UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Extração de features 3D para o reconhecimento de objetos em nuvem de pontos

Daniel Oliva Sales

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Daniel Oliva Sales

Extração de features 3D para o reconhecimento de objetos em nuvem de pontos

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

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Fernando Santos Osório

USP – São Carlos
Dezembro de 2017

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

S163e Sales, Daniel Oliva
Extração de features 3D para o reconhecimento de
objetos em nuvem de pontos / Daniel Oliva Sales;
orientador Fernando Santos Osório. -- São Carlos,
2017.
86 p.

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

1. Robótica Móvel. 2. Visão Computacional. 3.
Extração de features robustas. 4. Reconhecimento de
Padrões. I. Osório, Fernando Santos, orient. II.
Título.

Daniel Oliva Sales

3D feature extraction for objects recognition in point clouds

Doctoral dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Fernando Santos Osório

USP – São Carlos
December 2017

AGRADECIMENTOS

À minha família, especialmente meus pais e avós, pelo constante apoio, motivação e carinho em todos os momentos.

Ao meu orientador Fernando Santos Osório, pela confiança depositada em mim, pelo exemplo e conhecimento compartilhados que contribuíram tanto para minha formação acadêmica como crescimento pessoal.

Aos meus amigos e colegas de laboratório, pelas idéias compartilhadas e parceria nos trabalhos e artigos ao longo do doutorado.

Agradeço ainda ao CNPq, pela bolsa de doutorado concedida e aos demais órgãos de fomento à pesquisa, onde destacam-se a FAPESP e o INCT-SEC pela contribuição para a aquisição dos equipamentos utilizados durante o desenvolvimento dessa pesquisa.

RESUMO

SALES, D. O. **Extração de features 3D para o reconhecimento de objetos em nuvem de pontos**. 2017. 86 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

A detecção e reconhecimento de objetos é uma tarefa fundamental em aplicações relacionadas à navegação autônoma de robôs móveis e veículos inteligentes. Com a evolução tecnológica nos sistemas sensoriais, surgiram equipamentos capazes de detectar e representar os elementos presentes no ambiente de forma tridimensional, em estruturas chamadas nuvem de pontos. Os sensores 3D geralmente capturam um grande volume de pontos em curtos intervalos de tempo, o que demanda técnicas robustas para processamento dessa informação além de tolerância a eventuais ruídos nos dados. Uma abordagem frequentemente utilizada na área de Visão Computacional para redução de dimensionalidade é a extração de *features* robustas, armazenando um subconjunto de informações representativas e simplificadas do conjunto de dados. Esta tese apresenta uma metodologia de classificação de objetos em nuvens de pontos 3D através da extração de *features* 3D globais. Foi desenvolvido um novo descritor 3D invariante à escala, translação e rotação denominado 3D-CSD (*3D-Contour Sample Distances*) para representação da superfície dos objetos presentes no ambiente, e utilizado um método de aprendizado supervisionado para reconhecimento de padrões. Os experimentos realizados envolveram o uso de Redes Neurais Artificiais para o reconhecimento de diferentes classes de objetos, avaliando e validando a metodologia proposta. Os resultados obtidos demonstraram a viabilidade da aplicação desta abordagem para o reconhecimento de objetos em sistemas de percepção 3D.

Palavras-chave: Robótica Móvel, Visão Computacional, Extração de features 3D, Reconhecimento de Padrões.

ABSTRACT

SALES, D. O. **3D feature extraction for objects recognition in point clouds**. 2017. 86 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2017.

Objects detection and recognition is a critical task in applications for mobile robots and intelligent vehicles autonomous navigation. With the advent of many 3D sensors, environment elements can be detected and represented in three-dimensional mode, in structures known as point clouds. 3D sensors usually capture a large amount of points at high rates, requiring robust techniques to process this information and also deal with noise on input data. A common approach in the Computer Vision field for dimensionality reduction is the use of robust features extraction techniques. This way, only a subset with representative and simplified information from the dataset is processed. This thesis presents a methodology for objects recognition in 3D point clouds using global 3D features extraction. A novel 3D descriptor invariant to scale, translation and rotation named 3D-CSD (3D-Contour Sample Distances) was developed to represent the objects surface, and a supervised learning method used for pattern recognition. The experiments were performed using Artificial Neural Networks for the recognition of different classes of objects, evaluating and validating the proposed methodology. Obtained results demonstrated the feasibility of this approach application for object recognition in 3D perception systems.

Keywords: Mobile Robotics, Computer Vision, Feature Extraction, Pattern Recognition.

LISTA DE ILUSTRAÇÕES

Figura 1 – Veículos autônomos utilizados no projeto CaRINA.	24
Figura 2 – Exemplo de mapas métrico e topológico.	28
Figura 3 – Ambiente interno e resultado de classificação com três classes semânticas. . .	29
Figura 4 – Fluxograma de controle de navegação topológica.	30
Figura 5 – Sensores utilizados em Robôs Móveis e Veículos Autônomos.	31
Figura 6 – Exemplo de nuvem de pontos obtida com o sensor Velodyne HDL-32. . . .	32
Figura 7 – Representação dos dados em descritores locais baseados em assinatura. . . .	34
Figura 8 – Representação dos dados em descritores locais baseados em histograma. . .	34
Figura 9 – Estrutura para assinatura e histograma do descritor SHOT.	35
Figura 10 – Exemplo de histograma gerado com o descritor <i>Shape Distributions</i>	36
Figura 11 – Representação de objeto com o descritor <i>Spherical Harmonics</i>	36
Figura 12 – Múltiplas projeções 2D e uso do descritor SIFT nas imagens resultantes. . .	36
Figura 13 – Duas classes de animais representadas com o descritor <i>Skeleton-Based</i>	37
Figura 14 – Modelo matemático do neurônio artificial.	40
Figura 15 – Modelos de arquitetura de Redes Neurais Artificiais	41
Figura 16 – Fluxograma do sistema de classificação proposto.	43
Figura 17 – Resultado de segmentação em uma nuvem de pontos.	44
Figura 18 – Exemplo de subconjunto de pontos 3D segmentados	45
Figura 19 – Comparação entre faces geradas com triangulação e <i>Convex Hull 3D</i>	46
Figura 20 – Representação de esfera circunscrita a um objeto 3D.	47
Figura 21 – Distribuição de pontos de referência na esfera.	48
Figura 22 – Seção transversal no plano $y = 0$ da esfera.	49
Figura 23 – Arquitetura da Rede Neural utilizada.	50
Figura 24 – Detalhe de objeto detectado parcialmente na nuvem de pontos.	55
Figura 25 – Posicionamento do centro da esfera em relação aos pontos 3D segmentados. .	56
Figura 26 – Esquema de particionamento e execução da validação cruzada com $k = 3$. . .	57
Figura 27 – Pontos 3D e faces dos objetos convexos utilizados nos testes.	58
Figura 28 – Erro médio quadrático (MSE) da RNA 326-56-6 nos 5 testes.	59
Figura 29 – Pontos 3D e faces dos objetos utilizados nos testes.	60
Figura 30 – Erro médio quadrático (MSE) da RNA 326-50-5 nos 5 testes.	61
Figura 31 – Modelos 3D utilizados para representação da classe “árvore”.	62
Figura 32 – Exemplos rotulados que não se adequam ao modelo proposto.	64

Figura 33 – Resultado do *Convex Hull 3D* no modelo de treinamento e em um exemplo da base de dados. 64

Figura 34 – Nuvens de pontos dos exemplos utilizados no experimento. 64

Figura 35 – Imagens processadas e matrizes binárias correspondentes. 82

LISTA DE ALGORITMOS

Algoritmo 1 – K-Means	39
Algoritmo 2 – Extração da <i>feature</i> 3D-CSD	45
Algoritmo 3 – Geração do conjunto de distâncias	48
Algoritmo 4 – <i>Pipeline</i> de classificação de objetos	51

LISTA DE TABELAS

Tabela 1 – Comparação entre descritores 3D	52
Tabela 2 – Características das 6 classes de objetos utilizados.	58
Tabela 3 – Matrizes de confusão resultantes para as 6 classes de objetos convexos.	59
Tabela 4 – Características das 5 classes de objetos utilizados.	60
Tabela 5 – Matrizes de confusão resultantes para as 5 classes de objetos utilizadas.	61
Tabela 6 – Matrizes de confusão resultantes do novo treinamento da RNA.	62
Tabela 7 – Comparação entre acurácia obtida nos trabalhos relacionados.	63
Tabela 8 – Comparação entre a acurácia obtida nas duas versões do sistema.	82
Tabela 9 – Matrizes de confusão dos testes do novo modelo.	83

LISTA DE ABREVIATURAS E SIGLAS

3D-CSD	3D Contour Sample Distances
3DSC	3D Shape Context
CaRINA	Carro Robótico Inteligente para Navegação Autônoma
CGAL	Computational Geometry Algorithms Library
CMU	Carnegie Mellon University
CVFH	Clustered Viewpoint Feature Histogram
DARPA	Defense Advanced Research Projects Agency
FPFH	Fast Point Feature Histograms
GPS	Global Positioning System
GPU	Graphics Processing Unit
HIGH	Height Gradient Histogram
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
InSac	Instituto Nacional de Ciência e Tecnologia para Sistemas Autônomos Cooperativos
JavaNNS	Java Neural Network Simulator
KNN	K-Nearest Neighbors
KPQ	KeyPoint Quality
LiDAR	Light Detection and Ranging
LRM	Laboratório de Robótica Móvel do ICMC-USP
MeshHOG	Mesh Histograms of Gradients
MLP	Multi-Layer Perceptron
R-Prop	Resilient Propagation
RNA	Rede Neural Artificial
SHOT	Signature of Histograms of Orientations
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SNNS	Stuttgart Neural Network Simulator
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
SWD	Spherical Wavelet Descriptor
VFH	Viewpoint Feature Histogram

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Objetivo	23
1.2	Histórico e Motivação	23
1.3	Estrutura da Tese	25
2	REFERENCIAL TEÓRICO	27
2.1	Robótica Móvel	27
2.1.1	<i>Controle de navegação em modelos topológicos</i>	29
2.1.2	<i>Sistemas de percepção e representação do ambiente</i>	30
2.2	Extração de Features 3D	32
2.2.1	Descritores 3D locais	33
2.2.1.1	<i>Descritores locais baseados em assinatura</i>	33
2.2.1.2	<i>Descritores locais baseados em histograma</i>	34
2.2.1.3	<i>Descritores locais híbridos</i>	35
2.2.2	Descritores 3D globais	35
2.2.2.1	<i>Descritores globais baseados em histogramas</i>	35
2.2.2.2	<i>Descritores globais baseados em transformações</i>	35
2.2.2.3	<i>Descritores globais baseados em visão 2D</i>	36
2.2.2.4	<i>Descritores globais baseados em grafos</i>	37
2.2.3	Aplicações baseadas no uso de descritores 3D	37
2.3	Aprendizado de Máquina	38
2.3.1	Aprendizado não-supervisionado	38
2.3.1.1	<i>K-Means</i>	39
2.3.2	Aprendizado supervisionado	39
2.3.2.1	<i>Redes Neurais Artificiais</i>	40
2.4	Considerações Finais	42
3	METODOLOGIA	43
3.1	Segmentação	44
3.2	Extração de features 3D	45
3.2.1	Criação de superfícies	46
3.2.2	Seleção de pontos de referência	47
3.2.3	Cálculo de distâncias da superfície	48

3.2.4	<i>Interpolação</i>	49
3.3	Reconhecimento de Padrões	50
3.3.1	<i>Arquitetura da RNA</i>	50
3.3.2	<i>Geração do conjunto de treinamento</i>	51
3.4	Considerações Finais	51
4	EXPERIMENTOS E RESULTADOS	55
4.1	Adaptações voltadas à aplicações em Robótica	55
4.2	Treinamento da RNA e validação dos resultados	57
4.3	Validação inicial do descritor 3D	58
4.4	Classificação de obstáculos	60
4.5	Classificação de dados obtidos do sensor Velodyne	64
4.6	Limitações do sistema	65
4.7	Considerações finais	66
5	CONCLUSÃO	67
5.1	Contribuições	68
5.2	Trabalhos Futuros	69
REFERÊNCIAS		71
GLOSSÁRIO		79
APÊNDICE A	EXPERIMENTOS COMPLEMENTARES	81
A.1	Reconhecimento de padrões do ambiente usando detecção de bordas	81
APÊNDICE B	LISTA DE PUBLICAÇÕES OBTIDAS	85

INTRODUÇÃO

Em Robótica Móvel, a detecção e reconhecimento de objetos é uma tarefa crucial. Vários processos relacionados a aplicações de navegação autônoma dependem de uma classificação precisa como: desvio de obstáculos, rastreamento de objetos, reconhecimento de pontos de referência (naturais ou artificiais) e segmentação semântica.

A capacidade perceptiva dos robôs aumentou significativamente nos últimos anos graças aos avanços tecnológicos que tem tornado o hardware cada vez mais barato e eficiente, e ao desenvolvimento de algoritmos robustos para processamento dos dados sensoriais, controle e tomada de decisão (ROMERO *et al.*, 2014). O uso de sensores que fornecem informações tridimensionais sobre o ambiente como câmeras estereoscópicas, LiDAR e infravermelho tem se tornado cada vez mais frequente, uma vez que estes fornecem uma maior quantidade de dados e com alta precisão.

Em consequência do grande volume de informação fornecido pelos sensores 3D surgiu a demanda por métodos mais eficientes de processamento dos dados, já que robôs móveis em geral são sistemas de tempo-real que exigem rápido processamento para a tomada de decisão. Uma abordagem amplamente difundida na área de reconhecimento de padrões e visão computacional para redução da dimensionalidade é o uso de técnicas de extração de *features* robustas (KRIG, 2016). Este processo consiste na transformação dos dados de entrada em um conjunto reduzido de características (*features*) representativas e não-redundantes, diminuindo assim a demanda por recursos computacionais e ao mesmo tempo favorecendo o processo de aprendizado e generalização. Uma *feature* é portanto, uma representação compacta - porém rica - de um conjunto de dados específico.

As estruturas que armazenam as características extraídas de um conjunto de dados permitindo a comparação entre esses dados por alguma métrica são chamadas de descritores. Dois atributos importantes de um descritor 3D são a descritividade e a robustez. A descritividade está relacionada à capacidade de encapsular a informação predominante de um determinado

conjunto, fornecendo informação suficiente para distinguir diferentes classes. Já a robustez está relacionada à invariância a uma família de transformações ou tipos de ruído (TOMBARI; SALT; STEFANO, 2013).

Existem dois tipos principais de descritores: locais e globais. Descritores locais representam áreas específicas de um objeto como cantos, junções e bordas e as aplicações associadas a este modelo são geralmente robustas à oclusão e ruído. Já os descritores globais fornecem a descrição de um objeto como um todo, sendo mais diretamente aplicáveis a tarefas de classificação (TUYTELAARS; MIKOLAJCZYK, 2008). Embora diversas abordagens de extração de *features* 3D já tenham sido propostas na literatura, ainda é difícil definir qual descritor é mais adequado para cada tipo de aplicação de classificação, já que esses modelos foram criados para cenários muito específicos e avaliados com conjuntos de teste limitados. Guo *et al.* (2016) analisaram o desempenho de alguns descritores 3D locais, porém apesar desse estudo apresentar uma boa comparação entre os métodos que compõem atualmente o estado-da-arte, os autores identificaram um desempenho inferior ao esperado com o uso de dados obtidos de sensores 3D de baixo custo. Sendo assim, o desenvolvimento de descritores mais robustos e abrangentes, compatíveis com dados provenientes de sensores com diversos níveis de precisão ainda é um desafio.

Esta tese explora este contexto, apresentando uma nova abordagem de extração de *features* 3D globais. Essa abordagem tem como fundamento o princípio de que todos os elementos de uma mesma classe de objetos apresentam um aspecto geral único (também chamado de assinatura 3D). A extração desta *feature* é obtida através de uma transformação nos dados de entrada (coordenadas X, Y, Z obtidos dos sensores), resultando em uma representação simplificada do contorno do objeto detectado. Desta forma, a representação desta característica através de um descritor 3D permite a aplicação de técnicas de Aprendizado de Máquina para reconhecimento de padrões e classificação.

O uso de técnicas de Aprendizado de Máquina para o reconhecimento de padrões proporciona robustez ao sistema para tratar dados que apresentem imprecisão e ruídos, característica comum em sensores 3D de baixo custo. Além disso, graças à capacidade de generalização desses métodos não é necessário estabelecer matematicamente os limites que definem cada classe, permitindo uma implementação simples e intuitiva do sistema de classificação, independentemente das particularidades de cada aplicação.

O processo de extração de *features* 3D proposto neste trabalho utiliza apenas um conjunto de pontos segmentados como entrada, e não necessita de nenhum tipo de informação adicional proveniente de técnicas de fusão de sensores como cores, texturas ou mapas de profundidade, demandando uma menor quantidade de recursos computacionais e permitindo o desenvolvimento de sistemas de baixo custo.

1.1 Objetivo

O objetivo principal desta tese é apresentar uma metodologia de tratamento dos dados obtidos de sensores 3D através da extração de *features* 3D, de forma que seja possível identificar objetos e reconhecer padrões em tarefas de navegação autônoma em ambientes controlados.

Este trabalho busca apresentar contribuições inéditas e significativas para o estado-da-arte em sistemas de percepção para Robótica Móvel, tendo como objetivos secundários:

- a concepção de um processo de classificação que seja apropriado tanto para aplicações de navegação por pontos de referência como para detecção e reconhecimento de obstáculos;
- compatibilidade com dados provenientes de diferentes sensores 3D;
- a proposta e implementação de um novo descritor 3D invariante a escala, rotação e translação para representação simplificada e homogênea dos objetos tridimensionais;
- a avaliação da descritividade e robustez do método proposto com a aplicação de Redes Neurais Artificiais para o aprendizado e reconhecimento de padrões nos dados de entrada.

1.2 Histórico e Motivação

As pesquisas relacionadas ao desenvolvimento de robôs móveis e veículos autônomos têm evoluído significativamente nos últimos anos, e têm despertado grande interesse tanto da comunidade científica como das empresas. Diversos projetos estão sendo desenvolvidos nessa área, visando diversas vantagens como a diminuição de acidentes de trânsito causados por fatores humanos, redução das deficiências permanentes ou temporárias que impedem um condutor humano de dirigir, aumento da produtividade (deixando o condutor livre para exercer outras atividades), até o uso otimizado dos recursos veiculares através da utilização adequada dos componentes (WOLF *et al.*, 2009; LUETTEL; HIMMELSBACH; WUENSCH, 2012).

De acordo com Dudek e Jenkin (2000), um dos primeiros e mais famosos trabalhos nesta área, é o projeto ALVINN (*Autonomous Land Vehicle In a Neural Network*) desenvolvido por Dean Pomerleau junto a CMU nos anos 80. Este trabalho serviu de inspiração para inúmeros outros projetos de pesquisa. Os projetos ganharam força após a iniciativa européia em investir nessa tecnologia, criando o Projeto PROMETHEUS (*EUREKA Prometheus Project: "PROGRAMme for a European Traffic of Highest Efficiency and Unprecedented Safety"*) e com as competições americanas realizadas pelo *Defense Advanced Research Projects Agency* (DARPA)¹. Trabalhos como o de Thrun *et al.* (2006) e de outros pesquisadores no *Darpa Desert Grand Challenge* (BUEHLER; IAGNEMMA; SINGH, 2007) e *Darpa Urban Challenge* (BECK *et al.*, 2007; URMSON, 2008), trouxeram grande visibilidade à área de desenvolvimento de veículos

¹ <<http://www.darpa.mil>>

autônomos. Mais recentemente, empresas também iniciaram o desenvolvimento de projetos na área de veículos autônomos como a Google, que contratou pesquisadores envolvidos nos projetos vencedores das competições do DARPA para o desenvolvimento de seu *Google Self-Driving Car*, dando origem à empresa Waymo² em 2016. Empresas do ramo automotivo como Ford, Volkswagen, Volvo e Nissan também têm investido em projetos voltados a sistemas autônomos.

Estes projetos serviram de inspiração para o projeto de Mestrado “NeuroFSM” (SALES, 2012) que antecedeu este trabalho, onde foi demonstrada a viabilidade do uso de sistemas baseados em representações simplificadas do ambiente para o controle de navegação de robôs móveis autônomos em ambientes estruturados. O sistema implementado utilizava Redes Neurais Artificiais para o reconhecimento de características estruturais 2D do ambiente, que serviam de referência para a localização dos robôs móveis. Os resultados obtidos motivaram o desenvolvimento de estudos mais avançados na extração de informação semântica do ambiente, tanto para classificação dos elementos dinâmicos presentes no ambiente como para tarefas de detecção de pontos de referência e localização.

O trabalho desenvolvido nesta tese busca criar métodos computacionais que permitam a implementação de sistemas de percepção com o uso de sensores 3D, podendo integrar tanto módulos de sistemas de auxílio ao motorista e direção autônoma como robôs de pequeno porte em tarefas de patrulhamento e segurança, viabilizando a implementação de diversas aplicações comerciais.

Este trabalho contribui ainda com projetos maiores realizados no Laboratório de Robótica Móvel (LRM³) do Instituto de Ciências Matemáticas e de Computação (ICMC), ligado ao CRob (Centro de Robótica da USP de São Carlos), e ao Instituto Nacional de Ciência e Tecnologia para Sistemas Autônomos Cooperativos (InSAC⁴). Destacam-se neste contexto os projetos voltados ao desenvolvimento de robôs táticos e de segurança, e o projeto CaRINA⁵, em desenvolvimento desde 2010, e que já conta atualmente com três veículos autônomos (Figura 1).

Figura 1 – Veículos autônomos utilizados no projeto CaRINA.

Fonte: Elaborada pelo autor.

² <<https://waymo.com/>>

³ <<http://www.lrm.icmc.usp.br/>>

⁴ <<http://www2.eesc.usp.br/insac/>>

⁵ <<http://lrm.icmc.usp.br/carina/>>

1.3 Estrutura da Tese

Para melhor compreensão do sistema desenvolvido, o conteúdo desta tese está organizado da seguinte forma:

- Capítulo 2: Apresenta brevemente os principais conceitos necessários para compreensão do tema abordado e os trabalhos que serviram como referência para esta tese.
- Capítulo 3: Apresenta a metodologia proposta para classificação de objetos e todas as etapas envolvidas, desde a segmentação dos elementos detectados pelo sensor até o reconhecimento de padrões, especificando o novo descritor 3D proposto.
- Capítulo 4: Descreve os experimentos realizados para validação da metodologia proposta e discute os resultados obtidos.
- Capítulo 5: Apresenta a conclusão do trabalho, principais contribuições e possibilidades para trabalhos futuros.
- Apêndice A: Descreve experimentos complementares realizados durante a execução desta pesquisa que não envolveram o uso de extração de *features* 3D.
- Apêndice B: Lista as publicações obtidas durante o doutorado relacionadas ao trabalho desenvolvido.

REFERENCIAL TEÓRICO

2.1 Robótica Móvel

Robôs Móveis Autônomos são robôs capazes de se locomover de um ponto a outro sem auxílio de controle humano local ou remoto, apresentam diversos níveis de inteligência e podem operar de forma isolada, em equipes ou mesmo interagindo com humanos. São compostos basicamente por três módulos principais: Percepção, Controle e Atuação. O módulo de percepção é composto de um ou mais sensores para captura de informações sobre o ambiente. O módulo de controle efetua o processamento dos dados de entrada obtidos e gera comandos de atuação de acordo com esses dados. O módulo de atuação corresponde aos motores, para deslocamento do robô pelo ambiente (ROMERO *et al.*, 2014).

Um sistema de navegação autônoma geralmente executa as seguintes operações básicas: mapeamento, localização, planejamento de trajetórias, controle e navegação. (WOLF *et al.*, 2009). O mapeamento é a criação de um modelo para representação do ambiente. A tarefa de localização está associada com a estimativa da posição do robô em um mapa conhecido, com base nos dados obtidos dos sensores. O planejamento de trajetórias consiste na definição de um caminho a ser seguido no ambiente a partir da posição atual (inicial) até uma posição desejada. Por fim, o controle de navegação consiste no processo de obter informação do ambiente através dos sensores, processá-la e atuar sobre os motores fazendo com que o robô se mova de forma segura, buscando alcançar a posição alvo ao mesmo tempo que desvia e evita colidir com os obstáculos presentes em seu caminho.

Dentre estas operações, pode-se destacar o mapeamento como uma operação fundamental, uma vez que as demais operações estão diretamente relacionadas ao modelo de mapa adotado. Existem dois modelos principais de mapas: métricos e topológicos. Mapas métricos são modelos altamente detalhados no que diz respeito a distâncias e geometria do ambiente, em duas ou três dimensões. Já os mapas topológicos são caracterizados por uma modelagem mais compacta,

utilizando grafos com nós e arestas que descrevem apenas as principais estruturas do ambiente e suas conexões, em uma representação semântica. A Figura 2 ilustra dois modelos de mapas: à esquerda uma grade de ocupação (métrico) e à direita um grafo relacional (topológico).

Figura 2 – Exemplo de mapas métrico e topológico.

Fonte: Romero *et al.* (2014).

Muitos trabalhos seguiram a linha de pesquisa proposta por Sebastian Thrun, que se baseia no mapeamento preciso do ambiente e na estimativa de localização também bastante precisa. Neste contexto a técnica *Simultaneous Localization and Mapping* – SLAM (THRUN; BURGARD; FOX, 2005) é uma das mais conhecidas na área da robótica móvel e tem sido amplamente utilizada para construção de mapas geometricamente consistentes simultaneamente à tarefa de auto-localização. Entretanto, é uma abordagem com alto custo computacional, e possui limitações em ambientes dinâmicos já que os mapas métricos são bastante densos e geralmente recalculados à medida que elementos dinâmicos são incluídos no modelo.

Por outro lado, há trabalhos que buscam o uso de uma abordagem baseada em mapas topológicos, identificação de pontos de referência, e navegação topológica (não tão dependentes de mapas e de localização extremamente precisos), e que de certo modo vão em uma direção diferente da proposta nos trabalhos baseados em representações métricas, ao mesmo tempo em que apresentam resultados promissores (LUETTEL; HIMMELSBACH; WUENSCH, 2012; MUELLER *et al.*, 2011).

Neste cenário, há alguns trabalhos de Wolfram Burgard (MOZOS *et al.*, 2006; MOZOS; STACHNISS; BURGARD, 2005) voltados a essa segunda linha de pesquisa, priorizando detecção de informações semânticas e uso de mapas topológicos. Em (MOZOS; STACHNISS; BURGARD, 2005), foi apresentado um sistema que classifica as informações obtidas dos sensores em diferentes classes semânticas (Figura 3), sendo proposta em (MOZOS *et al.*, 2006) uma abordagem baseada nesse sistema para a construção de mapas topológicos.

Figura 3 – Ambiente interno e resultado de classificação com três classes semânticas.

Fonte: Adaptada de [Mozos et al. \(2006\)](#).

Este tipo de abordagem foi aplicada no projeto de Mestrado “NeuroFSM: aprendizado de Autômatos Finitos através do uso de Redes Neurais Artificiais aplicadas à robôs móveis e veículos autônomos” ([SALES, 2012](#)), onde foram desenvolvidos sistemas de navegação topológica utilizando sensores 2D para ambientes internos ([SALES et al., 2012a](#)) e externos ([SALES et al., 2012b](#)). Os resultados obtidos neste projeto demonstraram a viabilidade deste tipo de abordagem para navegação autônoma tanto em robôs de pequeno porte como em veículos autônomos. Outro trabalho similar nesta linha de pesquisa foi desenvolvido por [Cheng, Chen e Liu \(2015\)](#), com uso de um classificador Bayesiano para reconhecimento das características estruturais do ambiente e geração do mapa topológico pelo próprio robô.

2.1.1 Controle de navegação em modelos topológicos

De acordo com [Luettel, Himmelsbach e Wuensche \(2012\)](#) e [Romero et al. \(2014\)](#), há três tipos de arquiteturas de controle de navegação autônoma: navegação global (ou deliberativa), navegação local (ou reativa) e navegação guiada (ou híbrida). Na navegação puramente deliberativa, o robô utiliza um mapa global do ambiente como referência para o planejamento prévio da trajetória, sem considerar possíveis elementos dinâmicos. Já a navegação puramente reativa corresponde ao processo inverso: o robô não precisa conhecer o mapa do ambiente, e a trajetória é definida durante a própria navegação, de acordo com as observações obtidas dos sensores. A abordagem de navegação híbrida combina os modelos reativo e deliberativo, uma vez que utiliza um mapa do ambiente para planejamento de trajetória global e simultaneamente um módulo de navegação reativa para desvio de obstáculos.

Sistemas de navegação baseados em mapas topológicos utilizam arquiteturas de controle de navegação híbridas, combinando um controle deliberativo para o planejamento de trajetória global (deslocamento pelos nós do mapa topológico) com um controle reativo para o deslocamento em um nível local (desvio de obstáculos). A Figura 4 apresenta o fluxograma de controle de um sistema de navegação topológica, com o uso da mesma informação sensorial para tomada de decisão nos níveis local e global.

Figura 4 – Fluxograma de controle de navegação topológica.

Fonte: Adaptada de Sales *et al.* (2014).

O robô pode assumir comportamentos reativos distintos de acordo com o contexto relacionado ao nó atual em que se encontra, e a localização global é obtida a partir da detecção de características estruturais específicas do ambiente ou de pontos de referência. Portanto, é importante a escolha de um sistema sensorial adequado para a detecção dessas características, associado ao uso de técnicas que permitam a extração de informações semânticas sobre o ambiente e os objetos contidos nele para que o robô consiga se localizar corretamente e navegar de forma segura.

2.1.2 Sistemas de percepção e representação do ambiente

Para que um robô consiga navegar de forma satisfatória pelo ambiente é necessário que ele possua um sistema sensorial, formado por um ou mais equipamentos responsáveis por capturar informações sobre o ambiente e sobre o próprio robô, e convertê-las em um modelo que possa ser interpretado pelo sistema de controle. Os principais tipos de sensores utilizados em robótica móvel são (ROMERO *et al.*, 2014):

Odômetro ótico: Odômetros óticos ou *encoders* são sensores que controlam a posição e velocidade das rodas do robô, fornecendo uma estimativa da distância relativa navegada.

Sensores de posição/orientação: São sensores que estimam a posição do robô em relação a uma referência absoluta como GPS (*Global Positioning System*), Bússola, IMU (*Inertial Measurement Unit*), Acelerômetros e Giroscópios.

Sonar: Calcula a distância de um obstáculo ao robô através do tempo de propagação e reflexão de pulsos sonoros emitido pelo sensor.

Câmeras: Capturam comprimentos de onda específicos da luz incidente na lente. Podem ser monoculares (visão 2D), estereoscópicas (visão 3D), omnidirecionais (visão em 360 graus) ou infravermelho (câmeras térmicas).

Laser: Também conhecidos como LiDAR (*Light Detection and Ranging*), têm funcionamento similar ao sonar, porém emitindo sinais de laser pulsados. Um espelho giratório direciona o feixe de laser, permitindo uma varredura planar (2D) ou em três dimensões no ambiente.

A Figura 5 ilustra alguns sensores utilizados em Robôs Móveis e Veículos Autônomos: Odômetro (a), Sonar (b), IMU (c), GPS (d), Laser (e), Câmera Monocular (f), Câmera Estereoscópica (g), Câmera Omnidirecional (h), Câmera Térmica (i), Kinect (j) e Laser 3D (k).

Figura 5 – Sensores utilizados em Robôs Móveis e Veículos Autônomos.

Fonte: Elaborada pelo autor.

Uma forma comum de representação dos dados mapeados com o uso de sensores 3D é através de estruturas chamadas nuvem de pontos. Uma nuvem de pontos é um conjunto de pontos expressos em coordenadas X, Y, Z que representam as superfícies dos objetos detectados na área de alcance do sensor. Nos sensores do tipo LiDAR, os pontos são a representação direta da distância medida por cada feixe do sensor. Já nas câmeras estereoscópicas, a profundidade é estimada pelo cálculo da disparidade entre as imagens obtidas por cada lente.

Cada sensor possui uma resolução e alcance específicos, além de características particulares que os tornam adequados para um tipo de ambiente específico. Em ambientes internos, diversos trabalhos têm adotado o sensor Kinect (HAN *et al.*, 2013). O Kinect é um sensor 3D de baixo custo desenvolvido pela Microsoft para o vídeo-game XBOX 360 que permite que o jogador use seu próprio corpo como controle nos jogos. A estimativa de profundidade é feita através de um emissor e receptor de infravermelho. Além disso, o Kinect possui uma câmera monocular integrada, o que permite que seja associada informação de cor aos pontos detectados. O Kinect detecta 76800 pontos a uma taxa de 30Hz, e tem alcance de 1,2m a 3,5m. O curto alcance e a sensibilidade do infravermelho à luz solar limitam o uso do Kinect a aplicações em ambientes internos.

Já em veículos autônomos, é frequente o uso de sensores do tipo LiDAR 3D (URMSON, 2008). Um exemplo de LiDAR 3D é o Velodyne HDL-64, que tem capacidade de captar mais de 1,3 milhão de pontos por segundo, com alcance de até 120 metros e erro inferior a 2cm. É um sensor com custo ainda bastante elevado, porém existem modelos com menor alcance e resolução como o HDL-32 e VLP-16, disponíveis por um menor custo. A Figura 6 apresenta a renderização 3D de uma nuvem de pontos obtida com o sensor LiDAR 3D Velodyne HDL-32.

Figura 6 – Exemplo de nuvem de pontos obtida com o sensor Velodyne HDL-32.

Fonte: <<http://irm.icmc.usp.br/carina/>>

Sistemas de navegação autônoma normalmente utilizam a combinação de vários sensores para uma representação precisa do ambiente. O veículo autônomo “Boss”, vencedor do *DARPA Urban Challenge 2007* (URMSON, 2008) por exemplo, utiliza uma combinação de dezenove sensores para este fim, sendo que onze deles são do tipo LiDAR (dez 2D e um 3D), de grande precisão. Abordagens topológicas podem utilizar sistemas sensoriais bem mais simples, com até mesmo um único sensor tridimensional, desde que sejam utilizados métodos robustos para tratar os dados de entrada, extraíndo a informação semântica necessária para a tomada de decisão.

2.2 Extração de *Features* 3D

Métodos de extração de *features* em geral tem se tornado cada vez mais populares. Um grande número de problemas podem ser resolvidos através dessa técnica, com diversas aplicações em vários domínios. Entretanto, apesar do recente sucesso e popularidade, o uso de *features* robustas não é uma abordagem nova. As primeiras técnicas de extração de *features* surgiram no final dos anos 70, com uma diferente terminologia - “pontos de interesse” - e um nível de invariância bem menor do que o obtido atualmente, mas ainda assim podendo ser associadas à categoria de *features* locais (TUYTELAARS; MIKOLAJCZYK, 2008).

A criação de poderosos descritores locais por Lowe (1999) teve um impacto decisivo na popularização desta técnica. Os pesquisadores da área de visão computacional logo perceberam as vantagens decorrentes dessa abordagem e então surgiu uma grande quantidade de trabalhos utilizando a extração de *features* como uma ferramenta para processar imagens, com alto nível de confiabilidade e repetibilidade. Atualmente, o uso de extração de *features* em aplicações baseadas em visão 2D já é uma ideia amplamente difundida e consolidada. SIFT (LOWE, 2004) e SURF (BAY *et al.*, 2008) são exemplos de abordagens bem-sucedidas e bastante conhecidas de extração de *features* 2D.

Com o surgimento e evolução de sensores 3D como câmeras estereoscópicas, LiDAR 3D e Kinect, além da integração com dados provenientes de outros sensores como acelerômetros e unidades inerciais nas aplicações, o volume de informação processada é cada vez maior, aumentando também o consumo de memória. Isso criou uma demanda por métodos robustos para tratamento dos dados tridimensionais, estimulando a criação de descritores de *features* 3D (KRIG, 2016). O sucesso das abordagens baseadas em extração de *features* em visão 2D inspirou o desenvolvimento de abordagens similares na análise de formas tridimensionais, tanto através de analogias com os métodos já conhecidos de extração de *features* 2D como explorando as particularidades da construção dos elementos 3D (PEARS; LIU; BUNTING, 2012). Diversas abordagens para extração de *features* 3D já foram propostas até o momento, e assim como os descritores 2D, os descritores de *features* 3D são divididos de acordo com a proporção da informação que representam em relação ao objeto, podendo ser globais ou locais.

2.2.1 Descritores 3D locais

Descritores locais incorporam características da vizinhança de um ponto de referência (ou ponto-chave) específico. Possuem alta descritividade e são robustos a ruído e oclusão, mas estão associados a uma maior dimensionalidade. De acordo com Tombari, Salti e Stefano (2010), os descritores 3D locais são divididos em três sub-categorias: descritores baseados em assinatura, descritores baseados em histogramas e descritores locais híbridos.

2.2.1.1 Descritores locais baseados em assinatura

Nos descritores baseados em assinatura, uma ou mais relações geométricas são computadas individualmente em cada ponto da vizinhança do ponto de referência. Um sistema de referência (conjunto de vetores ortogonais) é utilizado como auxiliar para que os pontos sejam processados corretamente e para adicionar invariância à rotação e translação. Esta classe de descritores é geralmente altamente descritiva, porém mais sensível a ruído. Alguns exemplos de trabalhos nesta categoria são: *Point Signatures* (CHUA; JARVIS, 1997), KPQ (MIAN; BENNA-MOUN; OWENS, 2010) e 3D SURF (KNOPP *et al.*, 2010), por exemplo. A figura 7 ilustra a composição de um descritor local baseado em assinatura.

Figura 7 – Representação dos dados em descritores locais baseados em assinatura.

Fonte: Adaptada de [Salti, Tombari e Stefano \(2014\)](#).

2.2.1.2 Descritores locais baseados em histograma

Em descritores baseados em histograma, entidades topológicas locais como vértices e áreas são contadas de acordo com um domínio quantificável como por exemplo coordenadas dos pontos, curvaturas e ângulos normais às superfícies. Estes descritores são mais robustos a ruído, porém menos descritivos do que os modelos baseados em assinatura. Alguns exemplos são: *Spin Images* ([JOHNSON; HEBERT, 1999](#)), 3DSC ([FROME et al., 2004](#)), FPFH ([RUSU; BLODOW; BEETZ, 2009](#)), HIGH ([ZHAO; YUAN; DANG, 2014](#)). A figura 8 ilustra a representação dos dados em descritores locais baseados em histograma.

Figura 8 – Representação dos dados em descritores locais baseados em histograma.

Fonte: Adaptada de [Salti, Tombari e Stefano \(2014\)](#).

2.2.1.3 Descritores locais híbridos

Descritores locais híbridos combinam as duas abordagens anteriores, representando uma assinatura de histogramas. Alguns descritores conhecidos desta categoria são MeshHoG (ZAHARESCU *et al.*, 2009) e SHOT (TOMBARI; SALT; STEFANO, 2010) (Figura 9).

Figura 9 – Estrutura para assinatura e histograma do descritor SHOT.

Fonte: Adaptada de Salti, Tombari e Stefano (2014).

2.2.2 Descritores 3D globais

Descritores globais encapsulam informações referentes ao objeto como um todo em uma única estrutura. Este modelo é mais sensível a oclusão, porém provê maior invariância, sendo bastante adequado para aplicações de classificação. De acordo com a taxonomia apresentada por Akgul *et al.* (2009), os descritores 3D globais são divididos em quatro categorias: descritores baseados em histogramas, descritores baseados em transformações, descritores baseados em visão 2D e descritores baseados em grafos.

2.2.2.1 Descritores globais baseados em histogramas

De forma análoga aos descritores locais baseados em histogramas, estes descritores armazenam uma contagem de características globais da superfície do objeto 3D. É uma abordagem robusta, porém com menor descritividade. Alguns descritores desta classe são: Viewpoint Feature Histogram (VFH) (RUSU *et al.*, 2010), Clustered-VFH (ALDOMA *et al.*, 2011), 3D Shape Histograms (ANKERST *et al.*, 1999), Orientation Histograms (HORN, 1984) e Shape Distributions (OSADA *et al.*, 2002), ilustrado na Figura 10.

2.2.2.2 Descritores globais baseados em transformações

O descritor é obtido através de uma transformação geométrica que converte os dados originais para um domínio onde a representação seja mais compacta e invariante. Apenas um subconjunto de coeficientes é armazenado. Pertencem a essa classe: 3D Fourier Transform (DUTAGACI; SANKUR; YEMEZ, 2005), Angular Radial Transform (RICARD; COEURJOLLY; BASKURT, 2005), 3D Radon Transform (DARAS *et al.*, 2004), Spherical Wavelet Descriptor (SWD) (LAGA; TAKAHASHI; NAKAJIMA, 2006) e Spherical Harmonics (KAZHDAN; FUNKHOUSER; RUSINKIEWICZ, 2003), ilustrado na Figura 11.

Figura 10 – Exemplo de histograma gerado com o descritor *Shape Distributions*.

Fonte: Adaptada de Osada *et al.* (2002).

Figura 11 – Representação de objeto com o descritor *Spherical Harmonics*.

Fonte: Adaptada de Kazhdan, Funkhouser e Rusinkiewicz (2003).

2.2.2.3 Descritores globais baseados em visão 2D

Nesta classe de descritores, a superfície 3D é transformada em um conjunto de projeções 2D de diferentes ângulos. Em seguida, são computados descritores 2D convencionais em cada imagem resultante como: Fourier descriptors (VRANIC, 2004), Zernike moments (CHEN *et al.*, 2003) e SIFT (OHBUCHI *et al.*, 2008), ilustrado na Figura 12.

Figura 12 – Múltiplas projeções 2D e uso do descritor SIFT nas imagens resultantes.

Fonte: Adaptada de Ohbuchi *et al.* (2008).

2.2.2.4 Descritores globais baseados em grafos

Nos descritores baseados em representação por grafos, a topologia do objeto é representada em um grafo, que depois é convertido em um vetor com relações numéricas entre os nós. Algumas abordagens contidas nesta classe são: Reeb graph (TUNG; SCHMITT, 2005), Topology-based (HILAGA *et al.*, 2001) e skeleton-based (SUNDAR *et al.*, 2003), ilustrado na Figura 13.

Figura 13 – Duas classes de animais representadas com o descritor *Skeleton-Based*.

Fonte: Adaptada de Sundar *et al.* (2003).

2.2.3 Aplicações baseadas no uso de descritores 3D

A extração de features 3D é uma solução interessante para diversas aplicações em áreas como Realidade Aumentada, Reconstrução 3D, Interação Homem-Máquina e Robótica Móvel. No âmbito da navegação autônoma em ambientes internos, Kim e Sukhatme (2014) apresentam um sistema de segmentação semântica usando o sensor Kinect para geração da nuvem de pontos. Em (QIAN; YE, 2014), é apresentado um método para reconhecimento de objetos e características do ambiente. Já em (MADRY *et al.*, 2013) é utilizado um processo para seleção de subconjuntos essenciais de *features* locais para classificação e reconhecimento de objetos.

Já no domínio de veículos autônomos em vias urbanas, Gerónimo *et al.* (2010) apresentam uma série de trabalhos utilizando extração de *features* para reconhecimento de pedestres, Tombari *et al.* (2014) utilizam features locais e globais para detecção de postes, e em (YU *et al.*, 2015) são detectadas marcações na pista. Em (DOUILLARD *et al.*, 2014) e (ZHU *et al.*, 2010) são apresentados sistemas de segmentação semântica de objetos. (HABERMANN *et al.*, 2016) desenvolveram um sistema de detecção de guias e cruzamentos útil tanto para o reconhecimento de área navegáveis como para a detecção de características estruturais do ambiente, aplicando a abordagem de navegação topológica discutida no início deste Capítulo.

A aplicação de extração de *features* também é bem-sucedida em outras áreas como: reconhecimento de faces (ABIAN TUN; PRABHU; SAVVIDES, 2014; ALI, 2014), reconhecimento de gestos e pose (CHAN; KOH; LEE, 2013; WAN; GUO; LI, 2016) e análise de informações topográficas (ABELLAN; DERRON; JABOYEDOFF, 2016).

Apesar da grande variedade de técnicas e aplicações já desenvolvidas, a área de extração de *features* 3D ainda está relativamente no início de seu ciclo de desenvolvimento, e não há uma previsão exata de quais métodos serão mais frequentemente adotados (KRIG, 2016). Alguns trabalhos como (GUO *et al.*, 2016; KAZMI; YOU; ZHANG, 2013; ALI; MARTÓN, 2014; CANELHAS; STOYANOV; LILIENTHAL, 2016) apresentam comparações e análises entre diferentes descritores 3D visando identificar quais os descritores mais indicados para diferentes tipos de aplicação. No entanto, apesar de evidenciar as vantagens de cada método, estes trabalhos também concluíram que o desempenho dos descritores atuais com dados oriundos de sensores de baixo custo menos precisos é baixo.

Um dos desafios atuais nas pesquisas envolvendo descritores de *features* é encontrar uma boa proporção entre invariância à transformações, descritividade, robustez a ruídos e complexidade computacional. O equilíbrio ideal entre esses fatores ainda está longe de ser encontrado, já que estes parâmetros dependem fortemente da aplicação (PEARS; LIU; BUNTING, 2012).

Sendo assim, a criação de descritores adequados a múltiplas aplicações e compatíveis com diferentes sensores ainda é uma meta a ser alcançada (GAO; DAI, 2014). Estudos envolvendo o desenvolvimento de novas técnicas de extração de *features* 3D podem resultar em métodos robustos para superar as limitações dos descritores atuais, produzindo abordagens abrangentes e ao mesmo tempo eficazes.

2.3 Aprendizado de Máquina

O Aprendizado de Máquina é uma área relacionada a Inteligência Artificial que explora a construção de algoritmos capazes de fazer previsões ou tomar decisões baseados apenas em um modelo construído a partir de um conjunto de dados de entrada. Técnicas de Aprendizado de Máquina tem sido amplamente utilizadas para a solução de problemas em inúmeras aplicações. Na Robótica, são frequentemente utilizadas para segmentação e/ou reconhecimento dos objetos no conjunto de dados obtidos dos sensores. O aprendizado de máquina se divide em três categorias: aprendizado supervisionado, aprendizado não-supervisionado e aprendizado por reforço (ou semi-supervisionado) (MITCHELL, 1997).

2.3.1 Aprendizado não-supervisionado

O aprendizado não-supervisionado caracteriza-se pela ausência de rótulos nas diferentes classes de elementos. O foco é determinar como os dados estão organizados de acordo com a similaridade entre eles em alguma métrica, separando esses dados em classes ou agrupamentos. Quando dois elementos apresentam padrões semelhantes, ambos são associados a uma mesma classe. Quando um padrão não apresenta um nível mínimo de similaridade com nenhuma classe, o algoritmo cria uma nova classe para inserir este elemento.

Técnicas de agrupamento de dados ocupam um papel importante em várias áreas como mineração de dados, segmentação de imagem e reconhecimento de objetos, por exemplo. O aprendizado não-supervisionado dispensa a necessidade de um especialista para rotular manualmente os dados, o que torna essa abordagem interessante para tarefas de segmentação. De um modo geral, as técnicas de agrupamento de dados se dividem em hierárquicas, baseadas em partições, baseadas em densidade e baseadas em distribuição. No caso específico da segmentação em dados em nuvem de pontos, o modelo baseado em partições é o mais apropriado (KLASING; WOLLHERR; BUSS, 2008).

2.3.1.1 K-Means

O *K-Means* é um algoritmo de agrupamento baseado em partições bastante conhecido e utilizado. Inicialmente, são posicionados de forma aleatória k centroides que representam os centros de cada grupo. Cada ponto x_i é associado ao centróide C_k mais próximo e o conjunto de pontos associados a cada centróide passa a representar um grupo. Então, cada centróide é movido em direção à média de cada conjunto e as associações de pontos aos centroides são refeitas. Esse processo de mover os centroides é interrompido quando todos os pontos não mudam suas associações ou apenas quando parte deles muda. O algoritmo 1 apresenta o *K-Means* (MURPHY, 2012).

Algoritmo 1 – K-Means

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

2.3.2 Aprendizado supervisionado

O objetivo do aprendizado supervisionado é induzir conceitos a partir de um conjunto de exemplos pré-classificados, rotulados com uma classe conhecida, que formam o que é chamado de conjunto de treinamento. Se as classes possuírem valores discretos, o problema é categorizado como classificação. Caso elas possuam valores contínuos, o problema é denominado regressão.

Nos modelos de aprendizado supervisionado, é dado para o sistema um conjunto de pares de valores (entradas e saídas) compostos por elementos específicos de um conjunto de dados e suas respectivas classificações. O modelo precisa então passar por uma fase de treinamento, onde recebe os valores de entrada e verifica se para cada um deles foi gerada a saída esperada. Se isto não ocorrer, o modelo é reestruturado até que seja alcançado o menor erro neste processo, ou até atingir uma quantidade específica de iterações (MITCHELL, 1997).

2.3.2.1 Redes Neurais Artificiais

As RNAs tem seu funcionamento inspirado nos mecanismos e estruturas do cérebro humano, sendo formadas por unidades de processamento (neurônios) e conexões entre esses neurônios (sinapses) responsáveis pelo processamento de informações (SILVA; SPATTI; FLAUZINO, 2010). O neurônio artificial, ilustrado na Figura 14 é composto de três elementos básicos: conexões com pesos (w_i) que permitem propagação dos dados de entrada (x_i) para outros neurônios, um somador (Σ) que efetua a somatória dos dados que chegam até esse neurônio, e uma função de ativação/transferência ($g(u)$), que determina o valor da saída (y) de acordo com o resultado do somador.

Figura 14 – Modelo matemático do neurônio artificial.

Fonte: Silva, Spatti e Flauzino (2010).

O treinamento de uma rede neural é realizado através de um algoritmo de treinamento, e um conjunto de exemplos, também chamado de base de treinamento. Durante o treinamento, os pesos das conexões são ajustados para a melhor correspondência entre as entradas e saídas dos exemplos utilizados como base de treinamento (RUSSELL; NORVIG, 2003). A escolha do algoritmo de aprendizado afeta tanto a precisão da resposta quanto o tempo de treinamento. Alguns dos algoritmos de aprendizado mais utilizados são o algoritmo backpropagation (RUMELHART; HINTON; WILLIAMS, 1986) e suas derivações.

A forma como os neurônios se conectam é chamada de arquitetura, e está relacionada ao tipo de problema que a rede será capaz de tratar. As diferentes composições de neurônios considerando uma determinada arquitetura são chamadas de Topologia. Tradicionalmente, as topologias das RNAs apresentam seus neurônios dispostos em camadas. Há três tipos de arquitetura principais: *Perceptron*, MLP e Recorrente (SILVA; SPATTI; FLAUZINO, 2010).

O *Perceptron* (Figura 15a), também conhecido como *single-layer feedforward* é a configuração mais básica de RNA. Possui apenas uma camada de entrada e outra de saída, sem ciclos entre as conexões dos neurônios. Se restringe à resolução de problemas lineares (SILVA; SPATTI; FLAUZINO, 2010). O *Multi-layer Perceptron* (MLP) ou *multi-layer feedforward* (Figura 15b) adiciona camadas ocultas ao modelo *perceptron* tradicional, estendendo sua capacidade para resolução de problemas não-lineares, além de possibilitar o aumento na precisão dos resultados (HAYKIN, 1998). Na topologia Recorrente (Figura 15c) há a presença de um laço que conecta a saída da rede à sua entrada (ou a uma camada anterior), de forma a realimentar a RNA. As redes recorrentes possuem a capacidade de simular uma memória de curto prazo.

Figura 15 – Modelos de arquitetura de Redes Neurais Artificiais

Fonte: [Silva, Spatti e Flauzino \(2010\)](#).

As Redes Neurais Artificiais permitem uma boa tolerância ao ruído e a imprecisão nos dados de entrada, assim como são muito eficientes em generalizar o conhecimento, adaptando suas saídas em relação a diferentes entradas, mesmo que estas não tenham sido explicitamente ensinadas a rede ([HAYKIN, 1998](#)). Além disso, as RNAs aparecem como uma boa opção em aplicações que dependem de um baixo tempo de execução, uma vez que o treinamento é feito separadamente, antes da execução do sistema. Desta forma, como as Redes Neurais Artificiais são representadas através de vetores ou matrizes com o conjunto de pesos sinápticos ajustados, o tempo de processamento dos sinais de entrada pela rede durante uma aplicação é apenas o tempo de processamento de uma operação de multiplicação de matrizes ([SILVA; SPATTI; FLAUZINO, 2010](#)).

2.4 Considerações Finais

Robôs móveis e veículos inteligentes possuem sistemas sensoriais responsáveis por capturar informações do ambiente para a tomada de decisão tanto em um plano local (desvio de obstáculos) como global (localização em mapas topológicos e planejamento de rotas). Na grande maioria das situações é necessária a classificação dos dados obtidos dos sensores. Essa tarefa de reconhecimento de padrões pode ser complexa devido ao alto volume de dados capturados pelos sensores e a imprecisão característica de cada tipo de sensor, o que demanda técnicas robustas para o tratamento dessa informação.

A representação dos elementos através de descritores de *features* é uma estratégia amplamente utilizada neste tipo de problema para redução de dimensionalidade. Neste contexto, o uso de um método de aprendizado supervisionado como Redes Neurais Artificiais é uma abordagem interessante para o reconhecimento dos padrões representados nos descritores devido a sua capacidade de aprendizado através de exemplos e generalização desse aprendizado.

Neste trabalho foram combinadas estas abordagens, criando um sistema capaz de aprender e reconhecer classes de objetos detectados em nuvem de pontos 3D. O Capítulo 3 apresenta a metodologia adotada para este propósito.

METODOLOGIA

Este capítulo apresenta a metodologia empregada para a classificação dos objetos em uma cena 3D. A abordagem proposta é dividida em três etapas principais: segmentação dos objetos, extração de *features* 3D e reconhecimento de padrões. Na etapa de segmentação, os pontos 3D são agrupados de acordo com a sua posição no ambiente e armazenados em estruturas individuais. Em seguida, são extraídas as características da superfície de cada elemento segmentado através do uso de um descritor 3D invariante a escala, rotação e translação, que computa basicamente a distância do centro do objeto a um conjunto de pontos de referência. Por fim, cada elemento é rotulado de acordo com o padrão identificado no descritor. A Figura 16 apresenta o fluxograma do sistema como um todo. Cada etapa é descrita nas seções seguintes.

Figura 16 – Fluxograma do sistema de classificação proposto.

Fonte: Elaborada pelo autor.

3.1 Segmentação

A segmentação de elementos em nuvem de pontos tem os mesmos princípios e objetivos da segmentação em imagens, simplificando a representação do ambiente e possibilitando a análise individual dos elementos de interesse contidos no conjunto de dados. A escolha do método de segmentação mais adequado depende da complexidade do ambiente em cada aplicação, e do tipo de sensor utilizado. Em algumas aplicações, uma subtração de plano de fundo (BOUWMANS *et al.*, 2015) mais simples pode ser aplicada como, por exemplo, remover todos os pontos além de uma distância fixa do sensor ou de um plano específico. Em cenas mais complexas, é necessário aplicar técnicas mais robustas com algoritmos de agrupamento de dados como o K-Means ou ainda técnicas de segmentação baseadas em movimento.

Para aplicações em robótica móvel, a segmentação da nuvem de pontos é geralmente dividida em duas fases: extração de superfície (chão) e segmentação dos objetos que estão acima da superfície. Douillard *et al.* (2014) apresenta um pipeline para segmentação utilizando esta abordagem. Inicialmente os pontos são classificados em “chão” e “não-chão”. Em seguida, os pontos classificados como “não-chão” são agrupados de acordo com a adjacência entre suas coordenadas 3D. A Figura 17 ilustra o resultado deste procedimento em uma nuvem de pontos, representando com cores diferentes cada estrutura segmentada.

Figura 17 – Resultado de segmentação em uma nuvem de pontos.

Fonte: Douillard *et al.* (2014).

O objetivo final desta etapa de segmentação é dividir o conjunto inicial de pontos obtidos do sensor em um conjunto de estruturas cujos pontos pertençam somente a um único elemento da cena 3D, como ilustrado na Figura 18. Uma vez segmentados os dados, é possível iniciar a etapa de extração de *features* 3D para cada subconjunto.

Figura 18 – Exemplo de subconjunto de pontos 3D segmentados

Fonte: Elaborada pelo autor.

3.2 Extração de *features* 3D

Nesta etapa, um descritor 3D é utilizado para representar as estruturas segmentadas de forma simplificada e invariante, permitindo que um algoritmo de aprendizado de máquina seja utilizado para classificação. Foi criado um novo descritor 3D global chamado de 3D-CSD (*3D-Contour Sample Distances*), capaz de armazenar informação sobre a forma (contorno 3D) do objeto. Este descritor é inspirado em características de alguns métodos de extração de *features* 3D que compõem atualmente o estado-da-arte, como SHOT e Shape Distributions, combinando uma abordagem de representação esférica ao modelo de features baseadas em assinatura.

Basicamente é feita uma transformação nos dados de entrada, substituindo a representação inicial do objeto (conjunto de pontos 3D) por um *array* unidimensional de distâncias, medidas do centro de massa do objeto até a sua superfície, na direção de vetores formados entre o centro de massa do objeto e um conjunto de pontos de referência (amostras). Esses pontos de referência estão distribuídos em uma esfera virtual que circunscreve o conjunto de pontos 3D. Por fim, as medidas calculadas são interpoladas para um intervalo apropriado. O resultado é uma representação simplificada do contorno 3D do objeto, descrevendo a sua forma através de uma assinatura 3D global. O Algoritmo 2 apresenta o procedimento para extração dessa *feature* em um conjunto de pontos segmentados. Cada etapa é discutida individualmente nas próximas seções.

Algoritmo 2 – Extração da *feature* 3D-CSD

- 1: Criar as faces da superfície do objeto através de triangulação entre os pontos 3D
 - 2: Criar uma esfera que circunscreve o conjunto de pontos
 - 3: Selecionar um conjunto de pontos de referência na esfera
 - 4: **para todo** ponto de referência da esfera **faça**
 - 5: Calcular distância entre o centro de massa do objeto e sua superfície na direção do ponto de referência
 - 6: Interpoliar distância calculada para um intervalo definido
 - 7: **fim para**
-

3.2.1 Criação de superfícies

Antes de calcular as distâncias que descrevem a forma do objeto tridimensional, é necessário transformar a nuvem de pontos segmentada em um objeto sólido. Para isto, devem ser criadas as faces do objeto, que são os planos de forma poligonal que conectam os pontos da superfície. Este procedimento pode ser feito de duas maneiras: usando um algoritmo de triangulação tradicional ou aplicando um algoritmo de *Convex Hull 3D* nos pontos segmentados.

Os algoritmos de triangulação são baseados na Triangulação de Delaunay (BERG *et al.*, 2008), e criam as faces conectando 3 ou mais pontos com um mesmo plano. Basicamente, o algoritmo cria planos triangulares usando combinações de 3 pontos, de forma que para cada subconjunto nenhum outro ponto do conjunto principal esteja contido na esfera que contém o plano gerado. A Triangulação de Delaunay maximiza o menor ângulo de todos os triângulos na triangulação, evitando assim planos com ângulos internos muito pequenos.

Já o *Convex Hull 3D* (BERG *et al.*, 2008) cria o sólido convexo de menor volume possível, selecionando um subconjunto de vértices. A precisão na representação da forma original do objeto diminui, mas ainda assim é possível observar assinaturas 3D distintas entre classes. A grande vantagem do *Convex Hull 3D* é que número de faces geradas é bem menor em relação à triangulação de Delaunay, representando um ganho considerável de desempenho nas etapas seguintes da geração do descritor. A Figura 19 exibe uma comparação entre superfícies criadas com triangulação (esquerda) e com *Convex Hull 3D* (direita) em um mesmo conjunto de pontos.

Figura 19 – Comparação entre faces geradas com triangulação e *Convex Hull 3D*

Fonte: Elaborada pelo autor.

Bibliotecas como a *Computational Geometry Algorithms Library* (CGAL) ¹ possuem implementações dos dois algoritmos em linguagem C++, além de outras operações geométricas úteis para cálculos envolvendo estruturas 3D. Os dois algoritmos podem ser executados em tempo $O(n \log n)$ (BERG *et al.*, 2008), portanto a escolha do algoritmo de criação de superfícies mais adequado depende da aplicação, já que o nível de precisão necessário na representação do objeto pode variar conforme as classes a serem reconhecidas, quantidade média de pontos nos dados segmentados e distância entre esses pontos.

¹ <<http://www.cgal.org/>>

3.2.2 Seleção de pontos de referência

Após a segmentação dos elementos da cena e construção de superfícies para geração das faces do objeto 3D, é necessário definir os pontos de referência para o cálculo das distâncias que descreverão a forma do objeto. Para iniciar este processo, é necessário criar uma esfera que circunscreve o conjunto de pontos 3D, como ilustrado na Figura 20.

Figura 20 – Representação de esfera circunscrita a um objeto 3D.

Fonte: Elaborada pelo autor.

O centro da esfera C_{xyz} é posicionado no centro de massa do conjunto de pontos 3D, e o raio da esfera r é igual à distância euclidiana entre o centro de massa e o ponto mais distante a esse centro no conjunto de dados. Em seguida, são distribuídos os pontos de interesse na superfície da esfera. Cada ponto incluído será usado para a geração de uma reta utilizada no cálculo de distâncias, representando portanto uma amostra da superfície do objeto em uma região, e correspondendo a uma posição única na estrutura final do descritor (vetor de distâncias).

Os pontos de interesse podem ser distribuídos em intervalos constantes ou concentrados em áreas específicas da esfera, priorizando áreas potencialmente mais representativas das classes de objetos. A estratégia de distribuição pode variar de acordo com a aplicação. A Figura 21 ilustra a estratégia genérica de distribuição de pontos, onde os mesmos são posicionados de acordo com um intervalo fixo nos ângulos vertical e azimutal.

É importante ressaltar que como a quantidade de pontos distribuídos corresponde ao tamanho da estrutura do descritor, é preciso limitar a quantidade de pontos distribuídos para evitar incorrer em problemas de dimensionalidade na etapa de classificação.

Figura 21 – Distribuição de pontos de referência na esfera.

Fonte: Elaborada pelo autor.

3.2.3 Cálculo de distâncias da superfície

Uma vez definidos os pontos de referência, é possível iniciar o cálculo das distâncias para geração da estrutura (*array*) que corresponde ao descritor 3D-CSD. O Algoritmo 3 descreve a sequência de etapas deste procedimento.

Algoritmo 3 – Geração do conjunto de distâncias

```

1: para todo ponto de referência  $P_i$  na superfície da esfera faça
2:   Considere uma reta do centro de massa  $C_{xyz}$  até  $P_i$ 
3:   se há interseção da reta com alguma face do objeto em um ponto  $d_i$  então
4:     retorna distância euclidiana entre  $C_{xyz}$  e  $d_i$ 
5:   senão
6:     retorna -1
7:   fim se
8: fim para

```

Se houver interseção de uma reta com mais de uma face do objeto (situação possível em objetos côncavos), é considerado o ponto de interseção mais distante ao centro C_{xyz} . Em casos onde não há interseção da reta com nenhuma face (situação possível em Toróides), é armazenado o valor -1 na posição correspondente do *array*.

A sequência de processamento dos pontos de referência P_i em relação à esfera (por exemplo, esquerda para direita e cima para baixo) é importante e deve ser constante para a consistência das assinaturas 3D geradas.

A Figura 22 ilustra o processo de cálculo dos pontos de interseção em uma seção transversal (plano $y = 0$) de uma esfera circunscrita a um cubo.

Figura 22 – Seção transversal no plano $y = 0$ da esfera.

Fonte: Elaborada pelo autor.

3.2.4 Interpolação

Uma operação de interpolação linear é necessária para converter as distâncias calculadas em valores dentro de um intervalo discreto. Esse procedimento é importante para produzir uma entrada adequada para a etapa de reconhecimento de padrões, e ao mesmo tempo é fundamental para a obtenção de invariância à escala. Desta forma, a assinatura 3D que representa a forma do objeto é relacionada às proporções entre as distâncias medidas ao invés de seus valores absolutos.

As distâncias medidas variam inicialmente entre 0 e r , onde r é o raio da esfera. Para geração de uma entrada adequada para a etapa seguinte (reconhecimento de padrões), há duas possibilidades:

- Valores contínuos (intervalo $\{0,1\}$ no domínio $\mathbb{R}$);
- Valores discretos (intervalo $\{0,2^n\}$ no domínio $\mathbb{N}$), sendo n relacionado à resolução adotada para representação das distâncias.

Os valores interpolados d_N são obtidos através da fórmula:

$$d_N = \left(d \times \frac{\text{newMax} - \text{newMin}}{r} \right) + \text{newMin} \quad (3.1)$$

Valores contínuos são obtidos diretamente do resultado d_N , enquanto valores discretos são obtidos pela parte inteira de d_N , usando funções piso, teto ou arredondamento. Após a interpolação, o descritor está pronto para a etapa de reconhecimento de padrões.

3.3 Reconhecimento de Padrões

A última etapa do sistema de classificação é o Reconhecimento de Padrões. Nesta etapa, é utilizado um método de aprendizado supervisionado para reconhecimento da assinatura 3D global fornecida pelo descritor 3D-CSD. Foi adotada a técnica de classificação usando Redes Neurais Artificiais, devido a sua capacidade de detectar características complexas e não-lineares com um alto nível de precisão. Uma das principais vantagens no uso desta técnica para classificação é a capacidade de generalização, provendo robustez ao sistema em relação à ruídos e imprecisão nos dados dos sensores. Além disso, métodos de aprendizado de máquina são estratégias não-paramétricas, o que significa que não é necessário conhecer e estabelecer matematicamente os limites que definem cada classe, já que o sistema aprende através de exemplos.

3.3.1 Arquitetura da RNA

O sistema proposto utiliza uma Rede Neural Artificial de arquitetura MLP (*Multi-Layer Perceptron*), composta por três camadas: camada de entrada, camada escondida e camada de saída. A arquitetura de RNA proposta é ilustrada na Figura 23. A camada de entrada é formada pela mesma quantidade de elementos do *array* do descritor 3D-CSD, enquanto a camada de saída deve possuir um neurônio para cada classe do sistema. Não há um modelo matemático que defina a quantidade ideal de neurônios na camada escondida, entretanto deve-se ponderar que com uma quantidade insuficiente de neurônios na camada escondida a rede neural é incapaz de modelar dados mais complexos, resultando em uma baixa capacidade de generalização. Por outro lado, se uma quantidade exagerada de neurônios for utilizada, o treinamento da rede irá se tornar excessivamente longo e a rede poderá sofrer problemas de sobreajuste ou “*overfitting*”, resultando na perda da capacidade preditiva da rede. Sendo assim, é necessário verificar de forma empírica a quantidade mínima de neurônios necessária para que o sistema consiga convergir para o mínimo erro de acordo com a aplicação.

Figura 23 – Arquitetura da Rede Neural utilizada.

Fonte: Elaborada pelo autor.

3.3.2 Geração do conjunto de treinamento

Para o treinamento da RNA deve ser gerado um conjunto de treinamento adequado. É necessário portanto coletar uma quantidade de amostras representativas de cada classe, aplicar o descritor 3D-CSD e em seguida rotular manualmente cada exemplo. Para obter resultados satisfatórios no treinamento da rede, deve ser coletada uma quantidade suficiente de exemplos para que a base de dados esteja balanceada, isto é, os dados devem ser coletados na mesma proporção para cada classe. Além disso, é importante ressaltar também que a robustez à rotação é obtida com a inclusão de exemplos em rotações variadas no conjunto de treinamento. Neste caso, é preciso gerar uma quantidade maior de exemplos para que o conjunto de treinamento seja eficiente. É possível gerar uma grande quantidade de exemplos a partir de um único exemplo (modelo 3D), variando a rotação do objeto em torno do seu centro de massa (em relação a um ou mais eixos) e computando o descritor 3D-CSD novamente para cada nova rotação realizada.

Uma vez coletados e rotulados esses dados, é realizado o treinamento em uma determinada quantidade de ciclos até que se alcance uma convergência, e por fim é feita uma validação que define se a RNA está pronta para o uso. Essa validação é feita pelo método de validação cruzada (HAYKIN, 1998), onde parte dos dados é destinada ao treinamento, e parte utilizada como teste. Este método permite avaliar se a rede realmente aprendeu as diferentes classes ensinadas e tem capacidade de generalizar ou se apenas “decorou” os dados utilizados no treinamento. Após o treinamento da RNA, os pesos são ajustados e o sistema está finalmente pronto para a tarefa de classificação.

3.4 Considerações Finais

Neste capítulo foi apresentada a metodologia proposta para classificação de objetos através da especificação de todas as suas etapas, desde o tratamento inicial dos dados do sensor até a classificação dos objetos detectados na cena. O sistema baseia-se em uma sequência de três etapas principais, sendo elas: segmentação dos dados (para identificação e separação das estruturas presentes no ambiente), extração de *features* (para representação simplificada e invariante desses elementos) e reconhecimento de padrões. O Algoritmo 4 descreve o ciclo completo de processamento para classificação.

Algoritmo 4 – *Pipeline* de classificação de objetos

- 1: **para todo** conjunto de dados do sensor em um dado instante **faça**
 - 2: Segmentar elementos da cena 3D
 - 3: **para todo** elemento segmentado **faça**
 - 4: Compute o descritor 3D-CSD
 - 5: Utilize os dados do descritor como entrada de uma RNA
 - 6: **retorna** a classe obtida na saída da RNA
 - 7: **fim para**
 - 8: **fim para**
-

Considerando um conjunto de pontos já segmentados, o processo de classificação pode ser realizado no pior caso em tempo $O(n^2)$ em relação à quantidade de pontos presentes na estrutura, resultante da soma dos tempos associados a cada etapa realizada: criação de superfícies (triangulação) em $O(n \log n)$; cálculo das distâncias da superfície em $O(n \times f)$ - onde n é o número de pontos de referência selecionados e f o número de faces geradas pela triangulação; interpolação linear em $O(n)$; e classificação neural em tempo constante.

O descritor 3D-CSD encontra-se na categoria de descritores globais, uma vez que a estrutura armazena informação referente ao objeto completo, e dentro desta categoria pode ser identificado como um modelo baseado em transformação, já que converte o conjunto de coordenadas inicial em um padrão definido e invariante (*array* de distâncias normalizadas do centro do objeto à sua superfície, na direção dos pontos de referência da esfera).

O modelo proposto é naturalmente invariante à translação, já que devido à segmentação e ao ajuste em relação ao centro de massa todos os objetos são descritos em relação ao referencial definido pelo seu centro. A interpolação linear cria a invariância à escala, graças ao uso da esfera de raio r para ajuste da escala dos objetos segmentados. Já a robustez à rotação é opcional, sendo obtida com a geração dos exemplos em rotações adicionais na etapa de aprendizado.

A Tabela 1 apresenta algumas características do descritor 3D-CSD em comparação com outros descritores citados no Capítulo 2. O símbolo $(+++)$ indica invariância intrínseca do método, $(+)$ indica invariância obtida através de alguma operação adicional, e $(-)$ indica que o método não possui invariância ao tipo de transformação citado.

Tabela 1 – Comparação entre descritores 3D

Descritor	Tipo	Classe	Invariância			Uso de Cor/Textura
			Tr	Es	Rot	
<i>Point Signatures</i>	Local	Assinatura	+++	-	+++	Não suportado
3D SURF	Local	Assinatura	-	+++	+++	Não necessita
HIGH	Local	Histograma	-	-	+++	Não necessita
FPFH	Local	Histograma	-	+	+++	Não suportado
<i>Spin Images</i>	Local	Histograma	+++	-	+++	Opcional
SHOT	Local	Híbridos	-	+	+++	Opcional
MeshHoG	Local	Híbridos	+++	+++	+++	Opcional
<i>Shape Distributions</i>	Global	Histograma	+++	+	+++	Não suportado
VFH	Global	Histograma	+	+	+++	Não suportado
<i>Spherical Harmonics</i>	Global	Transformação	+	+	+	Não necessita
SWD	Global	Transformação	+	+	+	Não necessita
SIFT	Global	Visão 2D	+	+++	+++	Obrigatório
<i>Fourier Descriptors</i>	Global	Visão 2D	+	+++	+++	Obrigatório
<i>Skeleton-Based</i>	Global	Grafos	+	+	+++	Não suportado
<i>Reeb-Graph</i>	Global	Grafos	+	+	+++	Não suportado
3D-CSD	Global	Transformação	+++	+++	+	Não necessita

O descritor 3D-CSD proposto nesta tese obtém a invariância à rotação através de uma operação adicional (geração automática de novos exemplos com diferentes rotações), entretanto o sistema completo de classificação de objetos pode ser considerado um método completo e equivalente em termos de invariância à rotação (+++) em relação aos demais métodos. Além disto, o método proposto não necessita de informações de cor e textura para realizar a classificação dos objetos, considerando apenas a distribuição espacial dos pontos 3D. Entretanto, é possível em termos de trabalhos futuros, que se pense em estender o descritor 3D-CSD com a inclusão de informações de cor e textura referentes às respectivas faces usadas no cálculo das distâncias.

Os processos descritos neste capítulo correspondem à proposta geral do sistema baseado no descritor 3D-CSD, representando o processo conforme a sua concepção. Cada aplicação específica possui um conjunto de premissas e particularidades que definem as estratégias de implementação a serem adotadas bem como adaptações ao modelo proposto. As adaptações necessárias para implementação do sistema em aplicações de navegação em Robótica, e os experimentos realizados para validação da metodologia proposta neste tipo específico de abordagem são discutidos no próximo capítulo.

EXPERIMENTOS E RESULTADOS

Este capítulo apresenta a configuração do sistema proposto para classificação de objetos em nuvem de pontos no contexto de aplicações para navegação topológica e os experimentos realizados para validação e avaliação do sistema proposto.

4.1 Adaptações voltadas à aplicações em Robótica

Ao contrário de aplicações de modelagem e reconstrução 3D, onde a nuvem de pontos final é formada pela composição de detecções do objeto em diferentes ângulos, nas aplicações de navegação autônoma nem sempre é possível obter a informação completa dos objetos, devido à própria disposição física e deslocamento do sensor em relação ao ambiente. Como os pontos são obtidos pela reflexão dos feixes de laser (ou infravermelho) emitidos, não é possível obter informação sobre as faces não visíveis pelo sensor em uma única coleta. Isso cria a impressão de “sombras” na nuvem de pontos, decorrentes da oclusão natural causada pela presença de um elemento no ambiente, como pode ser observado na Figura 24.

Figura 24 – Detalhe de objeto detectado parcialmente na nuvem de pontos.

Fonte: Elaborada pelo autor.

Dada essa característica da aplicação, a região segmentada que contém os pontos de cada objeto possui informação apenas da região visível pelo sensor no instante da coleta dos dados. Com isso surgiu a necessidade de realizar adaptações em algumas etapas da metodologia proposta, adequando o sistema a essas condições.

A adaptação mais importante diz respeito ao posicionamento da esfera e escolha dos pontos de referência para o cálculo das distâncias que descrevem a superfície. Como os pontos segmentados correspondem a aproximadamente metade do objeto, o centro de massa real do objeto não coincide com o centro de massa do conjunto de pontos observados. Portanto, o centro da esfera é posicionado no plano imaginário que divide a parte visível pelo sensor da parte não visível, no ponto médio da projeção 2D dos pontos do objeto nesse plano, conforme ilustra a Figura 25.

Figura 25 – Posicionamento do centro da esfera em relação aos pontos 3D segmentados.

Fonte: Elaborada pelo autor.

Uma vez posicionado o centro da esfera é definido o raio (distância ao ponto do conjunto mais afastado do centro), e então a distribuição dos pontos de referência é feita apenas na semi-esfera que contém a região visível do objeto. Assim, o padrão representado pelo descritor 3D torna-se compatível com as representações parciais obtidas das leituras do sensor durante a navegação.

Seguindo este padrão, a geração do conjunto de dados de treinamento pode ser feita com modelos 3D completos (exemplos gerados pelo supervisor do sistema) ou modelos parciais (obtidos dos dados reais), desde que mantida a característica de geração do descritor com distribuição dos pontos de referência sempre no mesmo intervalo (semi-esfera). O conjunto de treinamento deve incluir a maior quantidade possível de variações de rotação dos elementos, garantindo que todas as classes sejam representadas por completo no conjunto.

Uma outra característica particular das aplicações em robótica diz respeito ao tipo de rotação dos elementos. Usualmente, os elementos estarão dispostos no ambiente com variações de rotação significativas apenas em relação ao eixo z . Sendo assim, não é necessário incluir no conjunto de treinamento elementos com rotações nos demais eixos.

4.2 Treinamento da RNA e validação dos resultados

Para geração e treinamento das Redes Neurais utilizadas no experimentos foi utilizada a ferramenta JavaNNS¹, que é uma interface gráfica desenvolvida em Java para o simulador SNNS (Stuttgart Neural Network Simulator)². O algoritmo de treinamento adotado é o *Resilient Propagation* (RIEDMILLER; BRAUN, 1994), uma vez que este tem apresentado um tempo de convergência inferior em relação ao *backpropagation* tradicional.

Para validação do classificador nos experimentos foi adotado o método *k-fold* de validação cruzada (REZENDE, 2003), que consiste em dividir o conjunto total de dados em k subconjuntos de tamanhos iguais mutuamente exclusivos, e utilizar $k - 1$ conjuntos para treinamento, deixando o conjunto restante para teste. O processo é realizado k vezes, alternando de forma circular o conjunto utilizado para teste, como ilustrado na Figura 26.

Figura 26 – Esquema de particionamento e execução da validação cruzada com $k = 3$.

Fonte: Elaborada pelo autor.

A acurácia final do sistema é obtida a partir da média das acurácias obtidas nos k treinamentos da RNA. Este procedimento garante uma medida mais confiável sobre a capacidade da RNA de generalizar o conhecimento a partir dos exemplos apresentados, e permite avaliar ainda se os exemplos fornecidos são representativos o suficiente, com base na diferença entre a acurácia obtida em cada execução.

É interessante também observar as matrizes de confusão (REZENDE, 2003) resultantes de cada iteração, onde é possível avaliar o erro por classe e identificar com que frequência o sistema classifica erroneamente uma classe, e entre quais classes esse erro é maior. Nos experimentos realizados foi adotado $k = 5$, o que resulta em 80% dos dados fornecidos usados para treinamento e 20% para teste em cada iteração.

¹ <<http://www.ra.cs.uni-tuebingen.de/downloads/JavaNNS/>>

² <<http://www.ra.cs.uni-tuebingen.de/SNNS/>>

4.3 Validação inicial do descritor 3D

Com o objetivo de avaliar a descritividade do descritor 3D-CSD foi desenvolvido um sistema de classificação envolvendo seis classes de objetos convexos, ilustrados na Figura 27: a) cone, b) cubo, c) icosaedro, d) toróide, e) octaedro e f) esfera (baixa-resolução). Os vértices e faces foram gerados manualmente, dispensando as etapas de segmentação e triangulação apresentadas na metodologia. A tabela 2 apresenta a quantidade de vértices, faces e o tempo de geração³ do descritor 3D-CSD para os elementos de cada classe.

Figura 27 – Pontos 3D e faces dos objetos convexos utilizados nos testes.

Fonte: Elaborada pelo autor.

Tabela 2 – Características das 6 classes de objetos utilizados.

Classe	Vértices	Faces	Tempo(ms)
Cone	17	30	24
Cubo	8	12	9
Icosaedro	12	20	16
Toróide	40	80	64
Octaedro	6	8	6
Esfera	42	80	64

A distribuição dos pontos de referência foi feita já de acordo com a configuração descrita na seção 4.1, considerando portanto uma abrangência de 180° nos sentidos vertical e horizontal (semi-esfera). Os pontos foram posicionados em intervalos de 10° em ambos os sentidos (vertical e horizontal), resultando em 326 pontos de referência. O conjunto de treinamento foi formado com a geração de 100 exemplos de cada classe. Cada exemplo é criado através da rotação do objeto em um ângulo aleatório (em um ou mais eixos). Desta forma, cada um dos conjuntos de treinamento e teste é composto de 480 e 120 exemplos, respectivamente. Por fim, as distâncias calculadas foram interpoladas para o intervalo contínuo $\{0, 1\}$.

³ em um processador Intel Core i7-3520M @2.90Ghz, com sistema operacional Windows 8 64-bit

A quantidade de neurônios na camada escondida da RNA foi definida com base em testes empíricos realizados previamente, resultando em uma RNA de topologia 326-56-6, com todos os neurônios com função de ativação logística, de forma a produzir na saída valores reais entre 0 e 1. O processo de treinamento foi realizado com os parâmetros padrão do R-Prop no JavaNNS ($\delta 0 = 0.1$, $\delta \max = 50$ e $\alpha = 4.0$).

Em todos os cinco testes da validação cruzada o erro médio quadrático da RNA convergiu para menos de 0.05 em menos de 200 ciclos de treinamento. A Figura 28 apresenta as curvas de erro resultantes dos processos de treinamento (cada cor corresponde a um dos cinco testes), e a Tabela 3 exibe as matrizes de confusão para cada conjunto de teste, onde foi obtida uma acurácia média de 99.5%.

Figura 28 – Erro médio quadrático (MSE) da RNA 326-56-6 nos 5 testes.

Fonte: Elaborada pelo autor.

Tabela 3 – Matrizes de confusão resultantes para as 6 classes de objetos convexos.

Teste 1						Teste 2						Teste 3					
20	0	0	0	0	0	20	0	0	0	0	0	19	1	0	0	0	0
0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0	0	0
0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0	0
0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0
0	0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0
0	0	0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20
Teste 4						Teste 5											
18	1	0	0	1	0	20	0	0	0	0	0	20	0	0	0	0	0
0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0	0	0
0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0	0
0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0	0
0	0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20	0
0	0	0	0	0	20	0	0	0	0	0	20	0	0	0	0	0	20

Os resultados obtidos neste experimento demonstraram a capacidade do descritor 3D-CSD de armazenar informação representativa da superfície dos objetos, conseguindo diferenciar com precisão mesmo elementos de estrutura geométrica relativamente similar, como o icosaedro e a esfera de baixa resolução. Os resultados foram apresentados no artigo “3D feature extraction for objects recognition”, publicado no *IEEE Latin American Robotics Symposium (LARS) 2017* (SALES; AMARO; OSÓRIO, 2017).

4.4 Classificação de obstáculos

Uma vez avaliada a descritividade do descritor 3D proposto, foi avaliado o comportamento do sistema na classificação de elementos com estrutura mais complexa, cuja presença é comum nas aplicações de navegação autônoma: construções, pessoas, árvores, motos e carros. A Figura 29 ilustra os vértices e faces dos modelos 3D utilizados neste experimento, e a Tabela 4 apresenta a quantidade de vértices, faces e o tempo de geração do descritor em cada classe, no mesmo equipamento do experimento anterior.

Figura 29 – Pontos 3D e faces dos objetos utilizados nos testes.

Fonte: Elaborada pelo autor.

Tabela 4 – Características das 5 classes de objetos utilizados.

Classe	Vértices	Faces	Tempo(ms)
Construção	16	28	33
Pessoas	738	1455	158
Árvore	53589	65533	5769
Moto	22183	33994	3007
Carro	839	1410	154

Manteve-se a mesma configuração de distribuição de pontos de referência do experimento anterior, com 326 pontos distribuídos em intervalos de 10° nos sentidos vertical e horizontal da semi-esfera. A quantidade de neurônios na camada escondida da RNA foi ajustada para 50 neurônios, gerando assim uma RNA com topologia 326-50-5 (todos os neurônios com função de ativação logística).

O conjunto de treinamento foi formado utilizando objetos no formato *wavefront* (.obj) obtidos de um repositório online de modelos 3D⁴. Foram gerados 100 exemplos de cada classe, derivados de rotações em ângulos aleatórios nos três eixos. Considerando a divisão para validação cruzada, os conjuntos de treinamento e teste ficaram com 400 e 100 exemplos cada. A RNA foi treinada com os mesmos parâmetros do experimento anterior (R-Prop, com $\delta_0 = 0.1$, $\delta_{\max} = 50$ e $\alpha = 4.0$). A Figura 30 apresenta a convergência do erro médio quadrático e a Tabela 5 exhibe as matrizes de confusão resultantes.

Figura 30 – Erro médio quadrático (MSE) da RNA 326-50-5 nos 5 testes.

Fonte: Elaborada pelo autor.

Tabela 5 – Matrizes de confusão resultantes para as 5 classes de objetos utilizadas.

Teste 1					Teste 2					Teste 3				
20	0	0	0	0	20	0	0	0	0	20	0	0	0	0
0	19	0	1	0	0	20	0	0	0	0	18	0	0	2
0	8	10	1	1	0	5	11	3	1	0	5	15	0	0
0	0	0	20	0	0	0	0	20	0	0	0	0	20	0
0	2	0	0	18	0	0	0	0	20	0	3	0	0	17

Teste 4					Teste 5				
20	0	0	0	0	20	0	0	0	0
0	20	0	0	0	0	18	0	2	0
1	6	13	0	0	1	3	14	2	0
0	0	0	20	0	0	0	0	20	0
0	0	0	0	20	0	2	0	0	18

⁴ <<http://www.cgtrader.com/>>

A acurácia final do sistema ficou em 90,2%, entretanto é possível concluir com a análise das matrizes de confusão que 75% dos erros de classificação ocorreram na classe “árvore”, indicando um problema com os exemplos dessa classe em especial.

Neste caso, além de uma alta quantidade de faces, as mesmas não apresentavam relação direta com a distribuição volumétrica do contorno do objeto, deixando intervalos vazios no modelo 3D final, o que causou um alto ruído no descritor 3D. Desta forma, o experimento foi repetido, com a substituição do modelo para representação da classe “árvore”, conforme ilustrado na Figura 31. O novo modelo consiste em uma representação menos sujeita a ruídos por ser um sólido convexo, possuindo 56 vértices e 192 faces, com geração do descritor 3D-CSD em 47ms. As matrizes de confusão resultantes do novo treinamento do sistema são apresentadas na Tabela 6.

Figura 31 – Modelos 3D utilizados para representação da classe “árvore”.

Fonte: Elaborada pelo autor.

Tabela 6 – Matrizes de confusão resultantes do novo treinamento da RNA.

Teste 1					Teste 2					Teste 3				
20	0	0	0	0	20	0	0	0	0	20	0	0	0	0
0	17	0	0	3	0	18	0	1	1	0	14	0	0	6
0	0	20	0	0	0	0	20	0	0	0	0	20	0	0
0	1	0	19	0	0	0	0	20	0	0	0	0	20	0
0	1	0	0	19	0	2	0	0	18	0	3	0	0	17

Teste 4					Teste 5				
20	0	0	0	0	20	0	0	0	0
0	17	0	1	2	0	15	0	2	3
0	0	20	0	0	0	0	20	0	0
0	0	0	20	0	0	0	0	20	0
0	0	0	0	20	0	2	0	0	18

Com o novo treinamento da RNA o erro da classe “árvore” foi eliminado, o que elevou a acurácia final para 94,4%. Este experimento demonstrou a viabilidade da aplicação da metodologia proposta no reconhecimento de elementos frequentemente presentes nos ambientes reais das aplicações de navegação autônoma.

A Tabela 7 apresenta uma comparação entre a acurácia obtida por alguns classificadores na literatura voltados ao mesmo tipo de aplicação. Os resultados obtidos nos experimentos descritos neste capítulo são similares aos dos trabalhos de Douillard *et al.* (2014), Zhu *et al.* (2010) e Habermann *et al.* (2013), que utilizaram conjuntos semelhantes de classes nos seus experimentos ({carros - postes - árvores - pessoas}, {construção - vegetação - carros - árvores - pedestres - bicicletas} e {carros - pessoas - muros - árvores} respectivamente).

Tabela 7 – Comparação entre acurácia obtida nos trabalhos relacionados.

Trabalho	Método	Origem dos dados	Exemplos	Classes	Acurácia
Zhu <i>et al.</i> (2010)	<i>Range Images</i>	5 SICK LiDARs	1414	6	89,56%
Douillard <i>et al.</i> (2014)	<i>3D Template Matching</i>	Velodyne HDL-64	38	4	92,1%
Habermann <i>et al.</i> (2013)	Projeção 2D dos pontos	Velodyne HDL-32	2365	4	88,1%
Esta tese	Descritor 3D-CSD	Modelos 3D (low-poly)	500	5	94,4%

Zhu *et al.* (2010) cria uma *Range Image* a partir dos dados de cinco sensores laser 2D combinados, aplica um algoritmo de segmentação e em seguida calcula descritores 3D locais nos pontos segmentados, armazenando *features* como os vetores normais aos planos formados pelos pontos e seus vizinhos. Uma vez computados os descritores, eles são utilizados como entrada para uma etapa de classificação usando *Support Vector Machines* (SVM).

Já no trabalho de Douillard *et al.* (2014), após a segmentação da nuvem de pontos, os dados são comparados com modelos pré-definidos de cada classe, em uma abordagem de *Template Matching*. Usa-se o algoritmo ICP (*Iterative Closest Point*) para alinhar as duas nuvens de pontos, e então é utilizada uma métrica para calcular a similaridade entre o modelo de uma classe e o objeto. Foi alcançada a mesma acurácia do melhor método baseado em extração de *features* (*Spin Images*) no mesmo conjunto de dados em um tempo consideravelmente menor. Ainda assim, o algoritmo ICP tem um custo computacional elevado, inviabilizando seu uso em aplicações onde a classificação deve ser feita em tempo-real, como a navegação autônoma.

Habermann *et al.* (2013) cria um *grid* de 64 células a partir da projeção 2D do objeto segmentado, e utiliza uma RNA para reconhecer os padrões representados nesse *grid*. Dentre as três abordagens citadas, é o método de menor custo computacional, entretanto a conversão 3D/2D e a redução de resolução oriunda da representação simplificada no *grid* diminuem a descritividade do modelo, limitando o escopo das aplicações possíveis.

A implementação baseada no uso do descritor 3D-CSD proposta nesta tese alcançou níveis de acurácia adequados para a classificação, mesmo com poucos ciclos de treinamento e com uma quantidade reduzida de exemplos, demonstrando a representatividade associada ao descritor 3D-CSD. Por ser uma abordagem baseada no uso de um descritor global, a metodologia proposta está sujeita a uma dimensionalidade menor do que a de sistemas baseados em extração de *features* locais, o que resulta em um menor tempo de execução.

4.5 Classificação de dados obtidos do sensor Velodyne

Após a avaliação do descritor e do classificador nas tarefas de reconhecimento de objetos 3D dos experimentos anteriores, foi avaliado o comportamento do sistema com o uso de dados obtidos do sensor LiDAR 3D Velodyne HDL-64E. Para isto, foi utilizada a base de dados de [Douillard *et al.* \(2014\)](#), que contém 88 exemplos pré-segmentados e rotulados como pertencentes à classe “veículo”. Como cada exemplo é um conjunto de pontos obtidos em uma leitura simples do sensor, os dados só fornecem uma vista parcial do objeto, e por isso muitos exemplos não apresentam uma disposição que resulte em uma estrutura com distribuição volumétrica similar a de um veículo (figura 32), o que comprometeria os resultados.

Figura 32 – Exemplos rotulados que não se adequam ao modelo proposto.

Fonte: Elaborada pelo autor.

Desta forma, 8 exemplos da classe “veículo” foram selecionados, assim como em [Douillard *et al.* \(2014\)](#). A figura 33 mostra o resultado do *Convex Hull 3D* em um dos exemplos utilizados em comparação ao modelo utilizado no treinamento da RNA, enquanto a figura 34 ilustra os conjuntos de pontos dos demais exemplos selecionados.

Figura 33 – Resultado do *Convex Hull 3D* no modelo de treinamento e em um exemplo da base de dados.

Fonte: Elaborada pelo autor.

Figura 34 – Nuvens de pontos dos exemplos utilizados no experimento.

Fonte: Elaborada pelo autor.

Para a classificação foi utilizada a mesma RNA do experimento anterior, treinada com os modelos em formato *wavefront* (.obj). Foi aplicado então o descritor 3D-CSD nos exemplos selecionados utilizando os mesmos parâmetros e intervalos para produzir as entradas da RNA. Conforme discutido na seção 4.1, as nuvens de pontos foram inscritas em uma semi-esfera, cujo centro coincide com o plano imaginário que divide a parte visível da parte não-visível do objeto.

Todos os oito exemplos foram corretamente classificados, o que demonstrou a viabilidade do uso do descritor proposto em dados 3D esparsos, possibilitando assim a implementação de sistemas de percepção para aplicações de navegação autônoma. Os resultados destes experimentos foram submetidos para o periódico *International Journal of Computer Vision (IJCV)*.

4.6 Limitações do sistema

A principal limitação do sistema está relacionada à própria natureza da *feature* 3D-CSD: por se tratar de uma *feature* global, problemas relacionados à etapa de segmentação como oclusão (levando a uma representação incompleta) ou adjacência (incapacidade de segmentar os objetos em estruturas distintas) podem levar a uma representação inadequada da forma do objeto, gerando erros de classificação.

A segmentação também é um ponto crítico do ponto de vista do tempo de execução. Métodos como K-Means e KNN são executados em tempo $O(n^{3k+1})$, onde n é a quantidade total de pontos na nuvem a serem segmentados e k a quantidade de grupos. A aplicação em sistemas de navegação autônoma requer o uso de métodos eficientes de segmentação, ou mesmo aceleração em GPU (*Graphics Processing Unit*) como a abordagem apresentada por [Christino e Osório \(2016\)](#) para processamento dos dados em um tempo adequado.

Outro ponto a ser observado é o tipo de triangulação utilizada para geração de faces. Nos sensores onde a nuvem de pontos é mais densa (como o Kinect), é necessário utilizar algoritmos que diminuam a quantidade de faces geradas, uma vez que o tempo de processamento para geração do descritor é proporcional à quantidade de faces, podendo inviabilizar a aplicação do método em sistemas de tempo-real. Já no caso de sensores com nuvens de pontos esparsas (como o Velodyne), ocorre o problema inverso: os objetos são representados com menos pontos à medida que se afastam do sensor, logo, é necessário adotar modelos de triangulação capazes de criar a superfície da forma mais precisa possível para que o descritor extraído seja representativo o suficiente.

Com relação à classificação final, é importante que o treinamento da RNA seja feito com exemplos representativos, que possuam as principais características esperadas de cada classe. Considerando a restrição de distribuição dos pontos chave em uma semi-esfera imposta pela especificidade dos sensores 3D utilizados em robótica, os conjuntos de treinamento precisam englobar elementos em diversas rotações, de forma que não existam faces completamente desconhecidas no modelo final.

4.7 Considerações finais

Este capítulo apresentou os experimentos realizados para validação da metodologia e avaliação do descritor 3D proposto, além das principais restrições e adaptações implementadas para adequação do sistema a aplicações de navegação autônoma.

O descritor 3D proposto apresentou boa capacidade de representação das características estruturais do contorno dos objetos, tanto no conjunto de formas convexas como no de obstáculos reais, possibilitando o aprendizado e reconhecimento dos padrões dos objetos usando Redes Neurais Artificiais.

O classificador desenvolvido apresentou níveis de acurácia similares ao de outros trabalhos bem-sucedidos voltados à mesma aplicação, o que sugere que a metodologia é adequada tanto para o reconhecimento de objetos no nível local de navegação (detecção semântica de obstáculos para desvio ou interação) como global (reconhecimento de estruturas específicas utilizadas como pontos de referência do ambiente).

CONCLUSÃO

Os estudos realizados durante este projeto de doutorado possibilitaram o desenvolvimento de uma metodologia para classificação de objetos em nuvens de pontos, com a proposta e implementação de um novo descritor de *features* 3D, utilizado para a representação aproximada da assinatura formada pelos contornos tridimensionais dos objetos.

O sistema não necessita de informações adicionais como textura e cor, utilizando apenas a informação de posição e profundidade proveniente dos pontos 3D detectados como entrada. Assim, não há demanda por métodos de fusão de sensores ou pelo uso de um tipo de sensor específico (como RGB-D) para extração da *feature* 3D-CSD, o que viabiliza a compatibilidade com sensores de diferentes categorias e propósitos.

O descritor 3D desenvolvido permite a redução da dimensionalidade do conjunto de dados, uma vez que conjuntos com quantidades variáveis de coordenadas tridimensionais são convertidos para uma quantidade fixa de distâncias medidas em pontos pré-estabelecidos, representando os objetos de forma simplificada e invariante. O sistema desenvolvido é naturalmente invariante à translação, uma vez que utiliza os dados já segmentados do ambiente e assume como referência o centro de massa do objeto; invariante à escala, devido à interpolação linear das distâncias calculadas; e a invariância à rotação pode ser obtida na etapa de aprendizado, com a criação de exemplos adicionais na elaboração do conjunto de treinamento do sistema. A geração de exemplos permite inclusive a formação de uma base de treinamento robusta, mesmo com uma quantidade inicial de dados limitada.

Utilizando o contexto de aplicações em navegação autônoma como referência, o estudo desenvolvido nesta tese teve foco apenas no sistema de percepção, visando a geração de informação semântica útil para aplicações de navegação em robótica: detecção de estados em mapas topológicos através do reconhecimento de pontos de referência do ambiente e/ou detecção semântica de obstáculos, não abordando portanto questões específicas de planejamento de trajetória ou do controle de navegação, já abordadas em trabalhos anteriores.

Os experimentos realizados para avaliação do sistema de classificação alcançaram índices de acurácia convenientes para a classificação de elementos representados na forma de nuvem de pontos, alinhados aos resultados obtidos em trabalhos com escopos similares, o que demonstra que a abordagem utilizada é promissora, além de ser compatível com diferentes sensores 3D frequentemente utilizados nas aplicações de Robótica Móvel.

5.1 Contribuições

Esta tese apresentou contribuições ao estado-da-arte em visão computacional 3D e sistemas de percepção para robôs móveis e veículos autônomos com a proposta do novo descritor 3D-CSD¹ e da metodologia de classificação desenvolvida, além de contribuir com os projetos em execução no Laboratório de Robótica Móvel do ICMC-USP, CRob-USP e InSac, uma vez que a metodologia desenvolvida pode ser empregada no desenvolvimento de sistemas de percepção tanto de robôs táticos para ambientes internos como de veículos terrestres autônomos.

No âmbito de contribuições gerais, além do próprio descritor 3D-CSD apresentado é possível destacar também as adaptações voltadas à integração do sistema em aplicações de navegação autônoma, como a técnica alternativa de construção do descritor utilizando pontos distribuídos em uma semi-esfera, a possibilidade de redução na quantidade de pontos processados com o uso do *Convex Hull 3D* na etapa de triangulação, e a extensão da base de treinamento através da geração de novos exemplos com a variação na rotação dos modelos. A metodologia desenvolvida é modular e parametrizável, sendo portanto adaptável a outras aplicações de classificação em áreas não exploradas nesta tese.

Já no contexto dos trabalhos em andamento no LRM, este trabalho é complementar aos estudos realizados pelo grupo em diversas frentes: Christino e Osório (2016) desenvolveram um sistema para tratamento dos dados provenientes do sensor LiDAR usando serviços em GPU, resultando em um método de segmentação de objetos em nuvem de pontos. Considerando a metodologia apresentada nesta tese, a extração de *features* e classificação correspondem às etapas seguintes a esse sistema, aproveitando o ganho de desempenho obtido na segmentação, que é uma das etapas mais críticas em termos de tempo de execução.

Habermann *et al.* (2016) desenvolveram um sistema de navegação topológica que determina o estado atual de um veículo autônomo através da detecção de cruzamentos e guias. O sistema de classificação desenvolvido nesta tese complementa esta abordagem, viabilizando a classificação de obstáculos e reconhecimento de pontos de referência mais específicos como monumentos e placas, aumentando o grau de confiança do estado detectado e provendo tolerância a eventuais falhas.

¹ Implementação do descritor 3D-CSD em linguagem C++ disponível em:
<<https://github.com/daniel-sales/3d-csd-descriptor>>

Rosero e Osório (2016) desenvolveram um sistema de percepção baseado na fusão de Radar e Câmera, onde a detecção dos obstáculos pode ser realizada com a metodologia apresentada nesta tese. Há ainda trabalhos focados na detecção de um tipo de objeto específico como o projeto de doutorado de Diego Bruno, que propõe a identificação de placas de trânsito em ambientes 3D, ou ainda os trabalhos de Ridet, Shinzato e Wolf (2015), voltados à detecção e rastreamento de pedestres para estimação de sua trajetória e localização para desvio inteligente de obstáculos. Neste contexto, o sistema apresentado nesta tese é útil e diretamente aplicável, estando portanto estreitamente ligado aos demais trabalhos do LRM.

5.2 Trabalhos Futuros

Como continuação aos estudos desenvolvidos nesta tese há a possibilidade de aperfeiçoamentos no descritor 3D-CSD, como a implementação de invariância à rotação na própria representação do descritor com o uso de histogramas de assinaturas, além da análise da melhor distribuição dos pontos de referência, visando otimizar a descritividade obtida em conjuntos de dados específicos e assim viabilizar a implementação de sistemas como o reconhecimento de gestos e pessoas.

No contexto de aplicações de navegação, a integração com os trabalhos anteriores voltados à navegação topológica em aplicações reais é uma sequência natural, criando implementações que possam tratar informações estruturais do ambiente e objetos presentes na cena 3D simultaneamente. Ainda nesse escopo, é possível incluir questões como o tratamento de oclusão parcial na etapa de segmentação e otimizações ligadas ao tempo de execução.

A criação de uma base de dados para avaliação de classificadores baseados em descritores 3D também é uma demanda no momento, uma vez que os conjuntos de dados disponíveis atualmente ainda são voltados para aplicações muito específicas e apresentam limitações.

Outro aspecto interessante que pode ser explorado é o uso de *Deep Learning* para a classificação dos objetos 3D, uma vez que o método desenvolvido possui características favoráveis à implementação deste tipo de técnica, como a geração de vários exemplos adicionais a partir de um conjunto inicial (formando uma base de dados em *Big Data*) e a própria arquitetura MLP do classificador utilizado, que já corresponde a um modelo “*DeepNN*” mais simples. Uma vez que o descritor 3D-CSD armazena informações sobre o contorno tridimensional dos objetos, o estudo de formas de implementação da convolução em três dimensões tem um grande potencial. Além disto, podem ser incorporadas informações locais de cor e textura referentes aos pontos e faces considerados na geração do descritor 3D-CSD para o reconhecimento dos objetos, estendendo o método proposto.

REFERÊNCIAS

- ABELLAN, A.; DERRON, M.; JABOYEDOFF, M. Use of 3d point clouds in geohazards. **Remote Sensing**, v. 8, 2016. Citado na página 37.
- ABIAN TUN, R.; PRABHU, U.; SAVVIDES, M. Sparse feature extraction for pose-tolerant face recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 36, n. 10, 2014. Citado na página 37.
- AKGUL, C. B.; SANKUR, B.; YEMEZ, Y.; SCHMITT, F. 3d model retrieval using probability density-based shape descriptors. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 31, n. 6, p. 1117–1133, 2009. Citado na página 35.
- ALDOMA, A.; BLODOW, N.; GOSSOW, D.; GEDIKLI, S.; RUSU, R.; VINCZE, M.; BRADSKI, G. Cad-model recognition and 6 dof pose estimation using 3d cues. In: **ICCV 2011, 3D Representation and Recognition (3dRR11)**. [S.l.: s.n.], 2011. Citado na página 35.
- ALI, A. M. A 3d-based pose invariant face recognition at a distance framework. **IEEE Transactions on Information Forensics and Security**, v. 9, n. 12, 2014. Citado na página 37.
- ALI, H.; MARTÓN, Z. Evaluation of feature selection and model training strategies for object category recognition. In: **Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2014. Citado na página 38.
- ANKERST, M.; KASTENMULLER, G.; KRIEGEL, H.-P.; SEIDL, T. 3d shape histograms for similarity search and classification in spatial databases. In: **Proc. Sixth Int'l Symp. Advances in Spatial Databases**. [S.l.: s.n.], 1999. p. 207–226. Citado na página 35.
- BAY, H.; ESS, A.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. **Computer Vision and Image Understanding (CVIU)**, 2008. Citado na página 33.
- BECK, J.; BROGGI, A.; SUNDARESWAN, V.; DERRICK, B. **Team oshkosh - darpa urban challenge**. [S.l.], 2007. Disponível em: http://www.darpa.mil/grandchallenge/Tech_Papers/Team_Oshkosh.pdf. Citado na página 23.
- BERG, M. de; CHEONG, O.; KREVELD, M. van; OVERMARS, M. **Computational Geometry - Algorithms and Applications**. [S.l.]: Springer, 2008. Citado na página 46.
- BOUWMANS, T.; PORIKLI, F.; HÖFERLIN, B.; VACAVANT, A. **Background Modeling and Foreground Detection for Video Surveillance**. [S.l.]: CRC Press - Taylor & Francis Group, 2015. ISBN 978-1-4822-0538-1. Citado na página 44.
- BUEHLER, M.; IAGNEMMA, K.; SINGH, S. **The 2005 darpa grand challenge: The great robot race**. [S.l.: s.n.], 2007. Citado na página 23.
- CANELHAS, D. R.; STOYANOV, T.; LILIENTHAL, A. J. From feature detection in truncated signed distance fields to sparse stable scene graphs. **IEEE Robotics and Automation Letters**, v. 1, n. 2, 2016. Citado na página 38.

CANNY, J. A computational approach to edge detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 1986. Citado na página 81.

CHAN, K.; KOH, C.; LEE, C. S. G. A 3d-point-cloud feature for human-pose estimation. In: **Proceedings of 2013 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2013. Citado na página 37.

CHEN, D.-Y.; TIAN, X.-P.; SHEN, Y.-T.; OUHYOUNG, M. On visual similarity based 3d model retrieval. **Computer Graphics Forum**, v. 22, p. 223–232, 2003. Citado na página 36.

CHENG, H.; CHEN, H.; LIU, Y. Topological indoor localization and navigation for autonomous mobile robot. **IEEE Transactions on Automation Science and Engineering**, v. 12, n. 2, p. 729 – 738, 2015. Citado na página 29.

CHRISTINO, L.; OSÓRIO, F. Gpu-services: Gpu based real-time processing of 3d point clouds applied to robotic systems and intelligent vehicles. In: _____. **Robotics: 12th Latin American Robotics Symposium and Third Brazilian Symposium on Robotics, LARS 2015/SBR 2015, Uberlândia, Brazil, October 28 - November 1, 2015, Revised Selected Papers**. Cham: Springer International Publishing, 2016. p. 152–171. ISBN 978-3-319-47247-8. Disponível em: <http://dx.doi.org/10.1007/978-3-319-47247-8_10>. Citado nas páginas 65 e 68.

CHUA, C. S.; JARVIS, R. Point signatures: a new representation for 3d object recognition. **International Journal of Computer Vision**, v. 25, n. 1, p. 63–85, 1997. Citado na página 33.

DARAS, P.; ZARPALAS, D.; TZOVARAS, D.; STRINTZIS, M. G. Shape matching using the 3d radon transform. In: **Proceeding of Second International Symposium on 3D Data Processing, Visualization, and Transmission**. [S.l.: s.n.], 2004. p. 953–960. Citado na página 35.

DOUILLARD, B.; UNDERWOOD, J.; VLASKINE, V.; QUADROS, A.; SINGH, S. A pipeline for the segmentation and classification of 3d point clouds. In: _____. **Experimental Robotics: The 12th International Symposium on Experimental Robotics**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 585–600. ISBN 978-3-642-28572-1. Citado nas páginas 37, 44, 63 e 64.

DUDEK, G.; JENKIN, M. **Computational principles of mobile robotics**. [S.l.]: Cambridge University Press, 2000. Citado na página 23.

DUTAGACI, H.; SANKUR, B.; YEMEZ, Y. Transform-based methods for indexing and retrieval of 3d objects. In: **Proceedings of Fifth International Conference on 3D Digital Imaging and Modeling**. [S.l.: s.n.], 2005. Citado na página 35.

FROME, A.; HUBER, D.; KOLLURI, R.; BULOW, T.; MALIK, J. Recognizing objects in range data using regional point descriptors. In: **Proceedings of European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2004. v. 3, p. 224–237. Citado na página 34.

GAO, Y.; DAI, Q. View-based 3d object retrieval: Challenges and approaches. **IEEE Multimedia**, 2014. Citado na página 38.

GERÓNIMO, D.; LÓPEZ, A. M.; SAPPA, A. D.; GRAF, T. Survey of pedestrian detection for advanced driver assistance systems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 32, n. 7, 2010. Citado na página 37.

GUO, Y.; BENNAMOUN, M.; SOHEL, F.; LU, M.; WAN, J.; KWOK, N. M. A comprehensive performance evaluation of 3d local feature descriptors. **International Journal of Computer Vision**, p. 66–89, 2016. Citado nas páginas 22 e 38.

HABERMANN, D.; HATA, A.; WOLF, D.; OSÓRIO, F. 3d point clouds segmentation for autonomous ground vehicle. In: **Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC)**. [S.l.: s.n.], 2013. Citado na página 63.

HABERMANN, D.; VIDO, C.; OSÓRIO, F. S.; RAMOS, F. Road junction detection from 3d point clouds. In: **Proceedings of International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2016. Citado nas páginas 37 e 68.

HAN, J.; SHAO, L.; XU, D.; SHOTTON. Enhanced computer vision with microsoft kinect sensor: A review. **IEEE Transactions on Cybernetics**, v. 43, n. 5, 2013. Citado na página 31.

HAYKIN, S. **Neural Networks: A Comprehensive Foundation**. 2nd. ed. [S.l.]: Prentice Hall, 1998. Citado nas páginas 40, 41 e 51.

HILAGA, M.; SHINAGAWA, Y.; KOHMURA, T.; KUNII, T. L. Topology matching for fully automatic similarity estimation of 3d shapes. In: **Proceedings of ACM SIGGRAPH**. [S.l.: s.n.], 2001. p. 203–212. Citado na página 37.

HORN, B. K. P. Extended gaussian images. In: **Proc. IEEE**. [S.l.: s.n.], 1984. v. 72, p. 1671–1686. Citado na página 35.

JOHNSON, A.; HEBERT, M. Using spin images for efficient object recognition in cluttered 3d scenes. **Transactions on Pattern Analysis and Machine Intelligence**, v. 21, n. 5, p. 433–449, 1999. Citado na página 34.

KAZHDAN, M.; FUNKHOUSER, T.; RUSINKIEWICZ, S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In: **Proceedings of 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing**. [S.l.: s.n.], 2003. p. 156–164. Citado nas páginas 35 e 36.

KAZMI, I. K.; YOU, L.; ZHANG, J. J. A survey of 2d and 3d shape descriptors. In: **Proceedings of 10th International Conference Computer Graphics, Imaging and Visualization (CGIV)**. [S.l.: s.n.], 2013. Citado na página 38.

KIM, D. I.; SUKHATME, G. S. Semantic labeling of 3d point clouds with object affordance for robot manipulation. In: **Proceedings of 2014 IEEE International Conference on Robotics & Automation (ICRA)**. [S.l.: s.n.], 2014. Citado na página 37.

KLASING, K.; WOLLHERR, D.; BUSS, M. A clustering method for efficient segmentation of 3d laser data. In: **IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2008. p. 4043–4048. Citado na página 39.

KNOPP, J.; PRASAD, M.; WILLEMS, G.; TIMOFTE, R.; GOOL, L. V. Hough transform and 3d surf for robust three dimensional classification. In: **Proceedings of European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2010. Citado na página 33.

KRIG, S. **Computer Vision Metrics - Survey, Taxonomy and Analysis of Computer Vision, Visual Neuroscience, and Deep Learning**. [S.l.]: Springer, 2016. Citado nas páginas 21, 33 e 38.

LAGA, H.; TAKAHASHI, H.; NAKAJIMA, M. Spherical wavelet descriptors for content-based 3d model retrieval. In: **Proceedings of IEEE International Conference on Shape Modeling and Applications**. [S.l.: s.n.], 2006. p. 15–25. Citado na página 35.

LOWE, D. G. Object recognition from local scale-invariant features. In: **Proceedings of the International Conference on Computer Vision**. [S.l.: s.n.], 1999. p. 1150–1157. Citado na página 33.

_____. Distinctive image features from scale-invariant keypoints. **International Journal of Computer Vision**, 2004. Citado na página 33.

LUETTEL, T.; HIMMELSBACH, M.; WUENSCH, H.-J. Autonomous ground vehicles - concepts and a path to the future. In: **Proceedings of the IEEE**. [S.l.: s.n.], 2012. v. 100, p. 1831–1839. Citado nas páginas 23, 28 e 29.

MADRY, M.; AFKHAM, H. M.; EK, C. H.; CARLSSON, S.; KRAGIC, D. Extracting essential local object characteristics for 3d object categorization. In: **Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2013. Citado na página 37.

MIAN, A. S.; BENNAMOUN, M.; OWENS, R. A. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. **International Journal of Computer Vision**, v. 89, n. 2-3, p. 348–361, 2010. Citado na página 33.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw-Hill, 1997. Citado nas páginas 38 e 39.

MOZOS, O. M.; ROTTMANN, A.; TRIEBEL, R.; JENSFELT, P.; BURGARD, W. Semantic labeling of places using information extracted from laser and vision sensor data. In: **IEEE/RSJ IROS Workshop: From Sensors to Human Spatial Concepts**. [S.l.: s.n.], 2006. Citado nas páginas 28 e 29.

MOZOS, O. M.; STACHNISS, C.; BURGARD, W. Supervised learning of places from range data using adaboost. In: **Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2005. Citado na página 28.

MUELLER, A.; HIMMELSBACH, M.; LUETTEL, T.; HUNDELSHAUSEN, F.; WUENSCH, H. J. Gis-based topological robot localization through lidar crossroad detection. In: **14th International IEEE Conference on Intelligent Transportation Systems (ITSC)**. [S.l.: s.n.], 2011. p. 2001–2008. Citado na página 28.

MURPHY, K. **Machine Learning – A probabilistic Perspective**. [S.l.]: MIT Press, 2012. Citado na página 39.

OHBUCHI, R.; OSADA, K.; FURUYA, T.; BANNO, T. Salient local visual features for shape-based 3d model retrieval. In: **Proceedings of IEEE Shape Modeling International (SMI)**. [S.l.: s.n.], 2008. Citado na página 36.

OSADA, R.; FUNKHOUSER, T.; CHAZELLE, B.; DOBKIN, D. Shape distributions. **ACM Trans. Graphics**, v. 21, n. 4, p. 807–832, 2002. Citado nas páginas 35 e 36.

PEARS, N.; LIU, Y.; BUNTING, P. **3D Imaging, Analysis and Applications**. [S.l.]: Springer, 2012. Citado nas páginas 33 e 38.

QIAN, X.; YE, C. 3d object recognition by geometric context and gaussian-mixture-model-based plane classification. In: **Proceedings of 2014 IEEE International Conference on Robotics & Automation (ICRA)**. [S.l.: s.n.], 2014. Citado na página 37.

REZENDE, S. **Sistemas Inteligentes: Fundamentos e Aplicações**. [S.l.]: Editora Manole, 2003. ISBN 85-204-1683-7. Citado na página 57.

RICARD, J.; COEURJOLLY, D.; BASKURT, A. Generalizations of angular radial transform for 2d and 3d shape retrieval. **Pattern Recognition Letters**, v. 26, n. 14, p. 2174–2186, 2005. Citado na página 35.

RIDEL, D. A.; SHINZATO, P. Y.; WOLF, D. F. Obstacle segmentation with low-density disparity maps. In: **7th workshop on Planning, Perception and Navigation for Intelligent Vehicles - International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2015. Citado na página 69.

RIEDMILLER, M.; BRAUN, H. techreport, **Rprop – description and implementation details**. 1994. Citado na página 57.

ROMERO, R. A.; PRESTES, E.; OSÓRIO, F. S.; WOLF, D. F. **Robótica Móvel**. [S.l.]: LTC, 2014. ISBN 8521623038. Citado nas páginas 21, 27, 28, 29 e 30.

ROSEIRO, L.; OSÓRIO, F. Detecção de pedestres e obstáculos usando visão computacional e radar em veículos automotivos. In: **CTDR/WTDR - Workshop on MSc Dissertation and PhD Thesis in Robotics**. [S.l.: s.n.], 2016. Citado na página 69.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. In: _____. **Parallel distributed processing: explorations in the microstructure of cognition**. [S.l.: s.n.], 1986. p. 318–362. Citado na página 40.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: A modern approach**. [S.l.]: Pearson Education, 2003. Citado na página 40.

RUSU, R.; BLODOW, N.; BEETZ, M. Fast point feature histograms (fpfh) for 3d registration. In: **Proceedings of the International Conference on Robotics and Automation (ICRA)**. [S.l.: s.n.], 2009. Citado na página 34.

RUSU, R. B.; BRADSKI, G.; THIBAU, R.; HSU, J. Fast 3d recognition and pose using the viewpoint feature histogram. In: **Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2010. Citado na página 35.

SALES, D. O. **NeuroFSM: aprendizado de Autômatos Finitos através do uso de Redes Neurais Artificiais aplicadas à robôs móveis e veículos autônomos**. Dissertação (Mestrado), 2012. Citado nas páginas 24 e 29.

SALES, D. O.; AMARO, J.; OSÓRIO, F. S. 3d feature extraction for objects recognition. In: **IEEE Latin American Robotics Symposium (LARS)**. [S.l.: s.n.], 2017. Citado na página 60.

SALES, D. O.; CORREA, D. O.; FERNANDES, L. C.; WOLF, D. F.; OSÓRIO, F. S. Adaptive finite state machine based visual autonomous navigation system. **Engineering Applications of Artificial Intelligence**, v. 29, p. 152–162, 2014. ISSN 0952-1976. Citado nas páginas 30 e 83.

SALES, D. O.; CORREA, D. O.; WOLF, D. F.; OSORIO, F. S. 3d vision-based autonomous navigation system using ann and kinect sensor. In: **13th Engineering Applications of Neural Network Conference (EANN2012)**. [S.l.: s.n.], 2012. Citado na página 29.

SALES, D. O.; FERNANDES, L. C.; WOLF, D. F.; OSORIO, F. S. Fsm-based visual navigation for autonomous vehicles. In: **Workshop on Visual Control of Mobile Robots - IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2012. Citado nas páginas 29, 81 e 82.

SALTI, S.; TOMBARI, F.; STEFANO, L. D. Shot: Unique signatures of histograms for surface and texture description. **Computer Vision and Image Understanding**, 2014. Citado nas páginas 34 e 35.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais para Engenharia e Ciências Aplicadas**. [S.l.]: Artliber, 2010. Citado nas páginas 40 e 41.

SUNDAR, H.; SILVER, D.; GAGVANI, N.; DICKINSON, S. Skeleton based shape matching and retrieval. In: **Proc. Shape Modeling Int'l**. [S.l.: s.n.], 2003. Citado na página 37.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. [S.l.: s.n.], 2005. Citado na página 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.; PALATUCCI, M.; PRATT, V.; STANG, P.; STROHBAND, S.; DUPONT, C.; JENDROSSEK, L.-E.; KOELEN, C.; MARKEY, C.; RUMMEL, C.; NIEKERK, J. V.; JENSEN, E.; ALESSANDRINI, P.; BRADSKI, G.; DAVIES, B.; ETTINGER, S.; KAEHLER, A.; NEFIAN, A.; MAHONEY, P. Winning the darpa grand challenge. **Journal of Field Robotics**, 2006. Citado na página 23.

TOMBARI, F.; FIORAIO, N.; CAVALLARI, T.; SALTI, S.; PETRELLI, A.; STEFANO, L. D. Automatic detection of pole-like structures in 3d urban environments. In: **Proceedings of 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2014. Citado na página 37.

TOMBARI, F.; SALTI, S.; STEFANO, L. D. Unique signatures of histograms for local surface description. In: **Proceedings of European Conference on Computer Vision (ECCV)**. [S.l.: s.n.], 2010. Citado nas páginas 33 e 35.

_____. Performance evaluation of 3d keypoint detectors. **International Journal of Computer Vision**, 2013. Citado na página 22.

TUNG, T.; SCHMITT, F. The augmented multiresolution reeb graph approach for content-based retrieval of 3d shapes. **International Journal on Shape Modeling**, v. 11, n. 1, 2005. Citado na página 37.

TUYTELAARS, T.; MIKOLAJCZYK, K. Local invariant feature detectors: A survey. **Foundations and Trends in Computer Graphics and Vision**, v. 3, n. 3, p. 177–280, 2008. Citado nas páginas 22 e 32.

URMSON, C. Autonomous driving in urban environments: Boss and the urban challenge. **Journal of Field Robotics**, p. 425–466, 2008. Citado nas páginas 23 e 32.

VRANIC, D. V. **3D Model Retrieval**. phdthesis, 2004. Citado na página 36.

- WAN, J.; GUO, G.; LI, S. Z. Explore efficient local features from rgb-d data for one-shot learning gesture recognition. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2016. Citado na página 37.
- WOLF, D.; OSÓRIO, F. S.; SIMÕES, E.; TRINDADE, O. Robótica inteligente: da simulacao às aplicacoes no mundo real. In: **JAI - Jornada de Atualizacao em Informatica da SBC**. [S.l.: s.n.], 2009. Citado nas páginas 23 e 27.
- YU, Y.; LI, J.; GUAN, H.; JIA, F.; WANG, C. Learning hierarchical features for automated extraction of road markings from 3-d mobile lidar point clouds. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 8, n. 2, 2015. Citado na página 37.
- ZAHARESCU, A.; BOYER, E.; VARANASI, K.; HORAUD, R. Surface feature detection and description with applications to mesh matching. In: **Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2009. Citado na página 35.
- ZHAO, G.; YUAN, J.; DANG, K. Height gradient histogram (high) for 3d scene labeling. In: **Proceedings of 2014 Second International Conference on 3D Vision (3DV)**. [S.l.: s.n.], 2014. Citado na página 34.
- ZHU, X.; ZHAO, H.; LIU, Y.; ZHAO, Y.; ZHA, H. Segmentation and classification of range image from an intelligent vehicle in urban environment. In: **Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**. [S.l.: s.n.], 2010. Citado nas páginas 37 e 63.

GLOSSÁRIO

Cena 3D: É a nuvem de pontos completa fornecida pelo sensor em uma coleta simples, de forma análoga a uma fotografia.

Descritor: Estrutura utilizada para o armazenamento e representação de uma *feature* de forma simplificada e invariante a um conjunto de transformações.

Feature: É uma característica singular da região adjacente a um ponto de interesse ou de um conjunto de pontos.

Pontos de referência: Estruturas especiais presentes no ambiente cuja detecção permite a estimação aproximada da localização do robô em um mapa local ou global. São classificados como “naturais” quando são estruturas pertencentes ao próprio ambiente (como árvores e montanhas), e artificiais quando são produzidos pelo homem (prédios, torres, monumentos).

EXPERIMENTOS COMPLEMENTARES

Esta seção apresenta experimentos complementares realizados durante a fase inicial deste projeto de doutorado, ilustrando a transição entre as abordagens baseadas em *features* 2D e 3D para implementação de sistemas de percepção em aplicações de navegação topológica.

A.1 Reconhecimento de padrões do ambiente usando detecção de bordas

Este experimento consiste em uma das primeiras tentativas de encontrar características adequadas para o reconhecimento de padrões em aplicações de localização em mapas topológicos, ainda utilizando um sistema de percepção 2D. A idéia principal deste experimento é a substituição da unidade de pré-processamento dos dados de entrada utilizada em (SALES *et al.*, 2012b), que utilizava uma câmera monocular como sensor principal para aquisição de dados do ambiente (vias urbanas reais). No experimento anterior era utilizado um classificador neural para pré-processamento das imagens, criando assim mapas de navegabilidade cuja função é indicar quais regiões na cena capturada representam áreas navegáveis e não-navegáveis. O conjunto de pixels rotulados era então utilizado como entrada para uma RNA, utilizada para induzir o contexto corrente do veículo em um mapa topológico com base nessa informação.

Sendo assim, foi proposto um novo sistema de pré-processamento das imagens, baseado na detecção de descontinuidades e formato da pista através da aplicação do algoritmo de detecção de bordas *Canny* (CANNY, 1986). Desta forma, este algoritmo foi aplicado aos frames originais, produzindo saídas binárias. Este novo frame binário é então dividido em blocos de 10×10 pixels para diminuição da resolução, e por fim é aplicado um threshold binário. A saída final é uma matriz binária de dimensões 24×32 . A Figura 35 ilustra alguns exemplos de dados processados. Uma vez aplicada a detecção de bordas para segmentação da área da pista, os dados são utilizados como entrada de uma RNA, para reconhecimento do contexto atual do robô (reta, curvas ou

cruzamentos), fornecendo assim informação semântica do ambiente.

Figura 35 – Imagens processadas e matrizes binárias correspondentes.

Fonte: Elaborada pelo autor.

A base de treinamento da RNA foi gerada com o processamento dos mesmos 2040 frames por classe (aproximadamente) utilizados no experimento anterior (SALES *et al.*, 2012b), resultando em uma base de dados com 8165 exemplos (pares entrada/saída). Foi selecionada uma área de interesse nas matrizes descartando a região acima da linha do horizonte, utilizando portanto 384 células da matriz binária como entradas da RNA. A quantidade de neurônios na camada escondida da RNA foi definida de forma empírica, onde chegou-se a um número de 24 neurônios (contra 384 do experimento anterior). A validação foi feita usando o método *k-fold* de validação cruzada, com $k = 5$.

A tabela 8 apresenta a comparação entre a acurácia obtida com o pré-processamento do experimento anterior (SALES *et al.*, 2012b) e com o uso do método proposto. A tabela 9 apresenta as matrizes de confusão resultantes desse experimento.

Tabela 8 – Comparação entre a acurácia obtida nas duas versões do sistema.

Método	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5
Mapas de navegabilidade	98,8%	99,0%	99,1%	99,3%	99,5%
Deteção de bordas	98,3%	98,4%	99,1%	98,0%	98,4%

Analisando a tabela 8 é possível observar que os resultados obtidos são muito próximos dos resultados apresentados em (SALES *et al.*, 2012b), com a vantagem de ser utilizada uma quantidade bem menor de neurônios na camada escondida da RNA deste novo método, o que diminuiu tanto o tempo de treinamento como o tempo de execução quando integrado ao sistema de controle do veículo autônomo. Foi possível concluir ainda que a detecção de bordas é uma abordagem tão representativa quanto a detecção de áreas navegáveis para reconhecimento de características estruturais do ambiente (formato da pista).

Tabela 9 – Matrizes de confusão dos testes do novo modelo.

Teste 1				Teste 2				Teste 3			
416	0	0	3	415	2	0	2	416	0	1	2
0	379	0	2	1	379	0	1	0	377	0	4
0	1	370	10	0	2	369	10	1	0	376	4
2	1	6	444	0	0	1	451	1	1	4	446

Teste 4				Teste 5			
416	0	0	3	413	0	2	3
0	377	0	3	1	377	1	1
0	0	378	2	1	0	373	6
1	0	2	449	2	0	2	447

Fonte: Elaborada pelo autor.

Este experimento foi descrito no artigo “*Adaptive finite state machine based visual autonomous navigation system*”, publicado no periódico *Engineering Applications of Artificial Intelligence* (EAAI) (SALES *et al.*, 2014). Os resultados obtidos motivaram a migração de implementações baseadas em *features* 2D mais simples em direção a abordagens baseadas em modelos de *features* mais complexos, até a consolidação da metodologia apresentada nesta tese para aplicações baseadas em sistemas de percepção 3D.

LISTA DE PUBLICAÇÕES OBTIDAS

- SALES, D. O.; OSÓRIO, F. S. “*3D shape retrieval using 3D-CSD feature*”. In: International Journal of Computer Vision (IJCV) - submetido
- SALES, D. O.; AMARO, J.; OSÓRIO, F. S. “*3D feature extraction for objects recognition*”. In: IEEE Latin American Robotics Symposium (LARS), 2017.
- SALES, D. O.; CORREA, D. O.; FERNANDES, L. C.; WOLF, D. F.; OSORIO, F. S. “*Adaptive finite state machine based visual autonomous navigation system.*” Engineering Applications of Artificial Intelligence, v. 29, p. 152–162, 2014. ISSN 0952-1976.
- FERNANDES, L. C; SOUZA, J. R; PESSIN, G; SHINZATO, P. Y; SALES, D. O; MENDES, C; PRADO, M; KLASER, R; MAGALHÃES, A. C; HATA, A; PIGATTO, D; CASTELO BRANCO, K; GRASSI JR, V; OSÓRIO, F. S; WOLF, D. F. “*CaRINA Intelligent Robotic Car: Architectural design and applications*”. In: Journal of Systems Architecture. v. 1, p. 372-392, 2014.
- DE CARVALHO SANTOS, VALÉRIA ; SALES, Daniel Oliva ; TOLEDO, CLÁUDIO FABIANO MOTTA ; OSÓRIO, FERNANDO SANTOS . “*A hybrid GA-ANN approach for autonomous robots topological navigation*”. In: the 29th Annual ACM Symposium, 2014, Gyeongju. Proceedings of the 29th Annual ACM Symposium on Applied Computing - SAC '14. v. 1. p. 148-153.