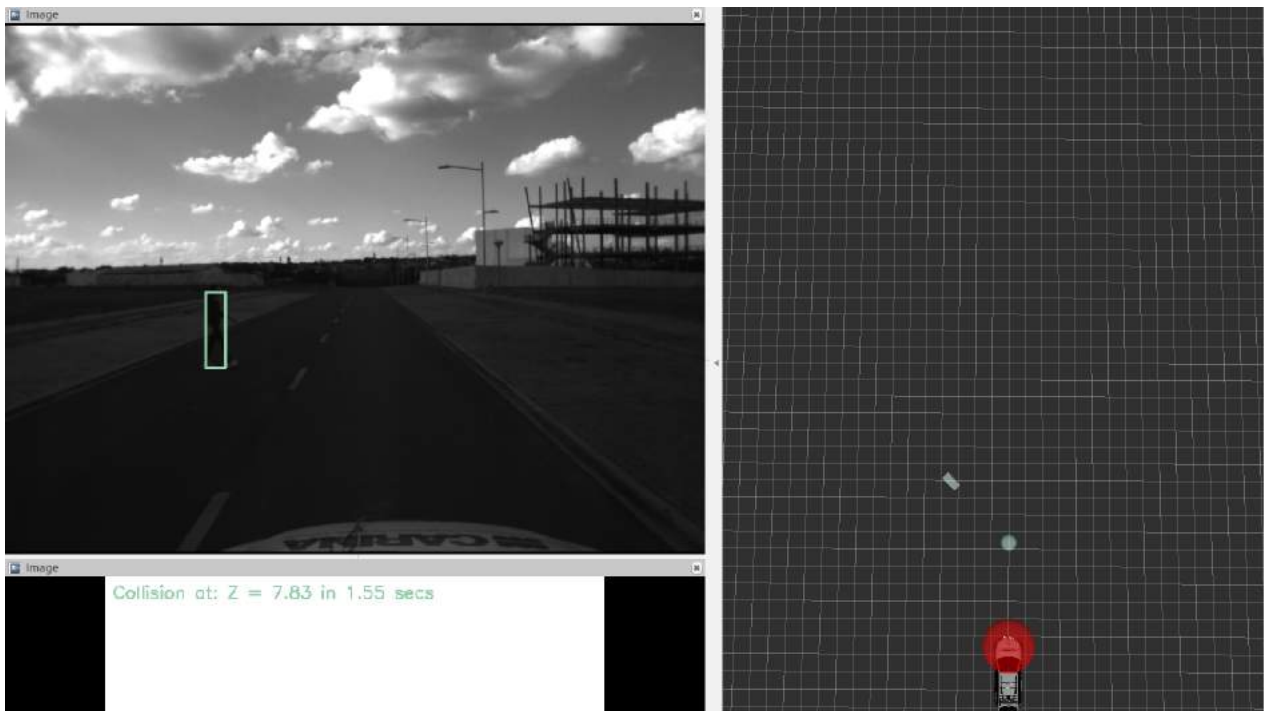
Figura 29 – Avaliação dos TTX do carro e do obstáculo na segunda sequência.

- (a) Podemos notar que o tempo para colisão do pedestre e do carro são diferentes nesse caso.
- (b) Gráfico representando as diferenças entre os TTX do carro e do obstáculo. Aqui a diferença dos tempos para colisão são maiores.



Fonte: Elaborada pelo autor.

Figura 30 – Exemplo da estimativa de colisão.



Fonte: Elaborada pelo autor.