Pós-Graduação em Ciência da Computação

# MAPAS AUTO-ORGANIZÁVEIS DE TOPOLOGIA VARIANTE NO TEMPO PARA SLAM VISUAL EM ESPAÇO DE APARÊNCIAS

Por

Daniel de Filgueiras Gomes

Tese de Doutorado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE

2015

Universidade Federal de Pernambuco
Centro de Informática
Pós-graduação em Ciência da Computação

Daniel de Filgueiras Gomes

# MAPAS AUTO-ORGANIZÁVEIS DE TOPOLOGIA VARIANTE NO TEMPO PARA SLAM VISUAL EM ESPAÇO DE APARÊNCIAS

*Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Doutor em Inteligencia Computacional.*

Orientador: *Aluizio Fausto Ribeiro Araújo*

RECIFE
2015

Catalogação na fonte
Bibliotecária Jane Souto Maior, CRB4-571

G633m Gomes, Daniel de Filgueiras
Mapas auto-organizáveis de topologia variante no tempo para SLAM visual em espaço de aparências / Daniel de Filgueiras Gomes. – Recife: O Autor, 2015.
64 f.: il., fig., tab.

Orientador: Aluizio Fausto Ribeiro Araújo.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, 2015.
Inclui referências e apêndice.

1. Inteligência computacional. 2. Processamento de imagem. I. Araújo, Aluizio Fausto Ribeiro (orientador). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2015-172

Tese de Doutorado apresentada por **Daniel das Filgueiras Gomes** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Mapas Auto-organizáveis de Topologia Variante no Tempo para SLAM Visual em Espaço de Aparências"**, orientada pelo **Prof. Aluízio Fausto Ribeiro Araújo** e aprovada pela Banca Examinadora formada pelos professores:

_____
Prof. Abel Guilhermino da Silva Filho
Centro de Informática / UFPE

_____
Prof. Paulo Salgado Gomes de Mattos Neto
Centro de Informática / UFPE

_____
Profa. Silvia Silva da Costa Botelho
Centro de Ciências Computacionais – C3 / UFRG

_____
Prof. Pedro Manuel González Del Foyo
Departamento de Engenharia Mecânica - UFPE

_____
Profa. Raquel Frizzera Vassalo
Departamento de Engenharia Elétrica - UFES

Visto e permitida a impressão.
Recife, 12 de março de 2015.

_____
**Profa. Edna Natividade da Silva Barros**
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

*Dedico este trabalho aos meus pais, irmã e esposa.*

*Today's scientists have substituted mathematics for experiments, and they wander off through equation after equation, and eventually build a structure which has no relation to reality.*

—NIKOLA TESLA ("Radio Power Will Revolutionize the World". Modern Mechanics and Inventions, July, 1934.)

# Resumo

Implementar agentes robóticos autônomos capazes de navegar entre duas posições ou explorar ambientes desconhecidos são problemas antigos e importantes em áreas como robótica e inteligência computacional. Um pré-requisito para subsidiar as atividades de navegação e exploração autônomas é dotar o agente robótico da capacidade de construir mapas do ambiente ao seu redor e de conseguir identificar a sua posição no mapa construído. Este é o problema de mapeamento e localização simultâneos de um ambiente desconhecido (Simultaneous localization and mapping ou SLAM) e pode ser modelado matematicamente como um problema de filtragem. Nos últimos 10 anos foram obtidos grandes avanços nesta área com o desenvolvimento de várias abordagens para a resolução deste problema. As abordagens mais populares utilizam filtros estatísticos, semelhantes ao de Kalman, e visam a construção de mapas com escala métrica, semelhantes aos utilizados em cartografia. Em relação ao aparato sensorial, os sensores mais comuns são do tipo *range finder*, ou seja, sensores que permitem, de modo direto ou indireto, captar as relações de distância métrica de um determinado ambiente. Exemplos de sensores desta natureza são: Lidares, Sonares, Radares. Este tipo de abordagem resulta, no geral, em um aparato sensorial caro e pesado, que necessita de um alto poder de processamento de dados, alto consumo de energia e de uma modelagem matemática complexa e específica para cada tipo de agente móvel. É uma abordagem, do ponto de vista sensorial, não natural, pois a maioria dos seres vivos não necessita de conhecimento métrico preciso do ambiente que está ao seu redor para se localizar no ambiente e explorar ambientes desconhecidos. Assim, esta tese propõe uma abordagem mais natural ao problema de SLAM, utilizando recursos disponíveis a maioria dos animais, como visão e memorização de relações de vizinhança. Este trabalho realiza uma revisão bibliográfica dos artigos e projetos científicos dos últimos 20 anos relacionados com a área de navegação inteligente e robótica e propõe uma nova abordagem para o problema de SLAM Visual em Espaço de Aparências utilizando apenas visão monocular como estímulo de entrada e mapas auto-organizáveis dinâmicos especializados.

**Palavras-chave:** Inteligência Computacional, Processamento de Imagens, Navegação Robótica, Navegação Autônoma, Mapeamento topológico, Localização visual, SLAM topológico.

# Abstract

Implement autonomous robotic agents able to navigate between two positions or explore unknown environments are old and important problems in areas such as robotics and artificial intelligence. A prerequisite to support the navigation and autonomous exploration activities is to provide the robotic agent ability to build maps of the around environment and be able to identify its position in the map. This is the simultaneous localization and mapping problem (SLAM) and can be mathematically modeled as a filtering problem. In the last 10 years were obtained great advances in this area with the development of various approaches to solving this problem. The most popular approaches use statistical filters, similar to the Kalman, and aim to build maps with metric scale, similar to those used in cartography. Regarding the sensory apparatus, the most common sensors type are range finders, it are, directly or indirectly, capture metric distance and the relationships of a given environment. Examples of sensors of this type are: Lidar, Sonar, Radar. This approach results, in general, in an expensive and heavy sensory apparatus, a high-powered data processing, high energy consumption and a complex and specific mathematical modeling for each type of mobile agent. It also does not appear to be a natural approach, since most living beings does not require metric knowledge need the ambiance this around to locate the environment and explore unknown environments. Thus, this thesis aims to provide a more natural approach to the SLAM problem using available resources most animals, such as vision and neighborly relations memorization. This thesis conducts a literature review of articles and scientific projects of the last 20 years related to the field of robotics and intelligent navigation and proposes a new approach for Visual SLAM problem using only monocular vision as input stimuli and self-organizing maps dynamic specialize in memorizing states trajectories.

**Keywords:** Computational Intelligence, Image Processing, Robotic Navigation, Autonomous Navigation, Topological Mapping, Visual localization, Topological SLAM.

# Lista de Figuras


2.1 Exemplo de uma rede SOM . . . . . 27

3.1 Módulos do sistema de navegação proposto por Mole. (Imagem retirada da tese de doutorado de MOLE (2010) ) . . . . . 37
3.2 SLAM . . . . . 41

4.1 Ilustração dos parâmetros $d_0$, $d_1$ e $d_2$. . . . . 50
4.2 Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{15}$, $d_1 = \frac{4}{3}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$ . . . . . 51
4.3 Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{30}$, $d_1 = \frac{2}{3}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$ . . . . . 52
4.4 Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{60}$, $d_1 = \frac{2}{6}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$ . . . . . 53
4.5 Ilustração da aquisição de pontos com espaçamento cartesiano não uniforme de um sensor de varredura tipico. . . . . 54
4.6 Localização Visual e Mapeamento Topológico . . . . . 57
4.7 Construção do mapa topológico . . . . . 57
4.8 Imagem da Avenida dos Reitores na UFPE. Cada circulo é um *keypoint* SURF-128, vetor de dimensão 128 e magnitude entre 0 e 1. Estes vetores são invariantes a translação, rotação e escala e robustos a mudanças de perspectiva. . . . . 58
4.9 Ilustração do processo de agrupamento dos descritores SURF . . . . . 59
4.10 Compartilhamento da informação . . . . . 61

5.1 Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede SOM ao longo do tempo. . . . . 65
5.2 Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede RECSOM ao longo do tempo. . . . . 66
5.3 Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede GNG ao longo do tempo. . . . . 66
5.4 Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede FSS-SOM ao longo do tempo. . . . . 67
5.5 *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal ingenuo para a série de Mackey-Glass. . . . . 69

5.7 *MSE* variando de $1 \leq k \leq 30$ do SOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass. . . . . 70
5.6 *MSE* variando de $1 \leq k \leq 30$ de um previsor do tipo *Random* para a série de Mackey-Glass. . . . . 70
5.8 *MSE* variando de $1 \leq k \leq 30$ do RecSOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass. . . . . 71
5.9 *MSE* variando de $1 \leq k \leq 30$ do GNG com um codebook de modo supervisionado como previsor para a série de Mackey-Glass. . . . . 71
5.10 *MSE* variando de $1 \leq k \leq 30$ do FSS-SOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass. . . . . 72
5.11 *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal ingenuo para a série SOI. 72
5.12 *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal tipo Random para a série SOI. . . . . 73
5.13 *MSE* variando de $1 \leq k \leq 30$ do SOM com um codebook de modo supervisionado como previsor para a série SOI. . . . . 73
5.14 *MSE* variando de $1 \leq k \leq 30$ do RecSOM com um codebook de modo supervisionado como previsor para a série SOI. . . . . 74
5.15 *MSE* variando de $1 \leq k \leq 30$ do GNG com um codebook de modo supervisionado como previsor para a série SOI. . . . . 74
5.16 *MSE* variando de $1 \leq k \leq 30$ do FSS-SOM com um codebook de modo supervisionado como previsor para a série SOI. . . . . 75
5.17 *MSE* por $Epoca$ do SOM, os resultados utilizando o RecSOM são praticamente identicos ao do SOM e não estão incluídos no gráfico. . . . . 76
5.18 *MSE* por $Epoca$ do GNG. . . . . 77
5.19 *MSE* por $Epoca$ usando o FSS-SOM. Realizando uma comparação entre as Figuras 5.19,do FSS-SOM, e 5.18, do GNG, é interessante notar que o GNG com o mesmo número de nodos e conjunto de treinamento só consegue um MSE abaixo de 1550 após 12 épocas, enquanto o FSS-SOM atinge este valor em aproximadamente 5 épocas, evidenciando a característica de convergência rápida do mapa. . . . . 77
5.21 Frames . . . . . 80
5.20 Semelhança versus frames . . . . . 80
5.22 Frames . . . . . 81
5.23 Semelhança versus frames . . . . . 81
5.24 Vista superior em ambiente externo no campus da UFPE . . . . . 82
5.25 Vista aérea da base *New College* . . . . . 84
5.26 Curva Precision-Recall utilizando o modelo proposto com a base de dados *New College*. . . . . 85
5.27 Vista aérea da base *City* . . . . . 85

5.28 Curva Precision-Recall utilizando o modelo proposto com a base de dados City. 86
5.29 Curva Precision-Recall utilizando o modelo proposto com a base de dados Kitti. 88
5.30 Resultados obtidos por Cummins . . . . . 89
5.31 Percurso mapeado em ambiente externo no campus da UFPE . . . . . 90
5.32 Imagens associada aos nodo 601 (a) e 8923 (b) do mapa topológico da Figura 5.31 91
5.33 Resultados do sistema de mapeamento topológico . . . . . 92
5.34 Resultados do sistema de mapeamento topológico . . . . . 93
5.35 Resultados do sistema de mapeamento topológico . . . . . 94

A.1 Diagrama do sistema robótico . . . . . 108
A.2 Mapeamento da memória do sistema robótico . . . . . 108
A.3 Codificação do acionamento dos motores . . . . . 109
A.4 Pacote tipo 1 . . . . . 110
A.5 Pacote tipo 2 . . . . . 111
A.6 Pacote tipo 3 . . . . . 111
A.7 Vista superior . . . . . 112
A.8 Vista frontal . . . . . 112
A.9 Vista lateral . . . . . 113

# Lista de Tabelas


5.1 Taxa de erros na localização dentro do Cin . . . . . 82
5.2 Taxa de erro na localização na Avenida dos Reitores, UFPE . . . . . 83
5.3 Configuração dos mapas utilizados no banco de testes *New College* . . . . . 84
5.4 Configuração dos mapas utilizados no banco de testes *City* . . . . . 86
5.5 Configuração dos mapas utilizados no banco de testes Kitti20110926 . . . . . 87
5.6 Configuração dos mapas utilizados no banco de testes Kitti20110929 . . . . . 87

# Lista de Acrônimos e Siglas

| | | |
|---|---|---|
| BoF | – | *bag of features* |
| BoK | – | *bag of keypoints* |
| BoW | – | *bag of words* |
| BoVW | – | *bag of visual words* |
| DIRD | – | *Dird is an Illumination Robust Descriptor* |
| EKF | – | *Extended Kalman Filter* |
| FFT | – | *Fast Fourier Transform* |
| FSS-SOM | – | *Fast Coverage of an N-dimensional State Space Self-Organizing Map* |
| FWT | – | *Fast Wavelet Transform* |
| GNG | – | *Growing Neural Gas* |
| GPS | – | *Global Positioning System* |
| GWR | – | *Grow When Required* |
| ICP | – | *Iterative Closest Point* |
| LARFSOM | – | *Local Adaptive Receptive Field Self-organizing Map* |
| LHC | – | *Latin Hyper Cube* |
| LIDAR | – | *LIght Detection And Ranging* |
| MLP | – | *Multilayer Perceptron* |
| MSE | – | *Mean Squared Error* |
| RADAR | – | *Radio Detection And Ranging* |
| RANSAC | – | *Random Sample Consensus* |
| RBF | – | *Radial Basis Function* |
| SIFT | – | *Scale Invariant Feature Transform* |
| SLAM | – | *Simultaneous localization and mapping* |
| SOI | – | *Southern Oscillation Index* |
| SOM | – | *Self-organizing Map* |
| SONAR | – | *Sound Navigation And Ranging* |
| SURF | – | *Speeded Up Robust Features* |

# Notação Utilizada

| Notações | |
|---|---|
| x,y,i | Variáveis escalares |
| t,k | Variáveis preferenciais para representar tempo |
| $\mathbf{v}$ | Vetor posição |
| $\mathbf{w}_i$ | Vetor de peso de um nodo $i$ |
| $\xi$ | Vetor de entrada ou de teste em um Mapa Auto-Organizável |
| $J$ | Utilizado preferencialmente para representar conjuntos |
| $f(.)$ | Uma função arbitrária |
| $\hat{f}(.)$ | Previsão para uma função arbitrária $f(.)$ |
| $\|.\|$ | Operador de cardinalidade de um conjunto |
| $\|\|.\|\|$ | Norma euclidiana de um vetor |
| $d(\mathbf{w}_1,\mathbf{w}_2)$ | Distância euclidiana entre os vetores $\mathbf{w}_1$ e $\mathbf{w}_1$ |
| $\sigma$ | Desvio padrão |
| $N(m,\sigma)$ | Distribuição normal com média $m$ e desvio $\sigma$ |
| $i = \mathbf{I}(x,y)$ | Imagens monocromáticas bidimensionais |
| $\mathbf{I}$ | Usado preferencialmente para representar imagens em geral (uma matriz $m \times n$) |
| $i_r$,$i_g$,$i_b$ | Intensidades Red, Green e Blue do sinal de cor de uma determinada imagem |

# Definições e Termos Utilizados

A área de estudos de navegação autônoma possui muitos termos que apresentam significados distintos dependendo da referência, do contexto ou da subárea de pesquisa. Com o intuito de produzir um texto mais claro e de melhor entendimento apresentamos a seguir uma lista com as principais definições operacionais dos termos mais utilizados neste trabalho.

**Agente** É uma máquina capaz de perceber informações sobre seu ambiente e usar o seu conhecimento sobre esse ambiente para mover-se de forma segura de modo significante e intencional (MOLE, 2010).

**Ambiente** Local ou região física ou virtual composta por várias partes menores onde está inserido um agente.

**Sistema Sensor** Um conjunto de sensores que produz um fluxo de vetores ao longo do tempo contendo informações do ambiente onde o agente está.

**Navegação autônoma** É a capacidade de um agente de se deslocar em um ambiente de modo proativo sem causar danos ao ambiente e a si mesmo. A navegação pode ser subsidiada por informações locais fornecidas pelo sistema sensorial do agente e de informações previamente fornecidas como mapas do ambiente no qual o agente está inserido (MOLE, 2010).

**Exploração autônoma** Exploração é o problema de controlar um agente com o objetivo de maximizar o conhecimento do agente sobre o mundo exterior (THRUN. SEBASTIAN BURGARD. WOLFRAM, 2006). Ou seja, considerando-se um agente autônomo inserido em um ambiente inicialmente desconhecido. O agente deve se movimentar no ambiente de tal forma a ampliar o seu conhecimento sobre o mesmo. A exploração autônoma está fortemente relacionada com o problema da localização e mapeamento simultâneo.

**Localização** É a habilidade de um agente autônomo identificar a sua posição atual em um dado mapa baseado em informações sensoriais presentes e passadas. Atualmente a maioria dos sistemas de navegação e exploração autônomos se utilizam principalmente de sistemas de posicionamento global para realizar a tarefa de localização. A localização também pode ser obtida através de reconhecimento de landmarks ou pontos de referência de um determinado cenário (CHOSET et al., 2005).

**Mapeamento** É a capacidade de um determinado agente autônomo de produzir mapas de uma determinada região ou localidade. Muitos sistemas de navegação autônoma não apresentam ou apresentam de modo limitado a capacidade de produzir os seus próprios mapas. A abordagem mais comum é utilizar georreferenciamento com imagens de satélite e sinal de GPS como subsídios básicos para a construção dos mapas (CHOSET et al., 2005).

**Planejamento de rotas** Dado um mapa, um objetivo e um critério de custo, planejamento de rotas é a capacidade que um agente autônomo possui de produzir sequencia de movimentos/ações que atendam ao objetivo estabelecido a um custo mínimo. Normalmente o planejamento de rotas visa encontra um caminho mais curto e seguro entre dois pontos em um mapa que o agente possa usar para se deslocar (CHOSET et al., 2005).

**Mapeamento local** É o mapeamento de um cenário ou ambiente que pode ser realizado pelo próprio sistema sensorial do agente sem que haja a necessidade de deslocamento do agente pelo ambiente.

**Mapeamento global** É o mapeamento de um cenário que não pode ser realizado pelo próprio sistema sensorial do agente sem que o mesmo se desloque para uma ou mais posições do ambiente.

**Localização e Mapeamento simultâneos (SLAM)** É o nome de um dos problemas centrais na área de navegação robótica. O problema de SLAM consiste em, utilizando um agente com um aparato sensorial capaz de realizar mapeamento local, realizar simultaneamente a construção de um mapa do ambiente desconhecido no qual agente está inserido e identificar sua localização no mesmo (CHOSET et al., 2005).

**SLAM Visual em Espaço de Aparências** Termo criado por CUMMINS (2009) para definir um novo tipo de sistema de SLAM onde o objetivo principal é estimar o local onde um agente se encontra e não uma coordenada do local.

**Bag of words** Em visão computacional um *bag of words* (BoW) é um vetor com as ocorrências de um conjunto de características que uma determinada imagem apresenta e que pode ser utilizado em um processo de classificação de imagens. Em visão computacional os termos *bag of visual words* (BoVW) ou *bag of features* (BoF) ou *bag of keypoints* (BoK) também são utilizados com o mesmo sentido (RAMANAN; NIRANJAN, 2012).

**Codebook** É um tipo de quantização vetorial, funciona como uma tabela que associa $n$ valores discretos (rótulos) com saídas $\mathbf{y}$ (MARTINETZ T., 1993).

**Localização visual** Nesta tese, localização visual é o procedimento de determinar a posição de um agente em um mapa utilizando apenas informações visuais locais. O agente deve possuir informações a priori do mapa (topológico ou geométrico) e de características visuais associadas a posições no mapa (WANG; CIPOLLA; ZHA, 2005; SHEN; HU, 2006; TORII; SIVIC; PAJDLA, 2011).

# Sumário


**1 Introdução** **18**

1.1 Enunciado do Problema . . . . . 22
1.2 Objetivos . . . . . 22
1.2.1 Objetivo primário . . . . . 23
1.2.2 Objetivos secundário . . . . . 23
1.3 Estrutura do Documento . . . . . 23
1.4 Publicação . . . . . 24

**2 Redes Neurais** **25**

2.1 Introdução . . . . . 25
2.1.1 Abordagem Não Supervisionada e Mapas Auto-Organizáveis . . . . . 25
2.1.1.1 Mapas de Kohonen . . . . . 26
2.1.1.2 Growing Neural Gas . . . . . 28
2.1.1.3 LARFSOM . . . . . 30

**3 Navegação Robótica e Autonomia** **33**

3.1 Abordagens para a Navegação . . . . . 33
3.2 Tipos de Mapas . . . . . 39
3.3 Tipos de Navegação . . . . . 39
3.4 Planejamento de Rota e Planejadores . . . . . 40
3.5 Simultaneous Localization and Mapping (SLAM) . . . . . 40
3.6 SLAM Visual . . . . . 41
3.7 Detecção Visual de Circuito Fechado . . . . . 42
3.8 Definição Formal do Problema Tratado pela Tese . . . . . 43
3.9 Métricas de Avaliação e Benchmarks . . . . . 44
3.9.1 Métricas de Avaliação . . . . . 44
3.9.2 Proposta de Avaliação do Sistema . . . . . 45

**4 Proposta de Solução** **47**

4.1 Introdução . . . . . 47
4.2 O Mapa Auto-Organizável de Cobertura Rápida de Espaço de Estados . . . . . 48
4.2.1 Pseudocódigo do FSS-SOM . . . . . 54
4.3 Sistema de Localização Visual e Mapeamento Topológico . . . . . 56
4.3.1 Descrição de Características Visuais de um Local . . . . . 58
4.3.2 Descrição do I-MAP . . . . . 59
4.3.3 Codebook-IP e Codebook-PI . . . . . 60

4.3.4 Critério de Semelhança entre dois Locais . . . . . 61
4.3.5 Descrição do P-MAP . . . . . 61

**5 Resultados** **64**
5.1 Metodologia para Validar o Desempenho do Mapa . . . . . 64
5.2 Resultados dos Testes com o FSS-SOM . . . . . 68
5.3 Metodologia Experimental para Validar o Sistema Proposto . . . . . 78
5.4 Resultados do Sistema de Localização Visual e Mapeamento Topológico . . . . 79
5.4.1 Avaliação da Medida de Semelhança do Sistema . . . . . 79
5.4.2 Avaliação da Capacidade de Localização Visual . . . . . 81
5.4.3 Avaliação da Capacidade de Localização Visual e Mapeamento topológico 89
5.5 Discussões dos resultados . . . . . 94

**6 Conclusão** **96**
6.1 Contribuições para a Ciência . . . . . 97
6.2 Trabalhos Futuros . . . . . 97

**Referências** **99**

**Apêndice** **106**

**A Plataforma Robótica** **107**
A.1 Plataforma Robótica . . . . . 107
A.2 Protocolo de comunicação PC-Plataforma Robótica . . . . . 109
A.3 Vistas da Plataforma Robótica . . . . . 112

# 1

# Introdução

Robôs com autonomia suficiente para planejar uma rota e se locomover de modo seguro entre uma origem e um destino apresentam um potencial econômico imediato e importância social por permitirem a automatização de atividades fundamentais como: transporte de cargas e de passageiros. Devido a estes aspectos, dotar máquinas de capacidade de navegação autônoma é um problema antigo e importante em áreas como a robótica e inteligência artificial (KRIEGMAN D. TRIENDL, 1989).

Uma das primeiras grandes iniciativas científicas no sentido de resolver este problema foi o projeto EUREKA Prometheus (GRAEFE; KUHNERT, 1992; GRAEFE, 1993). Este projeto de pesquisa e desenvolvimento na área de veículos autônomos foi financiado pela União Europeia entre 1987 e 1995 e consumiu o equivalente a 1 bilhão de dólares. O projeto envolveu várias universidades e fabricantes europeus de veículos. Um dos participantes do projeto foi o pioneiro em veículos sem motorista Ernst Dickmanns da Bundeswehr University of Munich em parceria com o fabricante de automóveis Dailmler-Benz (DICKMANNS et al., 1994).

No projeto EUREKA Prometheus, foram desenvolvidos sistemas de auxílio a navegação de automóveis em autoestradas para evitar colisão, para seguir veículos, rastrear automóveis na pista, identificar pista livre e realizar ultrapassagens controladas. Apesar do grande investimento, o projeto EUREKA não resultou, diretamente, em um produto comercial logo após o seu término em 1995. Somente em 2010 um produto comercial disponibilizado pela Volkswagen chamado Lane Assist (EURONCAP, 2010) conseguiu apresentar os resultados semelhantes aos obtidos no projeto EUREKA.

Uma outra grande iniciativa na área de navegação foi lançada pelo governo dos Estados Unidos em 2004 através de sua agência de projetos avançados de defesa dos Estados Unidos (DARPA), que promoveu a primeira grande competição de longa distância entre veículos completamente autônomos, o Grand Challenge 2004. A competição consistia em uma corrida de 240km *off-road* com veículos sem motoristas e completamente autônomos no deserto de Mojave, a equipe do primeiro veículo a alcançar a linha de chegada ganharia um prêmio de 1 milhão de dólares. O aparato sensorial predominante na competição consistiu na utilização de radares lasers para medir distâncias entre obstáculos e o veículo e GPS para informar a posição do veículo em

um mapa (THRUN et al., 2006a).

Os sensores de distância lasers foram utilizados na maioria das abordagens para, essencialmente, evitar colisões com objetos não mapeados. Sendo o GPS o principal equipamento utilizado para prover as informações necessárias para a localização do veículo em um mapa topográfico em 3D, já previamente fornecido. Nesta competição o principal desafio proposto pela DARPA era produzir um robô que fosse capaz de, tendo um mapa 3D do terreno, traçar uma rota factível entre dois pontos e realizá-la no menor tempo possível, sem colisões e sem oferecer risco ao veículo autônomo e sua carga útil.

Nenhum dos participantes conseguiu terminar a competição. Tony Tether, então diretor de desenvolvimento da DARPA, anunciou após o primeiro desafio em 2004 que a DARPA realizaria uma nova competição em 2005, o Grand Challenge 2005, com o dobro da premiação de 2004, 2 milhões de dólares. Desta vez, todas as equipes participantes completaram o percurso estabelecido. As equipes das universidades de Stanford (veículo Stanley) e Carnegie Mellon (veículos SandStorm e H1ghlander) ficaram com as três primeiras colocações na corrida, respectivamente. A equipe de Stanford foi liderada pelo Ph.D e cientista da computação Sebastian Thrun, as equipes de Carnegie Mellon foram lideradas pelo Ph.D em engenharia civil William L. Whittaker.

É interessante notar que Thrun e Whittaker foram colegas de pesquisa em Carnegie Mellon tendo vários trabalhos na área publicados em conjunto antes da ida de Sebastian Thrun para Stanford e, essencialmente, as abordagens de Thrun e Whittaker para o problema de navegação autônoma compartilhavam a mesma fundamentação. Ambas se baseavam em teoria de controle e derivações de filtros de Kalman (URMSON et al., 2007; THRUN et al., 2006b; DAUM, 2005). Assim, pode-se afirmar que os 3 primeiros colocados do desafio de 2005 empregavam, na realidade, abordagens muito próximas ou que no mínimo derivavam de uma linha de pesquisa comum que se baseava predominantemente em algoritmos de localização e mapeamento simultâneo (SLAM) com filtros estocásticos (THRUN et al., 2006a).

Após o bem sucedido desafio de 2005, a DARPA anunciou outro desafio em cenário urbano, o Grand Challenge 2007. Onde os veículos autônomos deveriam apresentar habilidades, além de planejar rotas e desviar de obstáculos estáticos, desviar de outros veículos automotivos e respeitar a sinalização de trânsito. Os maiores desafios nesta etapa foram implementar máquinas capazes de se deslocarem em trechos urbanos sem saírem das vias e sem colidir com outros veículos próximos.

Novamente as equipes de Carnegie Mellon e Stanford figuraram como as melhores colocadas. A equipe da universidade de Carnegie Mellon (Tartan Racing) liderada por Whittaker obteve a primeira colocação seguida pela equipe de Stanford (Stanford Racing) liderada por Sebastian Thrun, em segundo lugar na competição.

Os desafios de navegação autônoma promovidos pela DARPA geraram grande repercussão e empresas como o Google manifestaram interesse em desenvolver os seus próprios projetos de navegação autônoma para aplicações comerciais.

O Google DriverLess Car é um projeto do Google para veículos sem motoristas liderado pelo cientista da computação Sebastian Thrun (THRUN, 2011, 2015) . Outros nomes envolvidos no projeto são: Chris Urmson, Mike Montemerlon e Anthony Levandowski. Todos trabalharam com Thrun no DARPA Grand Challenge e no Urban Challenge. O principal objetivo do Google DriverLess Car é fornecer um serviço de transporte baseado em veículos autônomos. Entretanto o conceito de autonomia utilizado pelo Google é limitado. O carro do Google é capaz apenas de circular em percursos predeterminados, mapeados e sob a supervisão de dois operadores humanos que podem assumir remotamente o controle do carro a qualquer momento que seja necessário (RYAN, 2012). Todo o projeto do carro sem motorista do Google é mantido sob um forte esquema de sigilo industrial tornando difícil uma avaliação real do quanto o carro do Google se distingue em autonomia de um simples veículo operado remotamente (ROV).

Além do desenvolvimento tecnológico na área de carros sem motorista, o Google incentivou o desenvolvimento de legislações em vários estados americanos que permitissem e regulamentassem a utilização desta tecnologia. Em junho de 2011, o estado de Nevada foi o primeiro a aprovar uma lei que permitia a utilização de veículos sem motoristas, a lei passou a valer a partir de março de 2012 (SMITH, 2012). Em maio de 2012 o estado de Nevada licenciou o primeiro carro sem motorista, um Toyota Prius modificado com tecnologia experimental do Google. Em 2010, o Google alegava que os carros com a sua tecnologia já possuem mais de 1609km de direção sem nenhuma intervenção humana e 225308km com intervenções eventuais, ou seja, apenas em aproximadamente 0.7% do total de sua quilometragem o Google DriveLess Car realmente dirigiu sem intervenção humana (GOMES, 2014). Isto mostra o quão grande ainda são os desafios na área de navegação autônoma.

O conceito de autonomia varia bastante dependendo da aplicação ou abordagem. Neste trabalho adotamos a definição de autonomia dada por ARKIN (1998) que diz que um agente autônomo: “é uma máquina capaz de perceber informações sobre seu ambiente e usar o seu conhecimento sobre esse ambiente para mover-se de forma segura, de modo significativo e intencional”. Assim, atualmente veículos autônomos já são uma realidade, mas em ambientes controlados e com algum grau de supervisionamento e auxílios externos (ex: utilizando mapas já elaborados e/ou interventores humanos) para poder realizar a navegação.

Um elemento chave nas abordagens mais utilizadas atualmente é a capacidade do sistema de navegação autônoma resolver o problema de construção de um mapa do ambiente onde se encontra, utilizando aparato sensorial próprio, e de se localizar neste mapa. Este problema de construção do mapa e de se localizar no mapa construído é denominado de *Simultaneous localization and mapping* ou SLAM.

Atualmente, a grande maioria dos sistemas de navegação utilizam um mapeamento métrico, ou seja, na resolução do problema de SLAM o objetivo final é ter um mapa tipo topográfico em 3D do ambiente onde o agente está se locomovendo. A partir deste mapa, um sistema de roteamento e detecção de obstáculos é utilizado para traçar uma rota factível do ponto de origem até o destino final do agente robótico. Devido a este tipo de abordagem, há uma

forte dependência de um aparato sensorial que seja capaz de produzir uma visualização em 3D precisa. Normalmente o aparato sensorial utilizado para produzir tais visualizações é composto por um ou mais lidares (BUEHLER; IAGNEMMA; SINGH, 2009). Lidares são um tipo de radar que utiliza luz laser no lugar de ondas de rádio para detectar obstáculos no espaço. Devido à utilização de luz com um comprimento de onda centenas de vezes menor que uma onda de rádio, os lidares possuem resolução espacial da ordem de milímetros e o efeito de reflexão da luz do laser não ocorre apenas em superfícies com boa condutividade elétrica, como no caso dos radares. Entretanto, estes sensores apresentam as seguintes desvantagens: grande quantidade de dados a ser processada, são muito caros, lentos (baixo *frame rate*), pesados e consomem muita energia. Um outro problema é que por serem sensores ativos podem provocar interferência mútua em outros lidares próximos, por exemplo, em um ambiente urbano com vários veículos autônomos utilizando o mesmo tipo de sensor em uma mesma vizinhança.

A abordagem métrica baseada em um preciso conhecimento das dimensões do ambiente utilizando lidares, mapas em 3D e coordenadas de GPS não aparenta ser uma solução "natural" para o problema de navegação. Humanos, por exemplo, não necessitam de lidares para se deslocar e se localizarem em um determinado ambiente. A visão binocular humana, trás alguma informação espacial do mundo, mas com uma exatidão muito menor que a de um lidar e, essencialmente, produz a sensação de profundidade para objetos a poucos metros de distância (GONZALEZ; WOODS, 2010). Assim, podemos supor que, basicamente, utilizamos informações visuais em 2D em conjunto com uma noção grosseira de profundidade produzida pela visão binocular humana. Humanos, também, não necessitam de informações do ambiente com precisão métrica para realizar as mesmas tarefas de navegação, determinação de rotas e evitar colisões com objetos estáticos ou dinâmicos. Nesta perspectiva "humana", a utilização e a construção de mapas métricos em 3D de um ambiente para subsidiar atividades de navegação e localização aparenta ser algo desnecessário e custoso computacionalmente. A investigação de requisitos sensoriais e mecanismos de processamento destas informações para auxílio à navegação autônoma ainda possuem grandes lacunas e problemas não resolvidos (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011; ROMERO et al., 2014). Sensores como lidares e GPSs são abordagens sem correspondência no contexto biológico e não é por essa limitação que animais e outros sistemas biológicos não são capazes de realizar navegações autônomas com primazia.

Um dos problemas em aberto atualmente é projetar um sistema de navegação autônoma biologicamente inspirado ou plausível que utilize apenas os mesmos recursos sensoriais que um animal é capaz de utilizar, como: visão monocular, binocular, sentido de aceleração, equilíbrio, tato e direção magnética. Assim, se forem consideradas tais restrições sensoriais e a definição de autonomia de ARKIN (1998) o problema de navegação autônoma ainda não apresenta solução geral.

Neste trabalho, o enfoque principal é propor um novo método de localização e mapeamento não métrico para sistemas de navegação autônoma mais próximo de um contexto biológico

e humano (ou seja com conjuntos sensoriais semelhantes aos humanos). Desta forma, o método deve ser baseado em redes neurais, não necessitar de informações tridimensionais precisas e tirar proveito de imagens bidimensionais.

## 1.1 Enunciado do Problema

O problema tratado nesta tese é: dado um sequências temporal de observações visuais de um ambiente, identificar os locais visitados pelo agente que realizou a aquisição das observações e determinar as relações de vizinhança topológica destes locais. O problema apresenta alguns desafios como: a memorização e identificação visual de locais e o mapeamento topológico visual de um ambiente. Uma questão das mais óbvias para a identificação visual é que os locais observados podem conter elementos dinâmicos, como pessoas se movimentando em uma rua, que podem alterar significativamente o conteúdo das observações. As condições de iluminação também podem variar dependendo da hora que as observações são realizadas. Outro complicador é que determinadas características visuais podem ser idênticas em dois ambientes completamente diferentes, por exemplo, uma placa de trânsito com o símbolo "proibido estacionar" pode estar presente em diversas ruas de cidades diferentes.

Assim, este problema de determinação de locais a partir de sequências visuais está muito além de um simples sistema de casamento de imagens ou de busca por imagens. Este problema envolve aprender características marcantes que identificam unicamente uma localidade em um percurso. Outra característica que diferencia este problema de uma busca por imagens convencional é que as características marcantes são identificadas sobre a sequência de observações e não apenas em uma única observação. Um sistema com a habilidade de tratar tal problema pode ser utilizado para realizar mapeamento topológico visual de um ambiente através de sucessivos reconhecimentos de locais adjacentes, funcionando efetivamente como um sistema de SLAM Visual em Espaço de Aparência (CUMMINS; NEWMAN, 2008a). Podendo ser utilizado como suporte de um sistemas de navegação autônoma visual e em conjunto com um sistema de SLAM tradicional, que visa obter estimativas de coordenadas de posição.

## 1.2 Objetivos

O objetivo desta tese é propor e implementar um sistema capaz de prover localização visual e mapeamento topológico para um sistema de navegação e exploração autônoma utilizando mapas auto-organizáveis com topologia variante no tempo (ARAUJO; REGO, 2013) capazes de detectar sequencialidade temporal em entradas de estímulos sensoriais, como, por exemplo, imagens de vídeo.

Nas revisões bibliográficas realizadas foram encontrados modelos neurais autoassociativo simples baseados em estímulos visuais e odometria como o RatSlam (MILFORD; WYETH; PRASSER, 2006), que são capazes de realizar localização e navegação em ambientes *indoor* e

*outdoor* e propostas de sistemas baseados em mapas auto-organizáveis capazes de realizar navegação em ambientes simulados utilizando sensores de varredura 3d (MOLE, 2010). Sistemas de localização e detecção de circuitos fechados visuais (CUMMINS, 2009) e sistemas que realizam odometria visual e SLAM, (CABALLERO et al., 2009), em ambientes externos para navegação de pequenas aeronaves não tripuladas também estão presentes na literatura. Entretanto, não foi encontrado nenhum sistema que utilize auto-organização para lidar com padrões sequenciais, como o MSom (STRICKERT; HAMMER, 2005), para realizar as tarefas de localização em uma rota fechada ou identificar padrões visuais repetitivos na exploração de um ambiente. Assim, esta tese propõe um novo modelo de mapa auto-organizável que pode ser utilizado em sistemas de localização visual e mapeamento topológico com aplicações na área de navegação autônoma. Os objetivos deste trabalho podem ser divididos do seguinte modo:

### 1.2.1 Objetivo primário

- Propor um novo modelo de mapa auto-organizável dinâmico com crescimento e treinamento rápido que possa ser utilizado em um sistema de SLAM Visual em Espaço de Aparência.

### 1.2.2 Objetivos secundário

- Propor um sistema baseado em redes neurais não supervisionadas capaz de realizar localização visual e mapeamento topológico baseado unicamente em imagens monoculares, constituindo efetivamente em um SLAM Visual em Espaço de Aparência.
- Realizar comparações do desempenho da abordagem proposta com abordagens semelhantes de localização visual.

## 1.3 Estrutura do Documento

Esta tese está estruturada em 6 capítulos. O Capítulo 1 apresenta uma introdução, motivações, enuncia o problema concretamente tratado e os objetivos da tese. O Capítulo 2 apresenta uma pequena introdução sobre redes neurais, detalhando mais as redes neurais auto-organizáveis. O Capítulo 3 apresenta uma revisão cronológica de vários artigos na área de navegação robótica ao longo dos últimos 20 anos, mostrando como foi a evolução das abordagens de navegação, conceitos básicos sobre navegação robótica e apresenta a modelagem, a definição formal do problema, as métricas e os *benchmarks* utilizados. No Capítulo 4, são apresentados um novo modelo de Mapa Auto-organizável (FSS-SOM, Seção 4.2) e um modelo neural, utilizando o FSS-SOM, para tratar do problema de localização e mapeamento visual. O Capítulo 5 apresenta os resultados dos testes com o modelo proposto de localização visual e mapeamento topológico. O Capítulo 6 apresenta as conclusões e considerações finais. No Apêndice A é apresentado

um protótipo de sistema robótico móvel desenvolvido para testar algoritmos de navegação. Este sistema foi utilizado para coletar dados para os testes em ambiente *indoor* no Centro de Informática da UFPE.

## 1.4 Publicação

O FSS-SOM, apresentado no Capítulo 3, foi publicado e apresentado no BRACIS 2014 (D. F. GOMES, 2014).

# 2

# Redes Neurais

## 2.1 Introdução

As arquiteturas de redes neurais artificiais podem, basicamente, serem divididas em três categorias, cada uma baseada em uma filosofia diferente. Na primeira categoria, as *feedforward networks* (RUMELHART; HINTON; WILLIAMS, 1986), ou redes alimentadas para frente, transformam os conjuntos de sinais de entrada em conjuntos de sinais de saída. A transformação entrada-saída desejada é usualmente determinada por um processo de ajuste externo de parâmetros do sistema. Exemplos muito populares de redes que pertencem a esta categoria são a *Multi Layer Perceptron* (MLP) e as Redes de Função de Base Radial (RBFs). Na segunda categoria, as *feedback networks* (HOPFIELD, 1982), ou redes retroalimentadas, o sinal de entrada define o estado inicial de ativação do sistema de realimentação e, depois, o sistema passa por um processo de sucessivas transições de estado que, assintoticamente, conduzem a um estado final estável que é definido como a saída da rede. Na terceira categoria de modelos neurais, células vizinhas em uma rede neural competem entre si através de interações mútuas laterais e desenvolvem adaptativamente a capacidade de detectar diferentes padrões de sinais sem nenhum tipo de supervisão (KOHONEN, 1990). Esta categoria é baseada em aprendizagem competitiva (*competitive learning*).

### 2.1.1 Abordagem Não Supervisionada e Mapas Auto-Organizáveis

Os Mapas Auto-organizáveis, *Self-Organizing Maps* ou, simplesmente SOMs pertencem a terceira e última categoria de redes neurais. Existem diversas aplicações para este tipo de rede neural, mas as principais aplicações são: visualização de dados com alta dimensionalidade, redução de dimensionalidade de um conjunto de dados e categorização. A operação que um mapa auto-organizável realiza sobre um determinado conjunto de dados é um tipo de projeção não-linear da função de densidade de probabilidade destes dados. Normalmente a projeção é realizada em um espaço de dimensionalidade menor que a dos dados originais, mas isso não é uma condição necessária. Os dados aprendidos pelos SOMs distribuem ao longo dos nodos

do mapa numa configuração chamada de distribuição topológica. Esta característica faz com que, de modo geral, nodos próximos apresentem informações similares e, por sua vez, nodos distantes apresentam informações distintas. A estrutura básica dos modelos auto-organizáveis é formada por uma camada de entrada e uma camada de saída. A camada de entrada propaga os valores recebidos a todos os nodos da camada de saída, na qual é criada a representação do conhecimento. O processo de aprendizado dos SOMs além de não-supervisionado é competitivo e cooperativo. Isto é, os nodos da rede na camada de saída competem entre si pelo direito de serem atualizados. Apenas o nodo vencedor (competição), e em alguns casos seus vizinhos (cooperação), são atualizados. O nodo vencedor é aquele que apresenta maior semelhança, proximidade, ao padrão de entrada em questão e métricas diferentes podem ser usadas para mensurar o grau de proximidade.

#### 2.1.1.1 Mapas de Kohonen

O Kohonen SOM, SOM "clássico" ou, simplesmente, SOM deve-se a KOHONEN (1990). Este foi um dos primeiro pesquisadores a realizar um extenso trabalho sobre redes neurais artificiais não-supervisionadas nas décadas de 1970 e 1980. O funcionamento básico do SOM define um mapeamento partindo de um conjunto de dados de entrada no espaço $\mathbb{R}^n$ para uma malha, normalmente, bidimensional de nodos. Cada nodo $i$ da malha possui um vetor de pesos $\mathbf{w_i} \in \mathbb{R}^n$. Um vetor de entrada $\xi \in \mathbb{R}^n$ é comparado com todos os $\mathbf{w_i}$'s e o mais próximo do vetor de entrada é definido como sendo o *best-matching unit* (BMU) e o nodo que contém este vetor é considerado como o nodo "vencedor" ou mais apto a representar o vetor de entrada $\xi$, desta forma é realizada uma associação entre o vetor $\xi$ e o nodo $i$, ver Figura 2.1. A métrica de distância entre $\xi$ e $\mathbf{w_i}$ pode variar, normalmente é utilizada a distância Euclidiana. A saída da rede, depende essencialmente da aplicação, pode ser o próprio nodo ($i$) vencedor, o peso ($\mathbf{w_i}$) associado ao nodo vencedor, ambos ou parte da informação contida no peso do nodo vencedor. O algoritmo de treinamento do modelo de Kohonen está descrito no Algoritmo 1. Os passos principais consistem em sortear uma amostra do conjunto de treinamento (linha 2), encontrar no mapa o nodo mais próximo da amostra sorteada (linha 3) e atualizar o valor do nodo mais próximo e dos seus vizinhos de modo que a distância entre os nodos e a amostra diminua (linha 4). O modo como os nodos são atualizados varia exponencialmente de acordo com o tempo e a distância dos nodos para a amostra (Equações 2.3 a 2.6). Estes passos básicos estão presentes na maioria dos modelos de mapas auto-organizáveis. O modelo de Kohonen foi a base para a implementação de muitas outras variações de mapas auto-organizáveis.

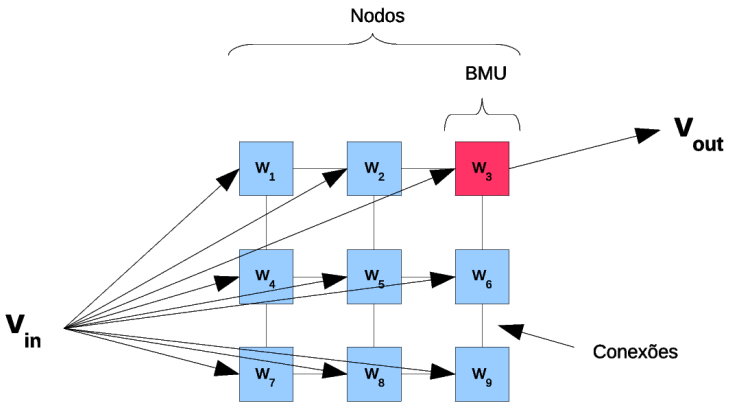


**Figure 2.1:** Ilustração de uma rede SOM, com um vetor de entrada $\mathbf{v_{in}}$ sendo comparado com os vetores dos nodos $\mathbf{w_n}$ e a escolha do BMU.

O SOM, apesar de ter se mostrado um potente modelo para categorização e ainda ser muito utilizado, apresenta algumas restrições importantes, algumas delas são:

- O número de nodos do mapa precisa ser conhecido previamente, sendo um parâmetro do sistema;
- O padrão de conexões entre os nodos é fixo e precisa ser determinado previamente.

Devido a estas restrições muitas abordagem foram desenvolvidas com o objetivo de implementar mapas auto-organizáveis com topologia e número de nodos dinâmicos, nas subseções seguintes serão apresentados dois destes modelos de mapas dinâmicos.

**Algorithm 1** Kohonen SOM

1: *Inicialização:* Escolha valores aleatórios iniciais para os vetores de peso $\mathbf{w_j}(0)$. A única restrição é que os valores de inicialização sejam sempre diferentes para os $\mathbf{w_j}(0)$ pesos, onde $j = 1,2,...,N$ e $N$ é o número de neurônios do mapa.

2: *Amostragem:* Apresenta um vetor de dados escolhido aleatoriamente, $\xi$, como entrada à rede.

3: *Casamento por similaridade:* Encontra o nodo mais próximo do vetor de dados escolhido utilizando o critério da menor distância Euclidiana.

$$i(\xi) = arg\,min_j \|\xi - \mathbf{w_j}\| \therefore j = 1,2,...,N \tag{2.1}$$

onde $N$ é o número total de nodos do mapa e $i(\xi)$ é o número do nodo mais próximo do vetor apresentado.

4: *Atualização:* Atualiza os pesos de cada nodo da rede utilizando a formula:

$$\mathbf{w_j}(n+1) = \mathbf{w_j}(n) + \eta(n) h_{j,i(\xi)}(n)(\xi(n) - \mathbf{w_j}(n)) \tag{2.2}$$

onde,

$$\eta(n) = \eta_0 e^{-\frac{n}{\tau_2}} \tag{2.3}$$

$$h_{j,i(\xi)}(n) = e^{\frac{d_{j,i}^2}{2\sigma^2(n)}} \tag{2.4}$$

$$\sigma(n) = \sigma_0 e^{-\frac{n}{\tau_1}} \tag{2.5}$$

$$d_{j,i}^2 = \|\mathbf{r_j} - \mathbf{r_i}\|^2 \tag{2.6}$$

$\eta$ é o parâmetro de aprendizagem, $h_{j,i(\xi)}$ é a função de vizinhança entorno do vencedor, $\sigma$ é a função de decaimento exponencial ao longo do tempo e $d_{j,i}^2$ é a distância topológica entre o nodo vencedor (i) e o nodo que está sendo atualizado (j).

5: *Continuação:* Continua a partir do passo 2 enquanto são notadas mudanças significativas nas posições dos vetores de pesos dos nodos

#### 2.1.1.2 Growing Neural Gas

O *Growing Neural Gas* ou, simplesmente, GNG é um modelo de mapa auto-organizado incremental desenvolvido por FRITZKE (1995) derivado a partir dos modelos de Martinetz e Schulten. O GNG cresce e adapta as conexões dos nodos de modo incremental para se ajustar a função densidade de probabilidade definida por um determinado conjunto de treinamento. O processo de crescimento ou de inserção de novos nodos é constante e múltiplo de um número ($\lambda$) de apresentações de exemplos do conjunto de treinamento. A inserção dos novos nodos é sempre realizada na vizinhança do nodo com o maior "erro" acumulado. Pelo fato do desempenho do GNG ter se mostrado significativo para uma diversidade de problemas muito grande, este

modelo se tornou praticamente uma referência de desempenho para realizar comparações com outros modelos dinâmicos de mapas. O processo de aprendizagem do GNG está ilustrado no Algoritmo 2. No GNG, os procedimento de inicialização, amostragem, busca do vetor de pesos e atualização de pesos, respectivamente linhas 1, 2, 3 e 6 são semelhantes aos adotados no mapa de Kohonen. A principal diferença esta no procedimento de envelhecimento de arestas de nodos pouco utilizados, linha 4, e no contador local de erro, linha 5. É baseado no erro acumulado em cada nodo e no envelhecimento das arestas que o GNG determina onde devem ser inseridos novos nodos e quais as conexões que devem ser desfeitas. O procedimento de inserção de novos nodos esta descrito na linha 9.

**Algorithm 2** GNG-SOM

1: *Inicialização do mapa:* Inicia o mapa com dois nodos com vetores de pesos aleatórios $\mathbf{w_a}$ e $\mathbf{w_b}$ no espaço $\mathbb{R}^n$

2: *Amostragem:* Selecione uma amostra $\xi$ de acordo com uma dada probabilidade $P(\xi)$ do conjunto de dados de treinamento.

3: *Casamento por similaridade:* Encontre o nodo, $s_1$, com os vetores de peso mais próximo e o segundo mais próximo, $s_2$, da amostra $\xi$.

4: *Envelhecimento:* Incremente a *age* de todas as conexões emanando de $s_1$.

5: *Incremento do contador local:* Incremente o contador local com o quadrado da distância entre a amostra e o nodo mais próximo ($s_1$).

$$counter(s_1) = counter(s_1) + \Delta erro(s_1) \text{ , onde } \Delta erro(s_1) = \|\mathbf{w_{s_1}} - \xi\|^2 \tag{2.7}$$

6: *Atualização dos pesos:* Movimente o nodo $s_1$ e os seus vizinhos topológicos em direção da amostra $\xi$ pela fração de $\varepsilon_b$ e $\varepsilon_n$, respectivamente, da distância total.

$$\Delta \mathbf{w_{s_1}} = \varepsilon_b(\xi - \mathbf{w_{s_1}}) \tag{2.8}$$

$$\Delta \mathbf{w_n} = \varepsilon_n(\xi - \mathbf{w_n}) \tag{2.9}$$

onde,

$$\varepsilon_b > \varepsilon_n \tag{2.10}$$

7: *Conectando os nodos mais próximos:* Se $s_1$ e $s_2$ estão conectados, iguale a zero a *age* da conexão. Se a conexão não existe, crie-a.

8: *Remoção de conexões antigas:* Remova uma dada conexão se tiver uma *age* maior que $age_{max}$. Se isto resultar em nodos sem nenhuma conexão remova o nodo também.

9: *Inserção de novos nodos:* Se o número de amostragens realizadas for múltiplo de um inteiro $\lambda$, insere um novo nodo da seguinte maneira:

- Determine o nodo $q$ com o maior erro acumulado.
- Insira um novo nodo $r$ no ponto médio entre o nodo $q$ e o nodo vizinho $f$ com o maior erro acumulado.

$$\mathbf{w_r} = 0,5(\mathbf{w_q} + \mathbf{w_f}) \tag{2.11}$$

- Insira uma conexão nova ligando $r$ a $q$ e a $f$ e remova a conexão original entre $q$ e $f$.
- Decremente o a variável de erro de $q$ e $f$ pela multiplicação por um fator constante de $\alpha$ e inicialize o erro de $r$ com o mesmo valor do erro de $q$, já atualizado.

10: *Decremento do erro:* Decremente todas as variáveis de erro pela multiplicação delas por uma constante $\beta$.

11: *Critério de parada:* Finalize o treinamento se o critério de parada foi atingido, se não, continue a partir de 1 (O Algoritmo do GNG não fixa nenhum critério de parada específico, pode ser o tamanho da rede, alguma medida de performance, entre outros).

#### 2.1.1.3 LARFSOM

O LARFSOM é um tipo de de mapa auto-organizável projetado originalmente para trabalhar com imagens coloridas. Ele pode ser utilizado tanto para realizar redução de cores em uma imagem como para realizar segmentação por cor. O algoritmo básico de treinamento do LARFSOM tem 11 passos. Há, também, variações que envolvem a parte final do algoritmo (Linha 11), uma descrição mais detalhada pode ser encontrada na dissertação de COSTA (2007) ou no artigo ARAúJO; COSTA (2009). O LARFSOM está descrito no Algoritmo 3. A ideia básica do LARFSOM é ter um campo receptivo adaptativo que varia de acordo com a densidade dos dados, linha 5. Assim, o LARFSOM tende a inserir mais nodos em regiões onde há uma maior densidade de amostras de treinamento e inserir poucos nodos em regiões de baixa densidade. Na linha 4 do algoritmo é calculada a ativação $a_{s_1}$ do nodo $s_1$ com vetor de pesos mais próximo da amostra $\xi$. A inserção de novos nodos no mapa ocorre se a ativação $a_{s_1}$ for menor que um limiar $a_T$.

**Algorithm 3** LARFSOM

1: Inicialize os parâmetros: taxa de aprendizagem final ($\rho_f$); modulador de taxa de aprendizagem ($\varepsilon$); limiar ou valor de corte de atividade ($a_T$); número de vitórias de cada nodo i ($d_i = 0$); número máximo de vitórias de cada nodo ($d_m$); iteração de tempo ($t = 0$); tempo máximo ($t_{max}$); erro mínimo ($deslocamento_{min}$); e o número de unidades iniciais ($N = 2$), que devem ser conectadas e cujo os pesos podem ser copiados de dois vetores, escolhido aleatoriamente, do conjunto de dados de entrada.

2: Apresente um pixel escolhido aleatoriamente da imagem original, $\xi = [rgb]^T$, como entrada à rede.

3: Calcule a distância Euclidiana entre o vetor de entrada, $\xi$, e o vetor de pesos, $\mathbf{w_i}$, de cada um dos nodos como a seguir:

$$d(\xi, \mathbf{w_i}) = \|\xi - \mathbf{w_i}\|^2 \therefore \|\xi - \mathbf{w_i}\|^2 = (r - w_{ir})^2 + (g - w_{ig})^2 + (b - w_{ib})^2 \tag{2.12}$$

A menor distância entre o vetor de entrada e todos os vetores de pesos de cada nodo, determinada a unidade vencedor ou BMU:

$$d(\mathbf{w_{s_1}}, \xi) \leq d(\mathbf{w_{s_2}}, \xi) \leq d(\mathbf{w_i}, \xi), ... \forall i \in NodeSet \tag{2.13}$$

onde , $NodeSet$ é o conjunto dos nodos.

Incremente o contador de vitórias de unidade vencedora $d_{s_1} = d_{s_1} + 1$

4: Insira uma conexão entre $s_1$ e $s_2$ se não existir.

5: Calcule o campo receptivo de $s_1$:

$$r_{s_1} = \sqrt{(w_{s_1 r} - w_{s_2 r})^2 + (w_{s_1 g} - w_{s_2 g})^2 + (w_{s_1 b} - w_{s_2 b})^2} \tag{2.14}$$

6: Calcule a atividade de $s_1$ por meio de uma função de base radial:

$$a_{s_1} = \frac{e^{-\|\xi - \mathbf{w_{s_1}}\|}}{r_{s_1}} \tag{2.15}$$

7: Insira um novo nodo se a ativação da unidade vencedora estiver abaixo do limiar($a_T$), caso contrário atualize os pesos da BMU Se $a_{s1} < a_T$

- Adicione um novo nodo com vetor de pesos $\mathbf{w_n} = \xi$
- Atualize o número de nodos $N = N + 1$
- Remova a conexão entre $s_1$ e $s_2$
- Calcule as distâncias $d(\mathbf{w_n}, \mathbf{w_{s_1}})$, $d(\mathbf{w_n}, \mathbf{w_{s_2}})$, $d(\mathbf{w_{s_1}}, \mathbf{w_{s_2}})$
- Insira conexão entre os nodos com as duas menores distâncias

Caso contrário atualize os pesos do vencedor:

$$\Delta \mathbf{w_{s_1}} = \rho(\xi - \mathbf{w_n}) \therefore \rho = \begin{cases} \varepsilon \rho_f^{\left(\frac{d_i}{d_m}\right)}, d_i \leq d_m \\ \varepsilon \rho_f, d_i > d_m \end{cases} \tag{2.16}$$

8: Atualize o número de interações ($t = t + 1$) e retorne ao passo 2, a não ser que o critério de parada da Equação 2.17 ou o número máximo de iterações tenha sido atingido.

$$deslocamento = \frac{1}{N} \sum_{i=0}^{N-1} \|\mathbf{w_i}(t) - \mathbf{w_i}(t+1)\|^2 \leq deslocamento_{min} \tag{2.17}$$

9: Remova os nodos desconectados

10: Após o processo de treinamento, atribua a cor representada por cada vetor de pesos dos nodos a uma cor da paleta de cores.

11: Na imagem final, substituir cada cor original dos pixels pela cor mais próxima encontrada na paleta de cores.

# 3

# Navegação Robótica e Autonomia

Neste capítulo são apresentados alguns dos principais trabalhos e conceitos na área de navegação robótica. Na seção 3.1, há uma revisão dos principais trabalhos relacionados com navegação robótica, aproximadamente ao longo dos últimos 15 anos. Na seção 3.2 é discutido o que é um mapa e quais são os tipos, as vantagens e as desvantagens dos mapas mais utilizados em sistemas de navegação robótica. Na seção 3.3, há a definição de algumas das principais classes de sistemas de navegação. A seção 3.4 apresenta o conceito de planejamento de rotas. Nas seções 3.5 e 3.6 são apresentados os conceitos de mapeamento e localização simultâneos (SLAM). Na seção 3.7 é discutido o conceito de detecção de circuito fechado e sua relação com o SLAM. Na seção 3.8 é apresentado a definição formal do problema tratado nesta tese. Por último, na seção 3.9 são apresentadas e discutidas métricas e benchmarks de avaliação para o sistema proposto.

## 3.1 Abordagens para a Navegação

As primeiras abordagens em navegação robótica autônoma partiam do princípio que o agente robótico já possuía um mapa do ambiente ao qual estava explorando (CHOSET et al., 2005) e o aparato sensorial do agente robótico servia apenas para localizar o agente robótico no mapa.

Um sistema de navegação *indoor* baseado em visão estereoscópica e odometria foi proposto por KRIEGMAN D. TRIENDL (1989). Neste modelo, a incerteza de localização baseado nos dados de câmera e odômetro são reduzidos utilizando filtros de Kalman extendidos (EKF). O sistema não é capaz de produzir os próprios mapas, podendo apenas realizar localização em um mapa pré-existente que precisa ser fornecido ao sistema. O mapa não precisa ser uma planta CAD do ambiente, isto é, ele pode apresentar características descritivas simbólicas e imprecisas de objetos. Embora o sistema possa detectar objetos tridimensionais para evitar colisões, os mapas fornecidos para auxiliar a navegação são apenas em duas dimensões, como uma planta baixa do ambiente. O sistema utiliza uma combinação de classificadores gaussianos para determinar as posições relativas dos objetos descritos no mapa e detectados no ambiente. A

inteligência do sistema é baseada em uma máquina de inferência baseada na linguagem simbólica Lisp.

O problema de interconectar deslocamentos através *dead reckoning*[1] e cenários para formar um grande mapa geral é o problema central no SLAM. Em trabalhos como em MONTEMERLO et al. (2002) são utilizados filtros estocásticos (como filtros de partículas ou de Kalman e suas variações) (DAUM, 2005) para desempenhar a tarefa de alinhamento e encaixe de várias partes de um mesmo ambiente. Uma característica da abordagem de Montemerlo é contar com sensores de integração (*dead reckoning*) e sequencialidade temporal, para produzir um referenciamento local, ainda que impreciso para auxiliar no processamento das "conexões" de mapeamento local [2] e produzir um mapa global [3].

O problema de um agente robótico identificar sua posição atual e simultaneamente construir mapas do ambiente desconhecido no qual ele está inserido, usando apenas informações locais produzidas pelo próprio sistema sensorial do agente já foi considerado anteriormente (DISSANAYAKE et al., 2001; MONTEMERLO et al., 2002; MILFORD; WYETH; PRASSER, 2004) como o problema central na área de veículos autônomos. Este problema é conhecido resumidamente pela sigla inglesa SLAM (Simultaneous Localization and Mapping). DISSANAYAKE et al. (2001) provaram que o problema SLAM é solucionável utilizando apenas informações locais e integração de caminho e subsequentemente surgiram várias proposições de algoritmos baseados em filtros de Kalman e filtros estendidos de Kalman (CHOSET et al., 2005).

MONTEMERLO et al. (2002) apresentaram um algoritmo rápido para o problema de SLAM. Muitas abordagens até 2002 que utilizavam filtros de Kalman estendidos para realizar SLAM (EKF-SLAM) apresentavam problemas computacionais de acordo com o aumento da área coberta pelo agente robótico devido à complexidade computacional ser do tipo $O(K^2)$, onde $K$ é o número de marcas (*landmarks*). Montemerlo et al apresentaram outra solução baseada em filtros de partícula[4] de Rao-Blackwellized (DOUCET et al., 2000; MURPHY; RUSSEL, 2001). Originalmente um filtro de Rao-Blackwellized apresenta uma complexidade computacional de $O(M \cdot K)$, onde $M$ é o número de partículas utilizadas no filtro e $K$ é o número de *landmarks*. Entretanto, Montemerlo et al propuseram uma modificação na forma como os coeficientes do filtro são calculados para uma configuração em árvore melhorando o desempenho do filtro[5] para $O(Mlog(K))$. A utilização deste filtro modificado de Rao-Blackwellized para realizar SLAM deu origem ao algoritmo FastSLAM.

MILFORD; WYETH; PRASSER (2004) propuseram um novo modelo computacional com inspiração neurobiológica para realizar SLAM em robôs utilizando informações visuais e de odometria. O modelo denominado RatSlam se baseia no sistema neural de posicionamento

[1] Dead-reckoning é o processo de estimar a posição atual baseado em informações de posições passadas.

[2] Ver Mapeamento Local na seção "Definições de termos utilizados no trabalho"

[3] Ver Mapeamento Global na seção "Definições de termos utilizados no trabalho"

[4] Em estatística filtros de partícula, também conhecidos como métodos de Monte Carlo sequenciais, são técnicas de estimação baseadas em simulação.

[5] A modificação proposta por Montemerlo é muito semelhante a outras utilizadas em transformadas rápidas como a FFT e a FWT.

de roedores (*Place Cells* e *Head-direction Cells*)[6] (ARLEO; GERSTNER, 2000). O modelo neural é do tipo autoassociativo (*competitive attractor network*) semelhante a uma rede neural de Hopfield, no qual cada nodo representa uma bacia de atração ou estado do sistema. A rede neural é dividida em duas partes uma que trata as informações provenientes do odômetro do sistema e outra para as informações visuais (*landmarks*). As duas partes são interligadas por conexões ponderadas através de aprendizagem Hebiana. Em essência, a informação visual é utilizada para corrigir as informações obtidas por integração do odômetro. Os estímulos fornecidos pela odometria são as informações de posição $x$, $y$ e $\theta$ (obtidos por integração). Os estímulos visuais são a informação de distância, direção e cor dos cilindros em relação ao robô no ambiente de navegação. A representação do ambiente capturada pela rede neural é intermediária entre uma relação métrica e topológica, ou seja, relações de distância são grosseiramente capturadas e relações de vizinhança são preservadas de acordo com a topologia real do ambiente.

Milford et al estenderam o modelo RatSlam para utilizar memória de objetivos, *goal memory* (MILFORD; WYETH; PRASSER, 2005), e mapas de experiência, *experience map* (MILFORD; WYETH; PRASSER, 2006), para auxiliar a navegação focada em objetivos. Estes elementos adicionais visam dar suporte ao RatSlam para, por exemplo, determinar quais são as rotas mais curtas para se chegar a uma determinada posição objetivo considerando a posição atual.

MANDUCHI et al. (2005) apresentaram um sistema baseado em visão estereoscópica e scanners lasers capaz de identificar automaticamente obstáculos potencialmente perigosos e tipos de terrenos no percurso de um agente autônomo. O sistema visa dar suporte ao sistema de navegação de modo a evitar colisões e a determinar qual a velocidade ideal de navegação para o agente considerando as características de rugosidade de um terreno *off-road*. A detecção de obstáculos foi implementada com um algoritmo semelhante a um filtro com limiarização e utiliza apenas os dados do scanner laser. O sistema de classificação de terrenos utiliza um classificador de mistura gaussiana e considera informações da câmera e do scanner. As informações visuais consideradas são, essencialmente, as cores que são consideradas como as características principais para a classificação. A informação do scanner laser é utilizada como entrada de um modelo estatístico para detectar vegetação rasteira e é utilizada como elemento discriminante auxiliar no classificador.

CASTELNOVI; ARKIN; COLLINS (2005) propuseram um sistema de controle para determinar a velocidade ideal de deslocamento para robôs autônomos de acordo com as condições de rugosidade do terreno imediatamente à frente de um robô. O sinal de entrada do sistema é a informação de distância do solo até o robô utilizando um scanner laser, o sistema realiza a computação de 4 métricas de rugosidade sobre estes sinais para determinar através de regras do tipos if -then três faixas de rugosidade distintas (plano, intermediário e rugoso) o conjunto de

[6] É importante salientar que Milford em seu trabalho se refere as células de posição e de direção da cabeça(respectivamente *Place Cells* e *Head-direction Cells*) apenas como *"Pose Cells"* essa simplificação provoca um pouco de confusão na busca por referências já que esse não é uma nomenclatura amplamente utilizada.

regras permite que robô acelere enquanto a percepção do terreno for de terreno plano, pare de acelerar em terrenos intermediários e desacelere em terrenos rugosos.

GALINDO et al. (2005) propuseram um método de utilizar informações semânticas em conjunto com mapas métricos utilizados por robôs nas suas respectivas ações navegacionais. O sistema proposto necessita como entrada informações semânticas descrevendo a configuração inicial do ambiente e permitir interpretar descrições de alto nível como (quarto, sala,etc) dadas por humano, permite classificar ambientes de acordo com a disposição dos objetos presentes, tratar ambiguidades e identificar erros de localização baseado em posições típicas de objetos. As informações semânticas são relacionadas com as informações métricas utilizando o conceito de ancoragem (*anchoring*).

CABALLERO et al. (2009) propuseram um sistema visual de odometria para veículos autônomos aéreos que dispensa a utilização de acelerômetros e identificação de *landmarks* e que oferece grande precisão de posicionamento e baixo erro de deriva, podendo ser utilizado inclusive para substituir GPS. O sistema de odometria visual é baseado no cálculo das transformação geométricas entre duas imagens consecutivas. Ou seja, supõe-se que as variação entre duas ou mais imagens consecutivas são suaves e que os elementos da cena, em sua maioria, são preservados. Uma análise de decomposição em valores singulares (SVD) é realizada para determinar as transformação sobre a sequência das imagens e deste modo calcular o deslocamento do agente. O sistema utiliza uma taxa de aquisição de imagens de 10Hz com resolução de 320x240 e visão monocular. Entretanto, o sistema proposto só apresenta um bom desempenho quando o agente está sobrevoando regiões relativamente planas e sem fortes características tridimensionais.

Segundo os estudos realizados por MOLE (2010) o problema de navegação e mapeamento é definido como um processo para construir, concorrentemente, um mapa cognitivo “uma representação do ambiente” e usá-lo no suporte à localização e à navegação pró-ativa. Mole utilizou uma abordagem com inspiração neuro-biológica que emprega redes neurais artificiais. Seu trabalho utilizou um ambiente e um agente robótico virtual. A modelagem matemática do agente robótico de Mole propõe um sistema sensorial bastante genérico e flexível o suficiente para uma grande gama de sensores, entretanto nas simulações e testes do modelo proposto o agente foi dotado de um sensor laser de varredura (LIDAR) explorando um ambiente inicialmente desconhecido.

Enquanto navega, o agente identifica características marcantes em um cenário que lhe permitem reconhecer o local corrente quando revisitado e constrói um mapa incremental desses lugares e suas interconexões. Ainda durante a fase de construção do mapa, o agente usa o estado atual do mapa para se localizar e se orientar, buscando a efetivação de comportamentos de navegação.

O modelo proposto por Mole em nível computacional é composto de quatro camadas completamente conectadas, ver Figura 3.1, (1-Sensorial Perceptivo, 2-Mapeamento Básico, 3-Mapeamento Superior e 4-Processos Cognitivos Superiores) que abordam o problema de

navegação em diferentes níveis de complexidade e processamento.

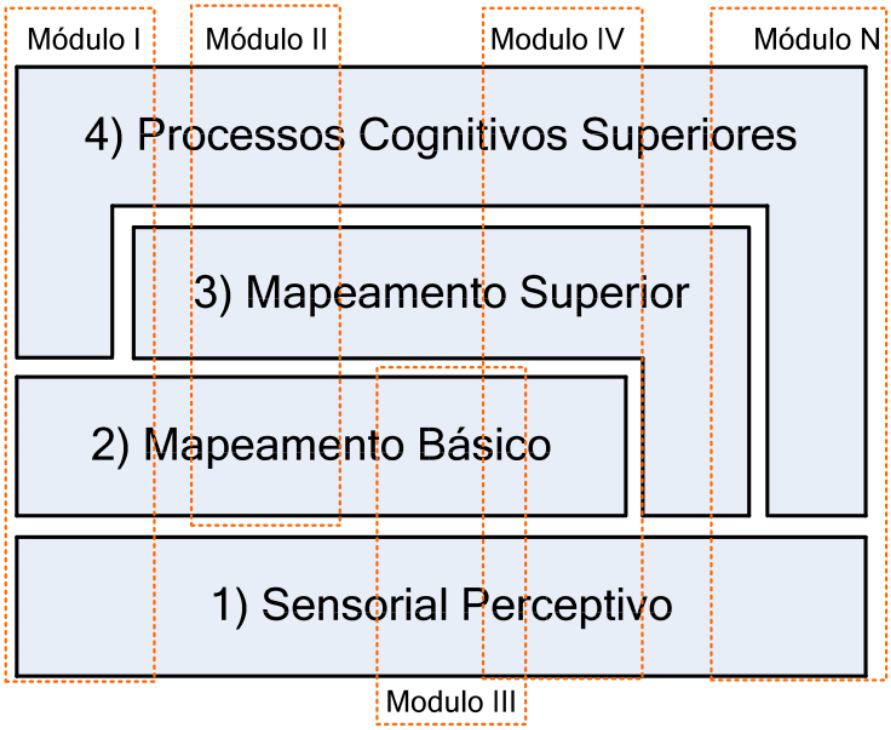


**Figure 3.1:** Módulos do sistema de navegação proposto por Mole. (Imagem retirada da tese de doutorado de MOLE (2010) )

Na camada Sensorial Perceptivo é realizado o pré-processamento dos fluxos sensoriais dos sistemas sensoriais disponíveis para o agente. O modelo matemático desta camada é genérico o suficiente para tratar informações proveniente de sensores diferentes, mas nas simulações, para simplificar a implementação, só foi considerado um sensor do tipo de varredura laser (LIDAR). A camada de Mapeamento básico é destinada a realização de mapeamentos de espaços perceptivos para a posição atual do agente utilizando Mapas Auto-organizáveis. Nas simulações, são construídos mapas auto-organizáveis com informações do ambiente local em que o agente se encontra.

A camada de Mapeamento Superior congrega informações originárias de processos cognitivos e mapeamentos inferiores, produzindo novas representações e abstrações. Podendo realizar fusões de mapas cognitivos locais ou espaços perceptivos produzidos na camada de mapeamento básico gerando representações expandidas. Tais mapeamentos poderiam considerar associações temporais ou de localidade, assim, as representações produzidas nessa camada poderiam, em tese, suportar memórias episódicas.

A camada de Processos Cognitivos Superiores é a que tem por finalidade realizar o processamento mais complexo e utiliza informações de todas as outras camadas. Realizando tarefas complexas de processamento como, por exemplo, a determinação de caminhos entre dois ou mais lugares distintos ou o planejamento e execução de comportamentos exploratórios pró-ativos

de alcance global. Em nível algorítmico o modelo proposto por Mole pode ser dividido em 9 módulos (1-Reconstrução, 2-Representação Topológica, 3-Percepção de Obstáculos, 4-Localização Local, 5 Localização Global, 6-Navegação Local, 7-Navegação Global, 8-Exploração Local, 9-Exploração Global). Destes foram implementados 6 módulos para uma abordagem específica de navegação utilizando scanners laser em ambiente simulado e relacionados, principalmente, com a navegação local. Os módulos responsáveis por realizar tarefas relacionadas a navegação e exploração global (módulos 5, 7 e 9) foram apenas propostos.

O trabalho de Mole serve principalmente como um modelo matemático para um sistema de navegação autônoma geral, mas não fornece muitas informações sobre como implementar ou solucionar problemas importantes como um algoritmo capaz de eficientemente conectar as várias representações de espaços perceptivos produzidas pelo agente ao se deslocar por um ambiente desconhecido. No modelo de mapa proposto por Mole não são consideradas informações de sequencialidade de estados, tendências de transição entre estados e integração de informações locais por odometria. Estas características estão presentes em muitas abordagens como o FastSLAM e o RatSLAM e faz parte do comportamento navegacional da maioria dos animais (WHISHAW; HINES; WALLACE, 2001)

Um sistema de processamento de imagens de baixa resolução (MURALI; BIRCHFIELD, 2012) foi proposto para navegação autônoma *indoor* baseado em visão monocular. Nesta proposta, o sistema é capaz de identificar a distância para o fim e a orientação de um corredor. As imagens utilizadas são monoculares de baixa resolução. A orientação do agente em relação ao corredor é uma função que realiza um mapeamento do tipo $f : I \rightarrow \theta$ onde $\theta$ é o ângulo entre a orientação do agente e do corretor. A função $f$ que realiza esse mapeamento é resultado da combinação de cinco métricas distintas: a entropia da imagem, a simetria como um medida da informação mútua, análise dos locais onde as bordas do corredor se tocam, pontos de fuga usando auto-similaridade da imagem e a mediana dos brilhos dos pixels. Já o sistema que estima a distância para o fim de um corredor utiliza uma combinação de três métricas distintas: *time-to-collision*, divergência de Jeffrey e entropia.

Os números de artigos publicados em periódicos nas áreas relacionadas à navegação e exploração autônoma cresceram de 2001 até atingir um máximo em 2009, sinalizando que já existe uma certa consolidação na área sobre os algoritmos e teorias desenvolvidas Entretanto, todas as principais abordagens vigentes partem do princípio que o agente robótico tem a disposição sensores de posicionamento global, odômetros e sensores de varredura ativa, como Lidares e Radares. O número de abordagens utilizando apenas câmeras e sensores inerciais ainda é menor que as utilizando sensores como Lidares. Evidenciando uma possível área promissora e uma tendência a simplificar o aparato sensorial necessários para dar suporte aos veículos autônomos.

## 3.2 Tipos de Mapas

Mapas são representações simbólicas de um determinado ambiente. Nas abordagens de navegação robótica existem dois tipos de paradigmas de mapeamento: métrico e topológico. Mapas métricos representam o mundo com exatidão, capturando em detalhes muitas das informações e distâncias métricas do ambiente. Estes mapas normalmente são produzidos utilizando um sistema de coordenadas globais. Os mapas métricos normalmente são considerados mais apropriados em aplicações para guiar veículos e para desviar de obstáculos. Entretanto, mapas métricos são mais difíceis de serem construídos e mantidos. A construção destes mapas também é considerada custosa do ponto de vista computacional.

Mapas topológicos representam o ambiente de um modo mais abstrato através de um grafo. Os nodos do grafo representam locais distintivos ou relevantes do ambiente e os arcos modelam as relações de vizinhança imediata destes locais. Não há, normalmente, informação de distância entre os locais conectados por um arco. E os locais ou nodos do gráfico normalmente não estão associados a uma posição ou a um sistema global de coordenadas. Deste modo, estes mapas são simples e compactos, podem representar locais grandes ou pequenos com eficiência e requerem muito menos espaço de memória que os mapas métricos. Normalmente os mapas topológicos não são considerados úteis para tarefas que exigem grande exatidão no posicionamento de agentes robóticos, como em tarefas de desvio de obstáculos. Entretanto, estes mapas podem ser de grande utilidade em tarefas mais abrangentes como o planejamento de rotas.

Existe, ainda, um terceiro paradigma de mapeamento chamado de híbrido. O mapeamento híbrido, na realidade é uma tentativa de combinar mapas métricos e topológicos de modo a obter o melhor dos dois mundos (GARCIA-FIDALGO; ORTIZ, 2015).

Nesta tese é apresentado um sistema capaz de realizar mapeamentos topológicos baseado apenas em visão monocular.

## 3.3 Tipos de Navegação

Os tipos de navegação autônoma podem ser divididos em três classes básicas de acordo com o tipo de sistema de mapeamento e localização utilizado, a saber:

1. *Map Based Navigation*: Neste tipo de navegação (DESOUZA; KAK, 2002) o agente recebe previamente um modelo do ambiente no qual irá se movimentar. O modelo pode variar de acordo com a quantidade de detalhes utilizada para a descrição do ambiente, variando desde um desenho em CAD do ambiente até um grafo de interelações ou conexões dos elementos em um ambiente.

2. *Map Building Based Navigation*: Neste abordagem (BONIN-FONT; ORTIZ; OLIVER, 2008) o agente primeiramente realiza um mapeamento do ambiente utilizando o seu

próprio sistema sensorial para produzir mapas 2D ou 3D do ambiente para posterior utilização durante o processo de planejamento de rotas.

3. *Mapless Navigation*: Nesta abordagem (BONIN-FONT; ORTIZ; OLIVER, 2008) todo o processo de navegação, incluindo o planejamento de rotas, é inicializado sem que o agente possua nenhum tipo de informação prévia sobre o ambiente. A movimentação do agente é realizada analisando o ambiente para evitar colisões e detectar características relevantes de um percurso de modo a não repetir o mesmo percurso em ações futuras. Neste tipo de navegação, o agente não necessariamente precisa ter informações sobre posições absolutas de si mesmo e dos objetos do ambiente.

As duas primeiras abordagens de navegação, baseadas em mapas e baseadas em construção de mapas, já apresentam soluções bem desenvolvidas e descritas pela literatura, entretanto a navegação sem mapas ainda é um abordagem de estudo aberto, contando com poucos trabalhos. Esta tese não tem como objetivo aborda um sistema de navegação completo, mas apresenta um sistema capaz de realizar localização e construir mapas topológicos utilizando apenas visão monocular que pode ser aplicado nos três tipos de sistemas de navegação descritos acima.

## 3.4 Planejamento de Rota e Planejadores

Planejamento de rota é a ação de, utilizando um mapa como base, construir um plano ou sequencia de ações que um agente robótico deve realizar para alcançar o seu objetivo. O planejador é o subsistema que é responsável por essa ação em um sistema de navegação (CHOSET et al., 2005). Planejadores podem ser desenvolvidos para utilizar mapa métricos, topológicos ou híbridos de mapas métricos e topológicos. Entretanto, normalmente o planejamento de rotas utilizando mapas topológicos é mais rápido computacionalmente (GARCIA-FIDALGO; ORTIZ, 2015).

## 3.5 Simultaneous Localization and Mapping (SLAM)

O problema de mapeamento e localização simultâneos ou *SLAM* consiste em construir um mapa de um ambiente no qual um agente está inserido utilizando apenas informações sensoriais fornecidas pelo próprio aparato sensorial do agente e determinar a posição atual do agente no mapa construído. Uma vez que o mapa tenha sido construído, o agente pode realizar um planejamento de como se deslocar no ambiente mapeado. O mapeamento e a identificação da localização do agente devem ser realizados de modo iterativo e a medida que o agente se desloca para explorar novos espaços ainda não mapeados. O agente ao se deslocar pelo ambiente busca, através do seu aparato sensorial, pontos característicos da cena atual que possam ser utilizados futuramente como pontos de referência. A cada deslocamento, o agente busca, além de novos

pontos de referência, identificar pontos já previamente mapeados e utilizar a nova posição para minimizar o erro na localização dos pontos anteriores. Assim, a medida que o agente se desloca pelo ambiente e obtém mais informações a incerteza associada a cada ponto do mapa é reduzida.

Na Figura 3.2 há uma ilustração de um agente (triângulo azul) realizando SLAM ao longo de uma trajetória $P_1$. O agente precisa sempre identificar pelo menos um *landmark* entre dois deslocamentos consecutivos para que possa reduzir as incertezas sobre a sua localização no ambiente. Além disso, a cada deslocamento é necessário a identificação de novos *landmarks* que permitam a continuidade do processo de localização ao longo da trajetória.

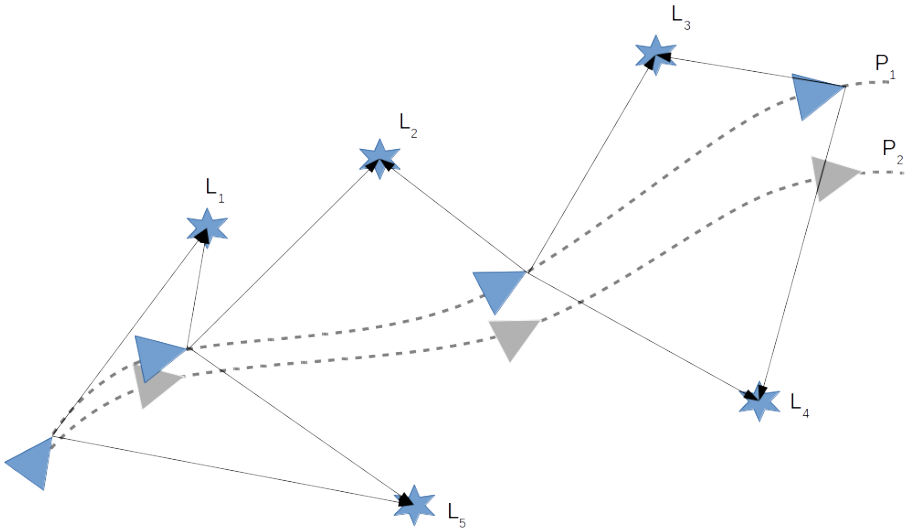


**Figure 3.2:** Ilustração de um agente realizando SLAM e corrigindo a informação da sua trajetória real ($P_1$) em relação a obtida por odometria ($P_2$) baseado nas posições dos *landmarks* $L_1$ a $L_5$.

O problema de SLAM pode ser interpretado como um problema de filtragem de um sinal ruidoso, onde o sinal filtrado é a posição espacial do agente ao longo do tempo. A primeira modelagem como um problema de filtragem foi realizada por SMITH; CHEESEMAN (1986) utilizando filtro de Kalman extendido (EKF). Este filtro é uma versão não-linear do filtro de Kalman clássico. Entretanto, devido a problemas de desempenho, somente após os trabalhos de MONTEMERLO et al. (2002) a abordagem com o EKF teve o seu uso mais difundido.

## 3.6 SLAM Visual

As primeiras abordagens que realizavam SLAM utilizavam sensores de varredura lasers. Estes sensores capturam relações métricas de distância no ambiente e, quando utilizados para fazer SLAM, possibilitam a criação de mapas métricos tridimensionais dos ambientes investigados pelo agente. Entretanto, estes sensores apresentam as desvantagens de possuir custo elevado, emitir radiação lasers, alto peso e consumo elevado de energia. Abordagens sensoriais baseadas

em câmeras são normalmente mais baratas e podem fornecer, também, mapas tridimensionais do ambiente (BONIN-FONT; ORTIZ; OLIVER, 2008). O conjunto de técnicas de processamento de imagens e visão computacional que possibilitam a construção de mapas e a localização de um agente via imagens é denominada SLAM visual (CABALLERO et al., 2009; STRASDAT; MONTIEL; DAVISON, 2010, 2012; FUENTES-PACHECO; RUIZ-ASCENCIO; MANUEL RENDON-MANCHA, 2015).

O SLAM visual, pode ser realizado baseado em técnicas de primitivas geométricas ou em aparência e textura. A abordagem utilizando primitivas geométricas necessita de um grande conhecimento sobre os parâmetros ópticos da câmera, apresenta normalmente um alto custo computacional e produz mapas métricos semelhantes ao de um sensor lidar 3D. Já as abordagens baseadas em aparência e textura, são mais rápidas e não necessitam de conhecimento preciso dos parâmetros da câmera, entretanto, não são capazes de produzir representações precisas de um ambiente (ROMERO et al., 2014).

## 3.7 Detecção Visual de Circuito Fechado

A detecção de circuitos fechados é um problema que está relacionado à capacidade de um agente autônomo realizar exploração e mapeamento em um ambiente desconhecido (REBAI; AZOUAOUI; ACHOUR, 2013). O problema de detecção de circuitos fechados pode ser enunciado como: dado duas sequências temporais de observações visuais de um ambiente, um vídeo por exemplo, determinar se são originárias de um mesmo local ou não (CUMMINS; NEWMAN, 2007). CUMMINS (2009), considera que este problema é equivalente ao problema de SLAM Visual em um espaço de aparência, onde o principal objetivo é estimar o local onde o agente se encontra e não a sua posição em um sistema de coordenadas.

O problema é aparentemente simples e as abordagens propostas para tratar tal problema (CUMMINS; NEWMAN, 2007, 2008a, 2011; GLOVER et al., 2012; REBAI; AZOUAOUI; ACHOUR, 2013; GUERRA; MUNGUIA; GRAU, 2014) aparentam ser de um sistema de busca por imagens. Entretanto, diferentemente de um sistema de busca por imagens, onde o objetivo da busca é retornar uma imagem com um conteúdo semelhante ao da imagem de busca, os sistemas de detecção de trajetórias fechadas tem por objetivo determinar se dois conjuntos de imagens pertencem ao mesmo local em uma determinada trajetória. Os maiores desafios destes sistemas são conseguir identificar características nas sequências de observações que identifiquem um local unicamente e que sejam robustas a possíveis variações de posição de objetos nas observações ou a variações de luminosidade. Por exemplo, observações de duas ruas distintas podem apresentar componentes idênticas, como sinalizações de trânsito, enquanto que observações de uma mesma rua podem variar significativamente dependendo da hora do dia, da disposição de pessoas e carros. Outra diferença, estes sistemas podem utilizar a informação sobre o histórico de observações no processo de identificação do local, enquanto que sistemas de busca por imagem normalmente não trabalham com informações temporais ou histórico de buscas (SIEGWART; NOURBAKHSH;

SCARAMUZZA, 2011).

A detecção de trajetórias fechadas é importante no processo de exploração autônoma e mapeamento de um ambiente desconhecido por ser necessário, essencialmente, cobrir todo o espaço de um ambiente investigado em um tempo finito. Para realizar tal tarefa de modo eficiente o agente precisa minimizar a quantidade de percursos já navegados no ambiente e evitar realizar circuitos fechados, caso contrário o agente estará gastando tempo e energia para realizar um novo mapeamento de uma área já mapeada. Assim, na exploração é importante que o agente perceba quando está preso em um circuito fechado, revisitando locais já explorados, para que possa tentar evitar a mesma rota em ações futuras. A habilidade de detectar circuitos fechados também é importante na navegação, já que é preciso comparar os locais visitados com as características visuais do local de destino para saber quando o agente robótico completou a sua trajetória.

A detecção visual de circuitos fechados também pode ser utilizada em conjunto com métodos de SLAM tradicionais para reduzir o erro de integração dos mapas métricos produzidos. Ao se detectar visualmente que o agente já passou por um determinado local, as informações geométricas deste local armazenadas previamente podem ser reutilizadas e combinadas com as informações geométricas atuais para produzir um mapeamento métrico mais preciso.

## 3.8 Definição Formal do Problema Tratado pela Tese

Navegação robótica envolve diferentes áreas de conhecimento: inteligencia computacional, controle, visão computacional, simulação numérica, instrumentação, entre outras. E em cada uma delas há uma diversidade própria de problemas relacionados à navegação robótica. Nesta tese, são considerados dois problemas fundamentais na área de navegação robótica: Mapeamento e Localização. Para limitar o escopo, as seguintes definições dos problemas são adotadas:

- Problema 1: Navegação Autônoma - O agente autônomo $f_a$, conhecendo apenas uma sequência de estímulos de entradas locais $I_j$ , previamente memorizada, de uma trajetória entre dois pontos $p_1$ e $p_2$ deve ser capaz de atingir $p_2$ uma vez posicionado em uma vizinhança $r$ de $p_1$.

- Problema 2: Exploração Autônoma - Considerando-se um agente robótico autônomo $f_a$, inserido em um ambiente inicialmente desconhecido $A$. Tal agente deve explorar seu ambiente de tal modo a cobrir todo o ambiente virtual em um número de passos menor que $S_{max}$.

- Problema 3: Mapeamento e localização simultâneos - O agente autônomo $f_a$, recebendo uma sequência de estímulos visuais locais $I_j$ , deve ser capaz de compor um mapa que permita o agente identificar a sua própria localização ao passar por um ponto $p_1$ previamente memorizado.

Esta tese se destina a resolver o **problema 3** através de um sistema de localização visual e mapeamento topológico com uma abordagem semelhante à utilizada para detecção visual de trajetórias fechadas ou *loop closure detection* (CUMMINS; NEWMAN, 2007, 2008a, 2011; GUERRA; MUNGUIA; GRAU, 2014).

O sistema deve possuir a habilidade de identificar nas observações visuais características especiais que permitam identificar o local onde elas foram realizadas.

## 3.9 Métricas de Avaliação e Benchmarks

### 3.9.1 Métricas de Avaliação

Um sistema de navegação realiza diversas tarefas que são necessárias para subsidiar ação de navegação propriamente dita, tarefas como: detectar e evitar colisões, realizar odometria, detecção de caminhos fechados, planejamento de rotas e SLAM. Cada uma destas tarefas apresenta um conjunto próprio de especificidades.

Sistemas de detecção de obstáculos visam garantir o deslocamento do agente de modo seguro ao longo de uma determinada rota, realizando apenas pequenos desvios em relação à trajetória original da rota. Utilizam critérios de desempenho como: número de colisões por distância percorrida, número de objetos possíveis de colisão, distância média, mínima e máxima para os obstáculos em uma determinada rota (CHOSET et al., 2005).

Já os sistemas de odometria são responsáveis pela localização atual de um agente em relação ponto de referência. Existem sistemas de odometria que podem medir distâncias baseadas em imagens de vídeo, em acelerômetros ou contadores de giro. A técnica para a comparação de métodos de odometria é comparar a trajetória registrada pelo odômetro com uma trajetória de referência usando uma métrica de distância, normalmente distância euclidiana. Em sistemas em ambiente externo (*outdoor*) a trajetória de referência é usualmente tomada como sendo a produzida por um sistema de GPS diferencial de precisão que é capaz de produzir trajetórias com erro menor que 10mm em relação a um referencial geográfico (um prédio, montanha, etc).

Sistemas de detecção de circuitos fechados são necessários para que um sistema de navegação identifique se está preso em um percurso fechado ou se já visitou um determinado sítio em um instante de tempo próximo. Essencialmente um sistema de detecção de circuitos fechados é um detector de padrões que identifica a localização atual do agente em uma trajetória já memorizada (CUMMINS; NEWMAN, 2011). O métodos de comparação utilizados para estes sistemas são métricas clássicas de avaliação de sistemas de reconhecimento de padrão como curvas Precison-Recall e ROC e/ou matriz de confusão.

O sistema de planejamento de rota é um método que produz rotas em um determinado mapa que minimiza um determinado critério que pode ser de: distância mínima entre dois pontos, menor tempo de viagem, menor energia. O sistema de planejamento de rota é a tarefa de comparação mais sensível que há em um sistema de navegação (CHOSET et al., 2005).

Isso porque, essencialmente, métodos de planejamento só podem ser comparados com precisão utilizando ambientes simulados que sejam capazes de reproduzir com fidelidade as condições do ambiente, do mapa e do agente que o sistema de navegação irá utilizar. A comparação de métodos de planejamento é realizada através de sucessivas iterações em ambientes simulados e da comparação dos seus resultados finais, em relação à distância percorrida pelo agente, velocidade média, máxima, mínima, distância média, máxima, mínima em relação aos obstáculos, suavidade das curvaturas da rota.

O SLAM é a tarefa de, simultaneamente, se localizar (o agente) e mapear um ambiente desconhecido. Esta tarefa exige normalmente a interação de dados de distância percorrida, com os dados fornecidos por odômetros e de reconhecimentos de *landmarks*, adquiridos por câmeras e radares. Usualmente a comparação é realizada através da diferença de distância euclidiana entre um mapa métrico de referência e o mapa métrico produzido pelo sistema de SLAM.

### 3.9.2 Proposta de Avaliação do Sistema

Resumidamente, o objetivos desta tese foi propor um sistema de localização visual e mapeamento topológico utilizando um novo tipo de mapa auto-organizável dinâmico com características úteis para a área de navegação. Desta forma, pode-se dividir o trabalho desta tese em duas partes essenciais: um novo mapa auto-organizável dinâmico com características úteis para a área de navegação e um sistema de localização visual e mapeamento topológico. O mapa proposto, FSS-SOM (Seção 4.2), foi projetado visando atender a algumas características desejáveis no sistema de mapeamento e localização, mas pode ser utilizado em problemas de agrupamento gerais. Assim, a avaliação do desempenho do mapa FSS-SOM como um agrupador foi realizada de modo separado do sistema completo de mapeamento topológico e localização visual (Seção 4.3).

A avaliação do mapa e do sistema completo estão apresentadas no Capítulo 5. Nos testes realizados com o mapa foram utilizadas nuvens de pontos proveniente de séries temporais como a de Mackey-Glass (MACKEY; GLASS et al., 1977) e a SOI (AUSTRALIAN GOVERNMENT, 2014a) para viabilizar a comparação de desempenho do FSS-SOM com outros modelos de Mapas Auto-organizáveis.

Os testes do sistema de mapeamento e localização foram conduzidos de duas formas: uma testando a capacidade de localização visual e uma outra considerando a localização e a capacidade de produzir mapas.

Nos testes da capacidade de localização visual, o principal objetivo foi determinar se o sistema consegue distinguir se um determinado local já foi visitado anteriormente ou não. Foram utilizadas sequências de observações realizadas em ambiente interno do CIN, utilizando uma plataforma robótica desenvolvida para esta finalidade (Apêndice A), em ambiente externo no canteiro central da UFPE e em dois benchmarks utilizados em SLAM visual e em detecção visual de circuitos fechados por CUMMINS (2009). As sequências de observações dos testes realizados

dentro do CIN e na UFPE não estão georreferenciados e, por isso, não apresentam um *ground truth* métrico para a comparação com outros sistemas, sendo apresentados apenas as taxas de erro no reconhecimento da localização de trechos do percurso selecionados manualmente. Nos benchmarks (CUMMINS, 2009) as informações de georreferenciamento fornecidas foram utilizadas para traçar curvas tipo Precision-Recall para a avaliação e comparação do desempenho do sistema com outras abordagens.

Nos testes do sistema considerando a capacidade de localização visual e de geração automática de mapas topológicos foram utilizadas as sequências de imagens do percurso em ambiente externo na UFPE e os benchmarks de SLAM Visual (CUMMINS, 2009). O principal objetivos destes testes foi determinar se o sistema é realmente capaz de produzir mapas topológicos coerentes utilizando unicamente informação visual monocular. Os mapas topológicos produzidos pelo sistema foram comparados com as topologias dos percursos de aquisição das imagens monoculares utilizadas nos testes.

# 4

# Proposta de Solução

Neste capítulo são apresentados os modelos propostos de Mapa Auto-organizável FSS-SOM (seção 4.2) e o sistema de localização visual e mapeamento topológico 4.3. A subseção 4.2.1, apresenta o pseudocódigo do FSS-SOM.

## 4.1 Introdução

Mapas Auto-organizáveis (SOMs) são redes neurais artificiais que trabalham realizando redução de dimensionalidade de conjuntos de dados utilizando aprendizagem não-supervisionada em um processo que tenta preservar as relações topológicas originais dos dados de entrada (HAYKIN, 1999).

A primeira abordagem de Mapas Auto-Organizáveis (SOM) foi proposta por KOHONEN (1990), consistia em um grid com topologia e número de nodos (neurônios) fixos e determinados previamente. Este tipo de rede tem sido utilizada com sucesso para realizar agrupamento (*Clustering*) de dados e categorização de padrões ruidosos (KOHONEN, 1990). Baseado no SOM original, numerosas variações foram propostas para permitir o uso de Mapas Auto-organizáveis em problemas sem determinação *a priori* da topologia ou do número de nodos do mapa (ver Capítulo 2). Estudos conduzidor por Martinetz usando o "Neuro Gas" (NG) (1993), Fritzke usando o Growing Neural Gas (GNG) (1995) e Araujo e Costa com o LARFSOM (2009) produziram mapas que podem se expandir em número de nodos e se adaptar a formas topológicas arbitrárias dependendo do conjunto de treinamento apresentado. Estes tipos de mapas são chamados de *Self-organizing Maps with Time-varying Structure* (ARAUJO; REGO, 2013). Existe também uma outra classe de variações do SOM que possuem características especializadas para tratar padrões como séries temporais como o RecSOM (VOEGTLIN, 2002) e o MSOM (STRICKERT; HAMMER, 2005). Estes mapas podem armazenar informações sobre uma sequência de apresentação de padrões.

O SOM e suas variantes são utilizados normalmente para realizar agrupamento com aprendizagem não-supervisionada, mas podem, utilizando técnicas de quantização vetorial (*Vector Quantization*), serem utilizados em abordagens tipicamente de aprendizagem supervisionada

como em previsão de séries temporais (KOHONEN, 1990; MARTINETZ T., 1993; CHERIF; CARDOT; BONé, 2011). Um tipo de quantização vetorial pode ser implementado utilizando-se codebooks[1]. Neste processo, os Mapas Auto-organizáveis são utilizados primeiramente para detectar as componentes mais importantes do conjunto de dados em uma etapa de agrupamento normal. Após esta etapa, os nodos dos mapas são rotulados com $n$ valores discretos e uma etapa supervisionada se inicia realizando associações dos nodos ativados com um determinado padrão de entrada e o correspondente valor de saída $\mathbf{y}$. A forma com que essa associação é realizada varia de acordo com a abordagem utilizada.

Os algoritmos atuais de navegação robótica e de SLAM (visual, em aparência ou não) utilizam nas suas abordagens, em grande parte, ferramentas clássicas de controle e filtro de séries temporais como filtros de Kalman e filtros de Kalman estendidos (EKF) (THRUN. SEBASTIAN BURGARD. WOLFRAM, 2006). Sendo este último, uma versão não-linear do filtro clássico de Kalman. Estes filtros são utilizados nos algoritmos de SLAM para compor uma informação com incerteza reduzida sobre a posição atual do agente autônomo a partir de sequências de observações imprecisas fornecidas por um ou mais sensores. Os principais problemas envolvendo esta abordagem é que apresenta forte dependência paramétrica do filtro, implementação não trivial e complexidade computacional, no melhor caso, de $O(n \cdot log(n))$ (MONTEMERLO et al., 2002) e no pior de $O(n^2)$, onde $n$ é o número de padrões fornecidos por cada sensor ao longo do tempo.

Mapas Auto-organizáveis podem ser, também, utilizados como filtros para sequência de observações, de modo semelhante aos de Kalman, com a vantagem de apresentarem uma implementação mais simples e uma quantidade menor de parâmetros (MARTINETZ T., 1993; CHERIF; CARDOT; BONé, 2011). Neste capítulo é apresentada na Seção 4.1 um novo modelo de Mapa Auto-organizável baseado em características do GNG (FRITZKE, 1995) e do LARFSOM (ARAúJO; COSTA, 2009). O desenvolvimento deste modelo surgiu como um alternativa neural às abordagens clássicas utilizando filtros para tratar problemas envolvendo séries temporais de observações no contexto de navegação autônoma. A abordagem proposta baseada em redes neurais visa ser rápida, não linear e de fácil implementação. Nas Seções 5.1 e 5.2, são apresentadas a metodologia experimental e os resultados comparativos usando o mapa proposto e outros tipos de Mapas Auto-organizáveis.

## 4.2 O Mapa Auto-Organizável de Cobertura Rápida de Espaço de Estados

O modelo proposto de mapa de cobertura rápida de espaço de estados n-dimensionais, (*Fast Coverage of an N-dimensional State Space Self-Organizing Map*, FSS-SOM) é um SOM

[1] Um codebook é um tipo de tabela que associa $n$ valores discretos (rótulos) com saídas $\mathbf{y}$. Ver (MARTINETZ T., 1993)

com estrutura variante no tempo (SOM-TVS) (ARAUJO; REGO, 2013) e foi idealizado para ser um mapa de uso geral, mas com características especiais para a implementação de filtros, previsores não-lineares ou para reduzir a dimensionalidade de dados. Estas características, são especialmente úteis para tratar problemas de SLAM Visual em Espaço de Aparências e navegação robótica. O mapa proposto realiza processamento e ajuste rápido de pesos e fornece um mecanismo de controle de densidade de nodos. O controle de densidade de nodos é uma característica especialmente útil quando se trabalha com conjuntos de dados com redundâncias. A concepção do mapa é inspirada no princípio de funcionamento de outros modelos de mapa como a estrutura dinâmica do GNG (FRITZKE, 1995) e o campo receptivo ajustável presente no LARFSOM (ARAúJO; COSTA, 2009).

O FSS-SOM apresenta topologia e número de nodos variáveis, processamento rápido e malhas de nodos com comprimento de conexões limitadas entre valores máximos e mínimos. O FSS-SOM, também não utiliza aprendizagem Hebbiana com decaimento exponencial, como presente em muitas versões de SOMs, não remove nodos inseridos, adapta os nodos do mapa com uma taxa de aprendizagem constante ao longo do tempo. No geral, estas características permitem que o mapa cresça sempre que necessário, sem realizar operações matemáticas custosas e com possibilidade de controle do tamanho máximo do mapa através de limitação do número de nodos do mapa (parâmetro do modelo $N_{max}$) ou da variação de comprimento de suas arestas (parâmetros do modelo: $d_0$, $d_1$, $d_2$).

O modelo proposto utiliza três princípios básicos, dispostos abaixo, que norteiam o seu algoritmo e modo de funcionamento.

1. Se uma amostra não estiver suficientemente perto do nodo mais próximo dela no mapa e este puder crescer, um novo nodo será adicionado ao mapa na posição da amostra em detrimento da realização de qualquer ajuste de pesos dos nodos do mapa.

2. Se uma amostra estiver suficientemente próxima de um nodo do mapa, é entendido que o nodo mais próximo já representa bem a amostra e que não é necessário um ajuste de pesos dos nodos do mapa.

3. Se uma amostra não estiver nem muito longe e nem muito perto do nodo mais próximo, ajusta-se os pesos do nodo mais próximo, e dos seus vizinhos imediatos, de modo proporcional à distância entre o nodo e a amostra.

No FSS-SOM, os parâmetros $d_0$, $d_1$, $N_{max}$ são usados para determinar se uma amostra está suficientemente perto de um nodo vencedor e qual ação deve ser tomada: inserir um novo nodo ou adaptar o nodo vencedor e os seus vizinhos mais próximos.

Os valores de $d_0$ e $d_1$ estabelecem, respectivamente a menor e a maior distância entre uma amostra e um nodo vencedor que resultará em uma adaptação do nodo vencedor e dos seus vizinhos mais próximos. Se uma amostra estiver a uma distância menor que $d_0$ do nodo vencedor nenhuma adaptação ou inserção de nodos é realizada, sendo assim caracterizada como

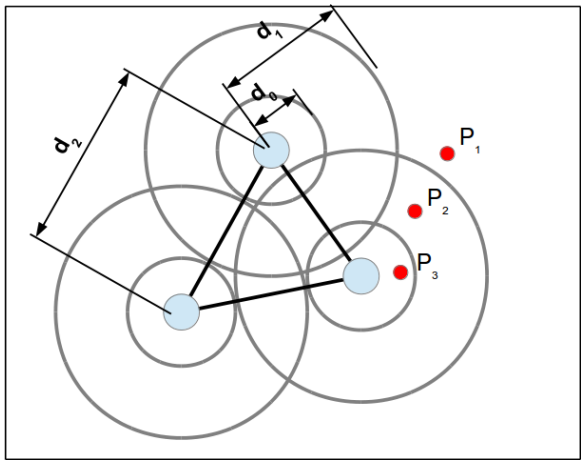


**Figure 4.1:** Ilustração dos parâmetros $d_0$, $d_1$ e $d_2$.

a distância mínima de reação do mapa. Se uma amostra estiver a uma distância maior que $d_0$ e menor que $d_1$ do nodo vencedor, uma adaptação de pesos do nodo vencedor e seus vizinhos mais próximos é realizada. E caso a amostra esteja a uma distância maior que $d_1$ e o mapa ainda apresente um número de nodos menor que $N_{max}$, um novo nodo é inserido no local da amostra. Assim, o número de nodos do mapa poder ser controlado, indiretamente, pelos parâmetros de distância entre os nodos $d_0$ e $d_1$, enquanto $N_{max}$ determina o limite máximo de nodos que o mapa poderá possuir. No caso, do mapa já apresentar o número máximo de nodos permitidos ($N_{max}$), o mapa passa a só realizar adaptações de peso e a remover e inserir conexões.

Adicionalmente a essas premissas, o mapa, também, apresenta o mesmo mecanismo de envelhecimento (*ageing*) de arestas que o utilizado no mapa GNG (FRITZKE, 1995), para penalizar conexões entre nodos que são pouco ativados. Assim, a cada interação do treinamento do mapa as arestas são envelhecidas em uma unidade. Se uma aresta atingir uma idade maior que $age_{max}$ ela será removida. A idade de uma aresta é zerada sempre que o nodo ao qual ela está conectada for selecionado como um nodo vencedor. Para impor um certo grau de regularidade de tamanho às conexões dos mapas produzidos pelo FSS-SOM, dois nodos quaisquer só serão conectados se a distância entre eles for maior que $d_0$ e menor que $d_2$. Assim, indiretamente, utilizando $d_0$ e $d_2$, é possível controlar a regularidade na formação das conexões dos nodos, já que estes parâmetros oferecem um mecanismo de limitar os tamanhos máximos e mínimos.

A Figura 4.1 ilustra as regras de inserção de nodos e conexões do mapa FSS-SOM. A amostra $P_3$, por exemplo, dentro do raio $d_0$, não provoca nenhum tipo de atualização no mapa. A amostra $P_2$, produz uma adaptação dos nodos mais próximos. A amostra $P_1$, produz uma inserção de um novo nodo no mapa, caso o número total de nodos seja menor que $N_{max}$. Quaisquer dois nodos do mapa com distâncias menores que $d_2$ e maiores que $d_0$ são conetados por uma aresta.

Controlando os parâmetro $d_0$, $d_1$ e $d_2$ é possível ajustar a granularidade da adaptação do mapa ao conjunto de treinamento. Os parâmetros $d_0$ e $d_1$ podem ser utilizados para controlar a densidade dos nodos criados no treinamento e $d_0$ e $d_2$ podem controlar o grau de conexão dos nodos do mapa.

O desempenho do mapa FSS-SOM depende fortemente de uma escolha adequada para os parâmetros $d_0$, $d_1$ e $d_2$. Uma forma de determinação inicial destes parâmetros que produziu resultados com um MSE[2] baixo em distribuições gaussianas de pontos, foi adotar frações do desvio padrão dos dados de treinamento para cada um destes parâmetros. Nos testes realizados com o FSS-SOM foram adotadas as seguintes taxas em relação ao desvio padrão do conjunto de treinamento: $d_2 = \sigma$, $d_1 = \sigma/10$ e $d_0 = \sigma/100$.

As atualizações Hebianas do nodo vencedor e dos seus vizinhos mais próximos no mapa FSS-SOM são realizadas por taxas de aprendizagem fixas, respectivamente, $e_b$ e $e_n$ (ver Seção 4.2.1). Onde, $e_b$ deve ser maior que $e_n$, da mesma forma adotada no GNG. O pseudocódigo do FSS-SOM está na subseção 4.

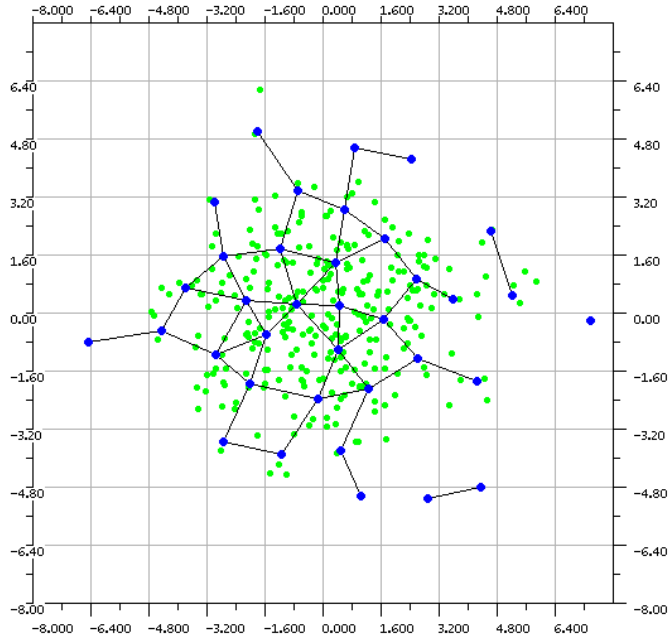


**Figure 4.2:** Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{15}$, $d_1 = \frac{4}{3}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$

A Figura 4.2 ilustra a adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$. O resultado apresentado na figura foi obtido após um treinamento de 200 épocas com os seguintes parâmetros: $d_0 = \frac{2}{15}$, $d_1 = \frac{4}{3}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$. Note-se que o mapa está cobrindo toda a distribuição dos dados de treinamento, que as conexões dos nodos são aproximadamente do mesmo tamanho e que, apesar de $N_{max} = 1000$, o mapa não apresenta mais que algumas dezenas de nodos. Ou seja, as escolhas dos parâmetros $d_0$, $d_1$ e $d_2$ limitaram o crescimento do mapa pela densidade de pontos.

[2]MSE - Mean Square Error

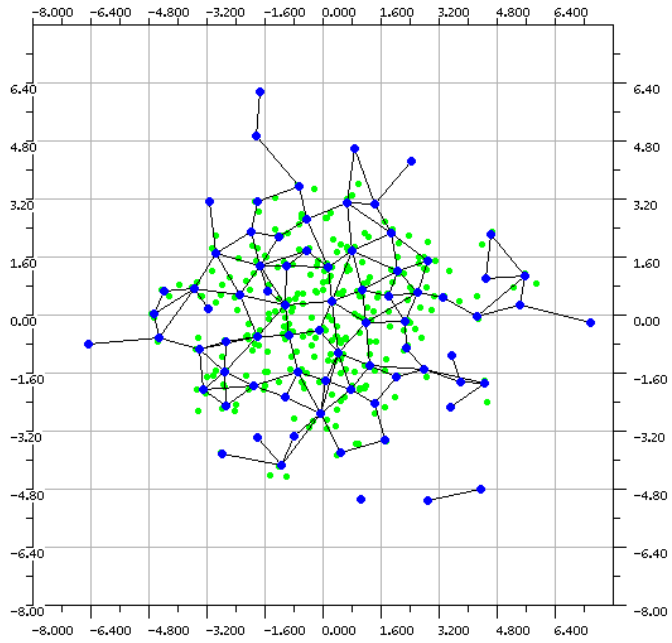


**Figure 4.3:** Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{30}$, $d_1 = \frac{2}{3}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$

A Figura 4.3 ilustra o mapa utilizado os mesmos parâmetros de treinamento do mapa da Figura 4.2 exceto em: $d_0 = \frac{2}{30}$, $d_1 = \frac{2}{3}$. Note-se o aumento na densidade de nodos do mapa sobre os pontos da nuvem gaussiana devido a redução dos parâmetros de distância $d_0$, $d_1$ em relação ao caso apresentado na Figura 4.2.

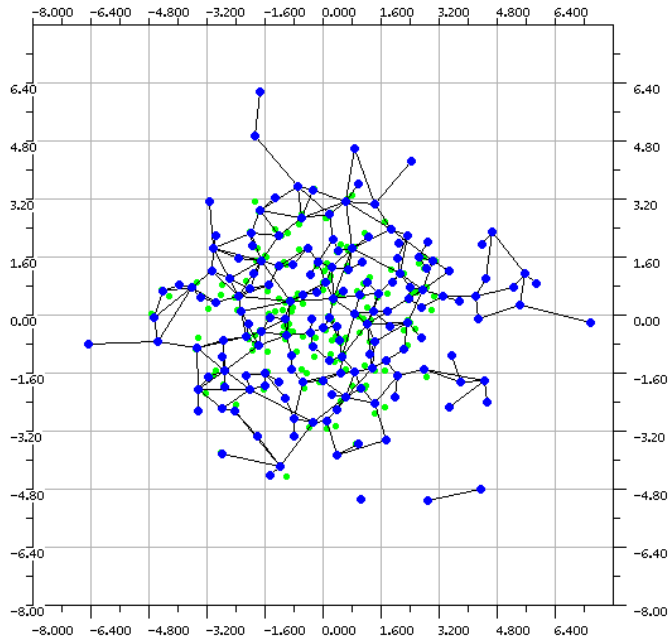


**Figure 4.4:** Adaptação do FSS-SOM a uma distribuição gaussiana de pontos bidimensional com desvio padrão 2 e média zero, $N(0,2)$ e parâmetros: $d_0 = \frac{2}{60}$, $d_1 = \frac{2}{6}$, $d_2 = 2$, $N_{max} = 1000$, $e_b = 0.1$ e $e_n = 0.01$

Na Figura 4.4, o FSS-SOM apresenta as seguintes modificações de parâmetros em relação as figuras anteriores (4.2 e 4.3): $d_0 = \frac{2}{60}$ e $d_1 = \frac{2}{6}$. Novamente, é possível notar o efeito da redução dos parâmetros de distância do mapa sobre a sua topologia. Neste caso, $d_0$ e $d_1$ apresentam 25% dos valores utilizados na Figura 4.2 e 50% dos utilizados na Figura 4.3.

As Figuras 4.2, 4.3 e 4.4 ilustram bem os efeitos sobre a topologia do mapa e o seu crescimento em função dos parâmetros de distância $d_0$ e $d_1$. Apesar de mantido para os três casos um mesmo número máximo de nodos, taxa de aprendizagem e número de épocas, os mapas apresentaram uma capacidade de adaptação ao conjunto de treinamento modulada pelos parâmetros de distância. As Figuras de 4.2 à 4.4 ilustram como o FSS-SOM pode ser ajustado para seguir uma densidade e distribuição de nodos mínima, não menor que a densidade do conjunto de treinamento. Essa é uma característica muito importante quando se está trabalhando com conjuntos de dados de treinamento muito redundantes e ruidosos e ajuda a evitar uma alocação de nodos acima do necessário para a realização de uma determinada tarefa. Este tipo de conjunto de dados, com informações redundantes, ruidosas e não uniformes é relativamente comum em nuvens de pontos produzidas por sensores de varredura como LIDARES, RADARES e SONARES. A aquisição de dados nestes sensores se dá, primariamente, de forma radial em grande ângulo de abertura. Entretanto, os mapas utilizados pelos sistemas de navegação exigem um sistema cartesiano e quando se transformam os dados provenientes destes sensores em nuvens de pontos produzimos não uniformidades na distribuição dos pontos. Na Figura 4.5 há uma

ilustração gráfica do problema.

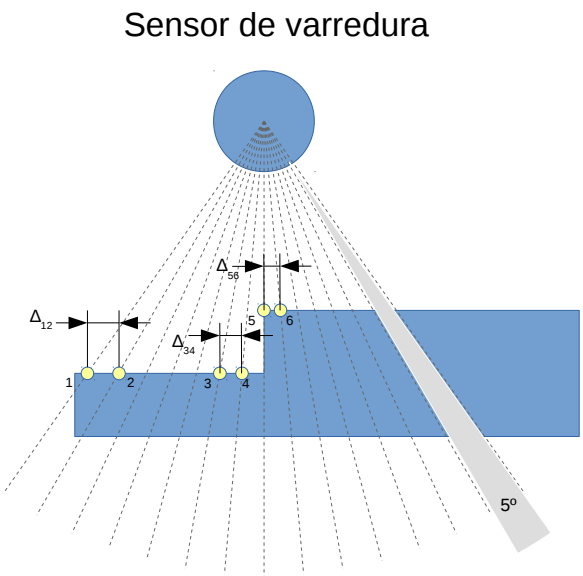


**Figure 4.5:** Ilustração da aquisição de pontos com espaçamento cartesiano não uniforme de um sensor de varredura tipico.

Sensores de varredura típicos utilizados na área de navegação autônoma como LIDAR, RADAR ou SONAR realizam aquisições de pontos em passos angulares regulares. No exemplo ilustrativo da Figura 4.5 é apresentado uma aquisição com passos de 5°. Entretanto, na maioria dos sistemas a representação interna do sistema de coordenadas é do tipo cartesiana. Assim, ao se projetarem esses pontos para um sistema cartesiano, a distribuição dos pontos se torna não uniforme. Mesmo se for analisada uma superfície completamente plana e perpendicular ao sensor, como no caso dos intervalos $\Delta_{12}$,$\Delta_{34}$ e $\Delta_{56}$ da Figura 4.5. Utilizando-se o mapa proposto, como um interpolador para as nuvens de pontos dos sensores de varredura mencionados acima, e ajustando-se os parâmetros $d_0$ e $d_1$ pode-se , por exemplo, limitar as não uniformidades a um valor fixo e reduzir a redundância dos dados facilitando processamentos subsequentes.

### 4.2.1 Pseudocódigo do FSS-SOM

A seguir é apresentado o pseudocódigo do FSS-SOM, Algoritmo 4. O Mapa FSS-SOM apresenta sete parâmetros básico: $N_{max}$ é a quantidade máxima de nodos que o mapa pode utilizar; $age_{max}$ é um parâmetro de envelhecimento do nodo; $d_0$, $d_1$ e $d_2$ são parâmetros de vizinhança que determinam quando um nodo pode ser atualizado ou inserido; e $e_b$,$e_n$ são, respectivamente, as taxas de atualização do nodo vencedor e dos vizinhos do nodo vencedor. O vetor $\xi$ representa uma amostra escolhida aleatoriamente do conjunto de treinamento. $\mathbf{w_{s_1}}$ é o vetor de pesos do mapa com menor distância euclidiana em relação a amostra $\xi$ e $\mathbf{w_{s_2}}$ é o segundo vetor mais

próximo.

**Algorithm 4** FSS-SOM

1: Inicialize parâmetros: $N_{max}$,$age_{max}$,$d_0$,$d_1$,$d_2$,$e_b$,$e_n$;
2: Inicialize o mapa selecionando aleatoriamente duas amostras retiradas do conjunto de treinamento espaçadas por uma distância maior que $3 * d_0$ para iniciar o mapa;
3: **repeat**
4: Selecione uma amostra ($\xi$) do conjunto de treinamento e encontra os dois nodos mais próximos: $\mathbf{w_{s_1}}$(o mais próximo) **and** $\mathbf{w_{s_2}}$(o segundo mais próximo);
5: Calcule a distância: $d_{12} = \|\mathbf{w_{s_1}} - \mathbf{w_{s_2}}\|^2$;
6: **if** ( $\mathbf{w_{s_1}}$ e $\mathbf{w_{s_2}}$ não estão conectados) **and** ($d_{12} < d_2$ ) **and** ($d_{12} > d_0$ ) **then**
7: Conecte $\mathbf{w_{s_1}}$ e $\mathbf{w_{s_2}}$ com uma aresta com age=0
8: **else if** $\mathbf{w_{s_1}}$ **and** $\mathbf{w_{s_2}}$ já estão conectados **then**
9: reinicie o age da aresta para zero;
10: **end if**
11: Calcule a distância: $d = \|\xi - \mathbf{w_{s_1}}\|^2$;
12: **if** ($d > d_1$) **and** ((Número de nodos do mapa) $< N_{max}$) **then**
13: Insira um novo nodo no mapa com vetor de peso igual ao da amostra $\xi$;
14: **if** $d < d_2$ **then**
15: Conecte $\mathbf{w_{s_1}}$ e o novo nodo;
16: **end if**
17: **else**
18: **if** $d > d_0$ **then**
19: Adapte a posição do nodo($\mathbf{w_{s_1}}$):
20: $\mathbf{w_{s_1}} = \mathbf{w_{s_1}} + e_b(\xi - \mathbf{w_{s_1}})$;
21: Adapte a posição dos nodos vizinhos($w_{nb}$):
22: $\mathbf{w}_{nb} = \mathbf{w}_{nb} + e_n(\xi - \mathbf{w}_{nb})$;
23: Incremente o age das arestas do nodo vencedor($\mathbf{w_{s_1}}$);
24: **end if**
25: **end if**
26: Remova todas as arestas com age maior que $age_{max}$
27: Remova todas as arestas maiores que $d_2$;
28: Remova todas as arestas menores que $d_0$;
29: **until** Critério de parada.

No Algoritmo 4, as linhas de 5 a 10 verificam se os vetores de peso $\mathbf{w_{s_1}}$ e $\mathbf{w_{s_2}}$ estão ou não conectados. Se houver conexão, verifica se atende as restrições de granularidade ($d_0$ e $d_2$) de conexão do mapa. O código entre as linhas 12 e 16 determina se a amostra $\xi$ esta suficientemente distante do vetor de pesos $\mathbf{w_{s_1}}$ para que seja inserido um novo nodo no mapa e se este novo nodo pode ou não ser conectado a $\mathbf{w_{s_1}}$ sem violar as restrições de granularidade do mapa. O código entre as linhas 18 e 24, determina se deve ou não haver adaptação dos pesos do nodo $s_1$ e de seus vizinhos. O nodo $s_1$ é o com vetor de pesos $\mathbf{w_{s_1}}$, mais próximo da amostra $\xi$.

## 4.3 Sistema de Localização Visual e Mapeamento Topológico

A proposta do sistema de localização visual e mapeamento topológico apresentada nesta tese é inspirada no FABMAP (CUMMINS; NEWMAN, 2008a), em sistemas de localização visual utilizando *bag of words* (FILLIAT, 2007) e em sistemas de SLAM Visual. Entretanto, o sistema proposto se baseia principalmente em um abordagem Hebbiana com Mapas Auto-Organizáveis.

O sistema proposto de localização visual e mapeamento topológico é, essencialmente, um sistema de detecção de circuitos fechados baseado em imagens monoculares capaz de construir mapas topológicos. Para isso, o sistema apresenta dois tipos básicos de estruturas. Uma capaz de armazenar e identificar características visuais salientes em uma imagem, que posteriormente são utilizadas para realizar uma codificação dos locais já visitados pelo sistema. No sistema, esta estrutura é denominada de I-MAP. Este nome é um acrônimo para *Image Map* ou "mapa de imagens" em referência ao fato de ser utilizada para armazenar características visuais de um local. Uma segunda estrutura, denominada de P-MAP é utilizada para armazenar os locais já visitados pelo agente. O P-MAP tem a finalidade não só de armazenar as codificações dos locais já visitados, como também a relação de vizinhança destes locais. O nome P-MAP é um acrônimo para *Place-Map* ou "mapa de locais" em referência ao tipo de informação que representa.

As informações visuais contidas no I-MAP são utilizadas para atualizar as informações de localização do P-MAP e as informações de localização e vizinhança do P-MAP são utilizadas para limitar a busca por nodos vencedores no I-MAP e acelerar o processo de identificação do próximo local visitado. O sistema de localização visual e mapeamento topológico proposto pode ser modelado como uma função $f$ que utiliza como parâmetros de entrada um I-MAP, um P-MAP, uma imagem teste($\mathbf{I}$) e um rótulo de um nodo do P-MAP($\mathbf{nodo}_{i-1}$), Equação 4.1.

$$\mathbf{nodo}_i = f(I-MAP, P-MAP, \mathbf{I}, \mathbf{nodo}_{i-1}). \tag{4.1}$$

O parâmetro de retorno do sistema é o nodo do P-MAP que representa o local atual que o agente está no P-MAP, $\mathbf{nodo}_i$. Assim, o principal objetivo das informações visuais observadas é servir de referência fixa para o P-MAP que irá construir um mapa topológico, onde cada nodo representará um local de um determinado ambiente e indicar onde o agente está neste mapa topológico.

Na Figura 4.6, há uma representação geral do funcionamento do sistema de localização visual e mapeamento topológico. O sistema realiza aquisição de imagem, extrai as características visuais do local recém adquirido, codifica a imagem utilizando o I-MAP e passa a codificação para o P-MAP.

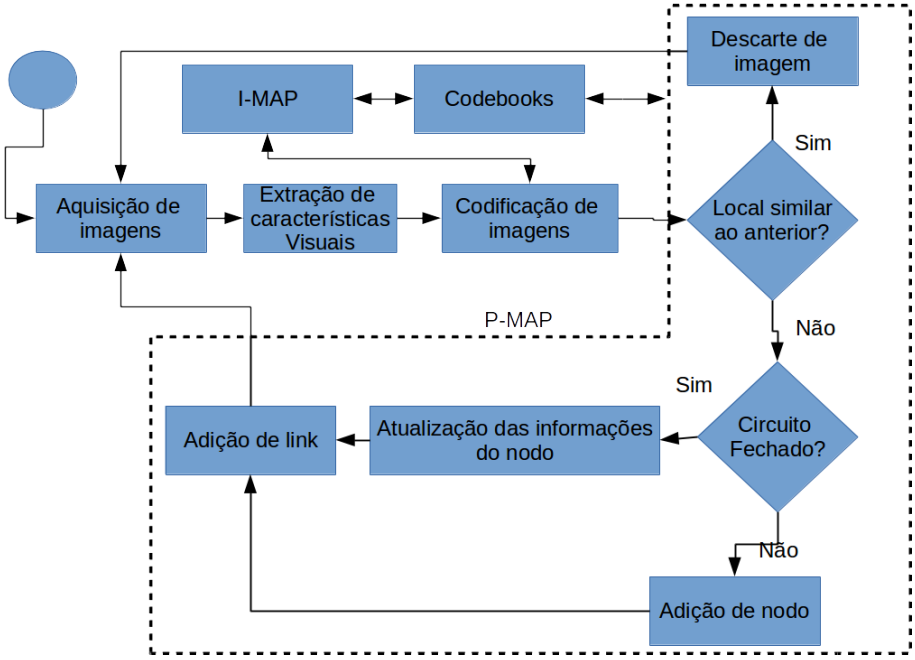


**Figure 4.6:** Diagrama de blocos do sistema de localização visual e mapeamento topológico proposto

O principal problema tratado pelo sistema é determinar se uma nova imagem adquirida corresponde a um novo nodo no P-MAP ou a um nodo já existente, como ilustrado na Figura 4.7. Para construir um mapa topológico coerente com os locais visitados pelo agente, o sistema deve ser capaz de identificar nas sequencias de imagens monoculares características que identifique unicamente o local. Assim, por exemplo, o sistema deve identificar se o local "8" (Figura 4.7), é na realidade uma outra vista do local "2" ou é de fato um local não visitado que deve ser adicionado ao P-MAP como um novo nodo.

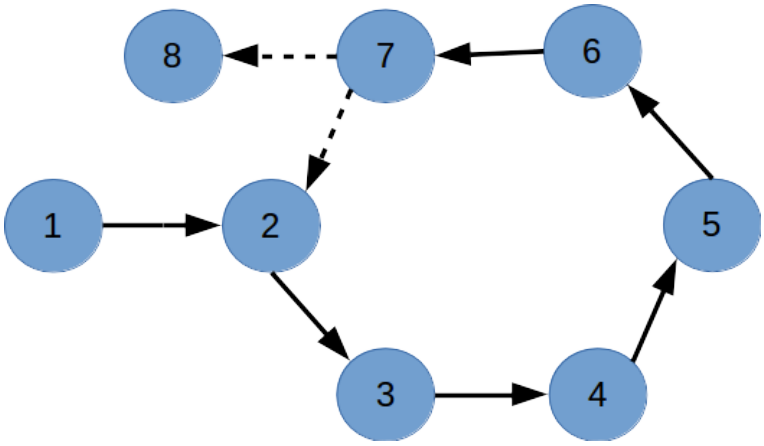


**Figure 4.7:** Construção do mapa topológico.

Nas subseções seguintes, são detalhadas a descrição das características visuais de um local, a utilização do I-MAP e do P-MAP e os codebooks utilizados para realizar as associações

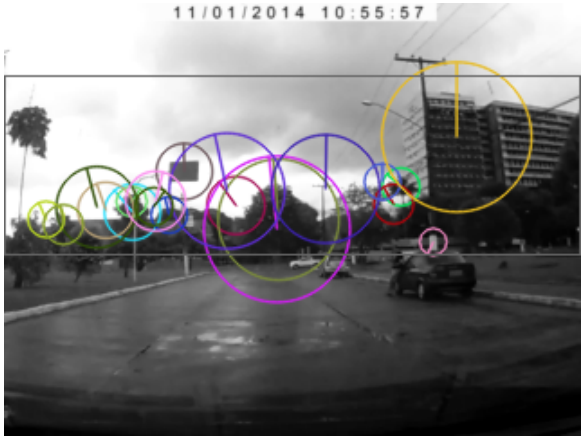


**Figure 4.8:** Imagem da Avenida dos Reitores na UFPE. Cada circulo é um *keypoint* SURF-128, vetor de dimensão 128 e magnitude entre 0 e 1. Estes vetores são invariantes a translação, rotação e escala e robustos a mudanças de perspectiva.

entre os nodos do I-MAP e do P-MAP.

### 4.3.1 Descrição de Características Visuais de um Local

As informações visuais escolhidas para serem utilizadas no sistema proposto foram descritores SURF (BAY H. ESS, 2008), um tipo de extrator de características muito utilizado em sistemas de registro e de busca por imagens (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011; DAWOOD et al., 2012). Nesta tese, o descritor SURF é modelado como uma função $SURF(\mathbf{I}, K)$, onde $\mathbf{I}$ é uma imagem digital e $K$ é o número de vetores de característica desejados. O retorno de $SURF(\mathbf{I}, K)$ é uma matriz $\mathbf{H}_{K,128}$. Cada linha da matriz $\mathbf{H}$ é um vetor de características associado a uma região ou objeto da imagem $\mathbf{I}$ que apresenta invariância a translação, rotação e escala realizadas sobre a imagem $\mathbf{I}$, Figura 4.8. Estas características de invariância conferem aos descritores SURF uma robustez para relacionar objetos semelhantes em cenas distintas. Entretanto, apesar de existir uma baixa probabilidade de um mesmo vetor de características estar associado a dois objetos distintos, não há garantias da unicidade da representação dos vetores de características. Em relação ao problema de associar imagens com locais, há um segundo problema envolvendo descritores SURF: os diversos vetores presentes na matriz $\mathbf{H}$ podem estar associados a características fixas de um determinado local observado, como uma árvore ou prédio, ou podem estar associados a uma característica móvel não representativa do local, como um carro ou uma pessoa. Entretanto, é razoável supor que a maior parte da imagem observada é de componentes estáticos e consequentemente a maior parte dos vetores de características estão, também, relacionados com características estáticas de uma cena observada. Esta é uma premissa utilizada em outras abordagens de SLAM Visual (GIBBENS; DISSANAYAKE; DURRANT-WHYTE, 2000; CUMMINS; NEWMAN, 2011), mas não é por si suficiente para garantir correspondências entre locais e sequências de imagens observadas.

### 4.3.2 Descrição do I-MAP

O objetivo principal do I-MAP é mapear características salientes em imagens que possam ser utilizadas para codificar locais. No sistema proposto o I-MAP é, na realidade, um mapa do tipo FSS-SOM. O sistema utiliza um banco de imagens previamente fornecido do qual se extrai uma lista de descritores SURF. Os descritores são utilizados como um conjunto de treinamento inicial para o I-MAP(FSS-SOM). O objetivo é que o I-MAP apresente nodos que representem os vetores SURF mais frequentes encontrados no conjunto inicial de imagens de treinamento, Figura 4.9. Assim, o I-MAP, através dos seus nodos, se torna um tipo de alfabeto para descrever imagens de locais. Deste modo, os locais visitados podem ser representados através de conjuntos de nodos ativados no I-MAP. Como os descritores SURF estão associados a microcaracterísticas presentes nas imagens e não as imagens em si, o banco de imagens fornecido para inicialização do I-MAP não precisa, necessariamente, estar relacionado com as imagens do ambiente no qual o agente irá se deslocar. Durante o deslocamento do agente, as imagens dos locais podem ou não serem utilizadas para atualizar o I-MAP.

Devido a utilização do FSS-SOM como I-MAP é possível controlar características do mapa como número máximo de nodos e distâncias mínimas entre nodos (parâmetros $d_0$,$d_1$,$d_2$). É possível ajustar o sistema para ter uma razão de compromisso entre tamanho da memória utilizada e fidelidade de representação dos vetores de características SURF. Há, também, a vantagem que vetores de características similares de observações distintas são mapeados nos mesmos nodos do mapa FSS-SOM.

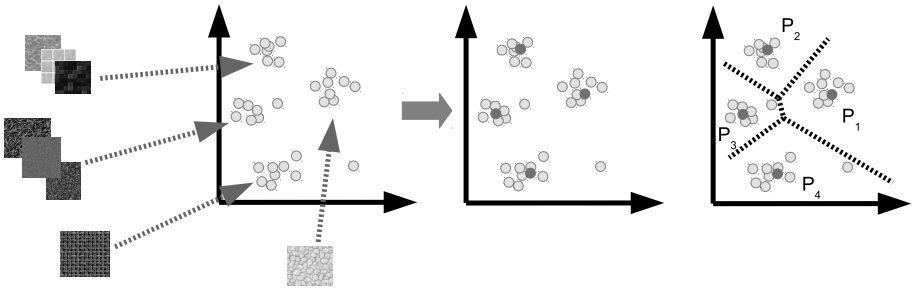


**Figure 4.9:** Ilustração do processo de agrupamentos dos descritores SURF utilizados no sistema de localização e mapeamento visual. A esquerda, composição de nuvens de pontos de descritores SURF de diferentes observações de uma determinada trajetória. No centro, agrupamento dos descritores e a direita a transformação de cada agrupamento em palavras que serão utilizadas para identificar os locais.

O procedimento de codificação de imagens de locais $\mathbf{I}_n$ utilizando o I-MAP consistem em determinar qual é o conjunto $J$ dos nodos do I-MAP ativados para os $k$ descritores SURF mais relevantes na imagem $\mathbf{I}_n$ (Algoritmo 5). Este procedimento é semelhante a criação de um *bag of words*. Entretanto, o retorno da codificação é um conjunto com a rotulação dos nodos ativados e não um vetor de ocorrências, linhas 4 e 5 do algoritmo. Em um *bag of words*, cada dimensão do vetor registra o número de ocorrências de uma determinada característica de

modo semelhante a um histograma. Quando o número de características se torna muito grande é comum o vetor de um *bag of words* se tornar esparso, apresentando muitas dimensões com ocorrência "zero". A utilização de conjuntos é uma forma de contornar a utilização de vetores esparsos e utilizar representações mais compactas de memória. A desvantagem desta abordagem é que utilizando-se conjuntos registra-se apenas se uma determinada característica foi observada ou não em uma determinada imagem. A informação da frequência da ocorrência não é registrada. Apesar deste fator negativo, se um I-MAP apresentar, por exemplo, 1000 nodos e for utilizado no mínimo conjuntos com 10 elementos para identificar locais visitados seria possível criar $C_{10}^{1000} = 2.63 \cdot 10^{23}$ combinações diferentes. Isso sugere que, mesmo sem registra a frequência da ocorrência de uma determinada microcaracterísticas em uma imagem é pouco provável que dois locais distintos possam ser representados com uma mesma codificação.

**Algorithm 5** Codificação de locais utilizando o I-MAP

1: Recebe uma imagem: $\mathbf{I}$
2: Recebe uma lista de nodos do I-MAP: $\mathbf{w}_i \therefore 0 \leq i < N_{max}$
3: Calcula os $\mathbf{w}_k$ descritores SURF mais relevantes da imagem $\mathbf{I}$, onde: $\mathbf{w}_k \therefore 0 \leq k < M_{max}$
4: Cria um conjunto $J = \{i \in \mathbb{N} | 0 \leq i < N_{max}\}$ dos rótulos dos $\mathbf{w}_i$ nodos mais próximos dos $\mathbf{w}_k$ descritores de $\mathbf{I}$
5: Retorna $J$

### 4.3.3 Codebook-IP e Codebook-PI

O conjunto dos rótulos dos $K$ nodos ativados do I-MAP para um determinado local fica armazenado em uma tabela denominada Codebook-IP. Esta tabela associa um determinado conjunto dos nodos do I-MAP ativados com um nodo do P-MAP. Uma outra tabela, codebook-PI, realiza a associação em sentido inverso. Ou seja, dado um nodo do P-MAP é possível identificar todos os nodos no I-MAP que devem ser ativados naquele local.

A principal finalidade dos codebooks utilizados pelo sistema é realizar as associação entre imagens, no I-MAP, e locais, no P-MAP e organizar as informações contidas no sistema. Permitindo associar uma imagem de um local com um nodo no mapa topológico sem a necessidade de realizar buscas exaustivas. Entretanto, dependendo da aplicação é possível associar outras informações ao locais e as imagens. Na Figura 4.10, há uma ilustração das informações contidas nos nodos dos mapas I-MAP, P-MAP e de informações compartilhadas do sistema proposto. Por exemplo, a partir de um nodo do P-MAP é possível acessar diretamente qual foi a imagem de referência que deu origem ao nodo, quais foram os keypoints SURF da imagem, quais são os nodos do I-MAP associados, etc. A quantidade de registros no codebook-PI é igual ao número de nodos no P-MAP e a quantidade de registros no codebook-IP, igual ou menor que o número de nodos no mapa I-MAP.

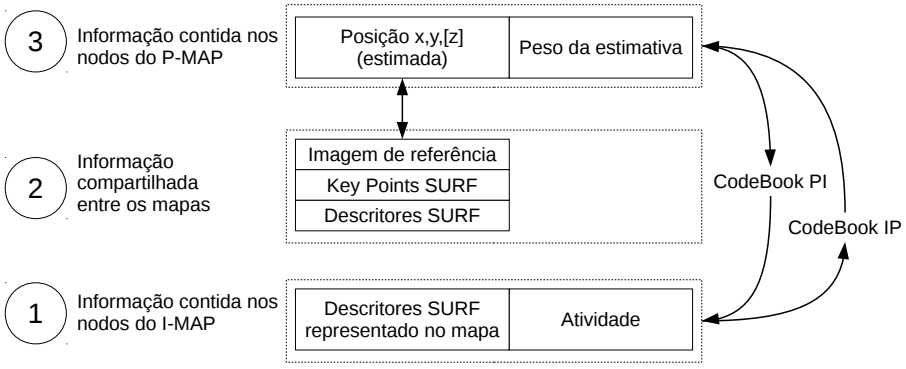


**Figure 4.10:** Ilustração do compartilhamento da informação visual entres os mapas I-MAP e P-MAP.

### 4.3.4 Critério de Semelhança entre dois Locais

O critério de semelhança utilizada pelo sistema para identificar se um determinado local já foi ou não visitado se baseia na contagem do número de elementos comuns entre duas codificações realizada pelo I-MAP. Assim, por exemplo, a semelhança $s$ entre duas imagens $\mathbf{I}_1$ e $\mathbf{I}_2$ é calculada como na Equação 4.2.

$$s = |J_1 \cap J_2| \tag{4.2}$$

Onde, $s$ é o valor da semelhança entre duas imagens, $J_1$ e $J_2$ são, respectivamente, as codificações dos locai das imagens $\mathbf{I}_1$ e $\mathbf{I}_2$ utilizando o I-MAP. O operador $|.|$ é a cardinalidade de um conjunto, ou seja, $|A|$ de um conjunto $A$ é igual ao número de elementos do conjunto $A$. Assim, quando maior $s$, mais semelhantes serão as imagens $\mathbf{I}_1$ e $\mathbf{I}_2$.

No sistema, a cardinalidade de uma codificação $J$ pode ser no máximo igual a quantidade $K$ de descritores SURF calculados sobre cada imagem. No melhor caso os $K$ descritores SURF serão associados a $K$ nodos diferentes do I-MAP, resultando em uma codificação $J$ com cardinalidade igual $K$. No pior caso, os $K$ descritores SURF podem todos serem associados a um único nodo do I-MAP, resultando em uma codificação $J$ com cardinalidade 1. Neste caso a comparação com outras imagens fica prejudicada, podendo resultar em associações espúrias entre duas imagens de locais distintos. Este tipo de associação é especialmente catastrófica no processo de construção das relações de vizinhança do P-MAP. Assim, pode-se afirmar que quanto menor o valor de $s$, menor será a confiabilidade na medida de semelhança entre as imagens.

### 4.3.5 Descrição do P-MAP

No sistema o P-MAP é responsável pela memorização dos locais visitados e geração do mapa topológico. Para gerar o mapa topológico, uma heurística semelhante a do mapa FSS-SOM é utilizada para realizar as conexões dos diversos locais memorizados pelo sistema. Ou seja, o

P-MAP segue, também, três princípios básicos:

1. Se a imagem capturada for muito diferente de todas as representações já armazenadas na memória do sistema, considerar que a imagem capturada é proveniente de um local não visitado e adiciona um novo nodo no mapa topológico conectado com o último local visitado.

2. Se a imagem capturada for muito semelhante a uma das representações já armazenadas e não corresponder ao local visitado imediatamente anterior, considera que o sistema está revisitando um local já memorizado e cria uma conexão entre o local memorizado e o local visitado imediatamente anterior.

3. Se a imagem capturada não for muito semelhante e nem muito diferente das representações de locais memorizadas, não realiza nenhuma inserção.

Os princípios utilizados para formar o mapa topológico do sistema são análogos aos utilizados no FSS-SOM. O mapa topológico gerado é um grafo formado $M$ nodos e $N$ conexões que deve preservar a mesma topologia da trajetória realizada pelo agente.

Diferentemente do I-MAP, que precisa ser inicializado com um conjunto de imagens antes do agente iniciar a navegação do ambiente, o P-MAP é gerado online durante o processo de navegação. O mapa inicialmente só possui um nodo que representa a posição atual do agente. A medida que o agente se desloca e realiza aquisições de imagens do ambiente o P-MAP é alimentado com novas imagens e cria novos nodos sempre que for alimentado com uma imagem de um local novo. O critério de comparação de locais utilizado no P-MAP é a discutida na subseção 4.3.4. Os novos nodos inseridos no P-MAP sempre são conectados com os nodos da posição topológica visitada no instante de tempo imediatamente anterior. Quando o sistema detecta que um determinado local já foi visitado, o nodo topológico associado a este local é conectado ao nodo visitado anteriormente. O procedimento de construção do P-MAP está sintetizado no Algoritmo 6.

O P-MAP utiliza dois limiares de semelhança $s_0$ e $s_1$ para controlar o crescimento do mapa topológico, com $s_0 < s_1$. O limiar $s_0$, determina qual o maior valor de semelhança que um local novo pode ter com qualquer outro já armazenado no mapa. O limiar $s_1$, indica o mínimo de semelhança para que dois locais sejam considerados iguais. Assim, o sistema não realiza modificações no P-MAP se um local inspecionado apresentar sempre uma semelhança maior que $s_0$ e menor que $s_1$ em relação a qualquer outro local memorizado no P-MAP.

O sistema de comparações do P-MAP utiliza, também, um outro limiar, $c_{min}$, para ignorar todas as imagens que apresentem uma codificação $J$ com cardinalidade baixa, $|J| \leq c_{min}$ (Linha 5 do Algoritmo 6). A principal finalidade é evitar que comparações de locais sejam realizadas quando as codificações $J$, realizadas pelo I-MAP, apresentarem uma cardinalidade muito baixa e, por consequência, pouco confiável.

As informações associadas aos nodos do P-MAP dependem, essencialmente, da aplicação do sistema. No sistema implementado nesta tese, o codebook-PI associa os nodos do P-MAP com três informações básicas: nodos ativados no I-MAP para um determinado local, imagem de referência associada ao local, posição cartográfica do local (quando disponível).

**Algorithm 6** Construção do P-MAP

1: Inicializa as variáveis: $s_0$, $s_1$ e $c_{min}$
2: Inicializa o grafo: $G = \{nodo_i, con_j\}$
3: Inicializa a referência: $vlocal \rightarrow nodo_0$
4: **while** Nova codificação $J$ **do**
5:  **if** $|J| > c_{min}$ **then**
6:  $s_k = MAX_{0 \leq i \leq N-1} (|J_i \cap J|)$
7:  **if** $s_k < s_0$ **then**
8:  Insere um novo $nodo_x$ em $G$
9:  Conecta $(nodo_x, vlocal)$ em $G$
10:  Atualiza a referência $vlocal \rightarrow nodo_x$
11:  Atualiza o codebook-PI com $nodo_x$
12:  **else if** $s_k > s_1$ **then**
13:  Conecta $(nodo_k, vlocal)$ em $G$
14:  Atualiza a referência $vlocal \rightarrow nodo_k$
15:  **end if**
16:  **end if**
17: **end while**

No Algoritmo 6, o grafo $G$ com $i$ vértices($nodo_i$) e $j$ aresta($con_j$) representa o mapa topológico gerado durante a navegação. A referência $vlocal$ é uma variável que sempre aponta para o endereço do último local visitado no P-MAP. A variável $s_k$ é a semelhança entre as codificações $J$ e $J_k$ , associada ao $nodo_k$ do grafo $G$. Na linha 6 do Algoritmo, a semelhança $s_k$ é definida como o número de elementos comuns aos conjuntos $J$ e $J_k$, onde $J_k$ é a codificação memorizada que mais possui elementos em comum com $J$. As variáveis $s_0$, $s_1$ e $c_{max}$ são os limiares utilizados para determinar se um novo nodo deve ou não ser inserido no mapa topológico. Na linha 7 do Algoritmo, o sistema considera que um local não foi visitado se $s_k$ for menor que um limiar mínimo de semelhança $s_0$. Nas linha 8 e 9, se o local for considerado não visitado, um novo nodo $x$ no grafo $G$ é criado e conectado com o último nodo conhecido visitado pelo sistema. Na linha 12 e 13 do Algoritmo, o sistema considera que um local já foi visitado se $s_k$ for maior que o limiar $s_1$ (linha 12) e realiza a conexão do último local visitado com o local atual no grafo $G$ (linha 13). As linhas 10 e 14 do Algoritmo, indicam o procedimento de atualização da referência do último local conhecido visitado pelo sistema.

# 5

# Resultados

Neste capítulo são apresentados os resultados obtidos com o modelo proposto de Mapa Auto-organizável e com o novo sistema de localização visual e mapeamento topológico. As primeiras seções apresentam os resultados do desempenho apenas do Mapa Auto-organizável utilizado como um método de agrupamento e como um previsor temporal. A partir da seção 5.3, são apresentados os testes e resultados com o sistema neural de localização visual e mapeamento topológico proposto.

## 5.1 Metodologia para Validar o Desempenho do Mapa

O modelo do Mapa Auto-organizável FSS-SOM foi testado utilizando dados de duas séries temporais não lineares diferentes, a de Mackey-Glass (MACKEY; GLASS et al., 1977) e a do índice do oscilador do pacífico (Southern Oscillation Index ou SOI). A série de Mackey-Glass é uma série temporal caótica produzida por simulação numérica de uma equação diferencial, ver Equação 5.1, e é frequentemente utilizada para testar modelos de previsão temporal. Nesta tese, a série de Mackey-Glass possui os seguintes parâmetros de formação: 900 pontos, intervalo de integração de 0.01, valor inicial de $x = 1.2$ e coeficientes $a = 0.2$, $b = 0.1$ e $d = 17$ da Equação 5.1. Estes valores foram adotados a partir do trabalho de STRICKERT; HAMMER (2005), que também utiliza a série de Mackey-Glass para realizar testes comparativos em modelos de previsão temporal baseados em Mapas Auto-organizáveis.

$$\frac{dx}{d\tau} = bx(\tau) + \frac{ax(\tau - d)}{1 + x(\tau - d)^{10}} \,. \tag{5.1}$$

A segunda série temporal adotada nos teste foi a SOI (AUSTRALIAN GOVERNMENT, 2014a), que é uma série de um indicador climático oscilatório do oceano pacífico, utilizada para realizar previsões sobre a ocorrência anual de eventos como o El Niño. Esta série é formada por 1656 medidas mensais de Jan/1876 à Dec/2013. Estes dados são disponibilizados livremente pelo Bureau de Meteorologia do Governo da Austrália (AUSTRALIAN GOVERNMENT, 2014b).

Nos testes, as entradas do modelo são vetores $\mathbf{v}(t)$ de dimensão 30, produzidos a partir de uma janela deslisante sobre as séries utilizadas como conjuntos de treinamento. A saída

do sistema, um valor escalar $\hat{y}(t+k)$, é a previsão do valor que a série temporal irá assumir em $k$ instantes de tempo a frente da janela utilizada na entrada ($\mathbf{v}(t)$). Assim, o índice $t$ denota a posição da janela deslizante na série temporal utilizada. Para as duas séries temporais utilizadas, o conjunto de treinamento foi formado por pares entrada-saída $S_{tk} = \{\mathbf{v}(t), \hat{y}(t+k)\}$ construído a partir de $1/3$ dos dados de cada série amostradas em intervalos regulares com deslocamento de 3 ao longo das respectivas séries. O conjunto de teste, com pares entrada-saída $W_{tk} = \{\mathbf{v}(t), \hat{y}(t+k)\}$ foi formado a partir de deslocamentos unitários.

Os mapas utilizados nos testes foram treinados com 9000 épocas para cada conjunto de treinamento com os valores de $k$ variando de 1 a 30. Os mapas de topologia fixa, SOM e RecSOM, foram construídos em forma de grid de 10 por 10 nodos e os de topologia variável com o tempo, GNG e FSS-SOM, foram ajustados para crescerem até aproximadamente 100 nodos. Após o treinamento não-supervisionado dos mapas, um codebook foi compilado para realizar as associações entre os nodos disparados com um vetor de entrada $\mathbf{v}(t)$ e um valor médio associado de saída $\hat{y}(t+k)$.

Nas Figuras 5.1, 5.2, 5.3 e 5.4 é possível visualizar o resultado iniciais das previsões temporais dos mapas em relação a um trecho da série de Mackey-Glass. Visualmente é possível identificar que os mapas GNG e FSS-SOM realizam uma previsão que aparenta ser melhor. Entretanto, a simples inspeção visual não é suficiente para realmente quantificar o desempenho do FSS-SOM em relação aos demais mapas.

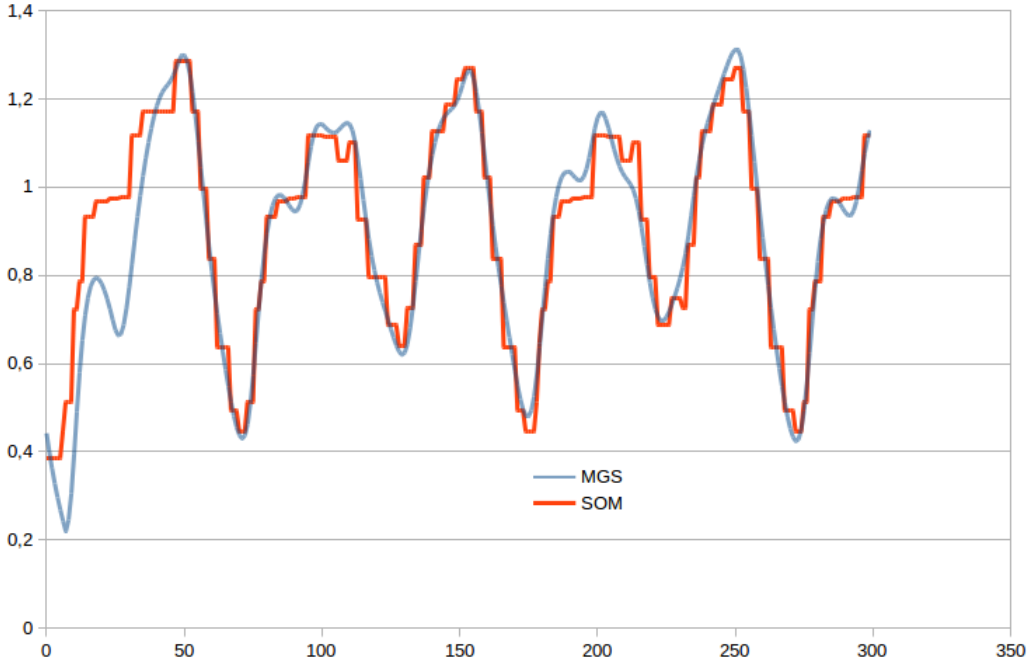


**Figure 5.1:** Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede SOM ao longo do tempo.

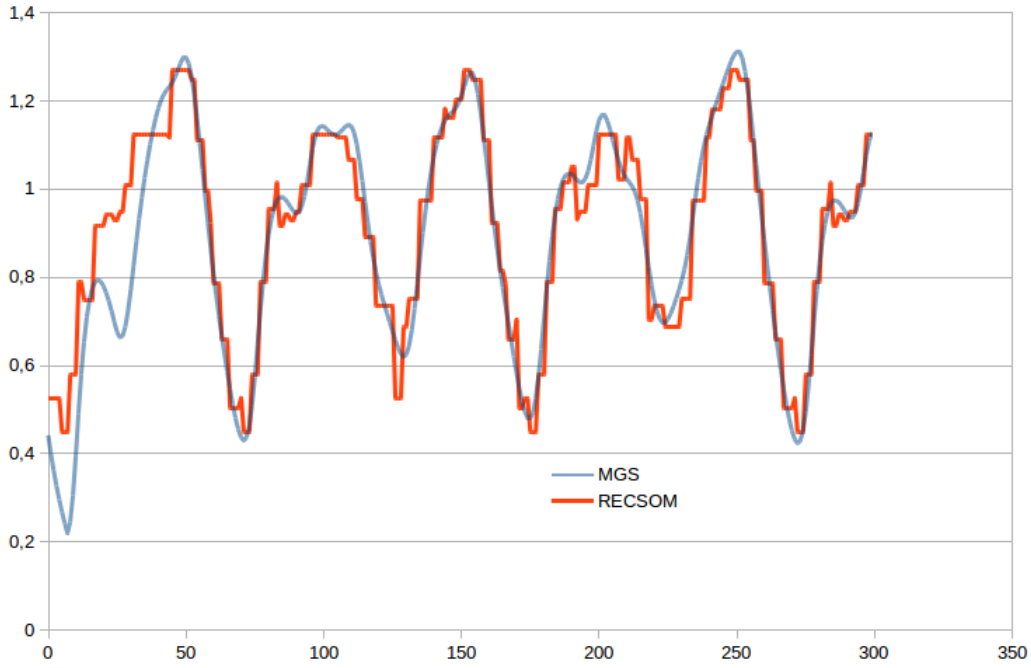


**Figure 5.2:** Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede RECSOM ao longo do tempo.

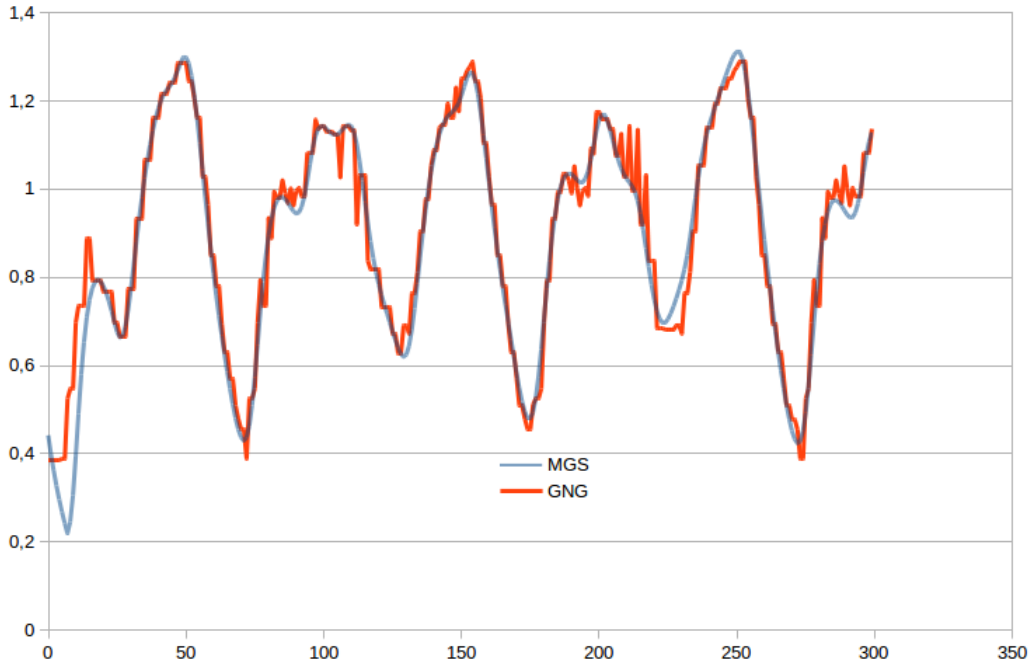


**Figure 5.3:** Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede GNG ao longo do tempo.

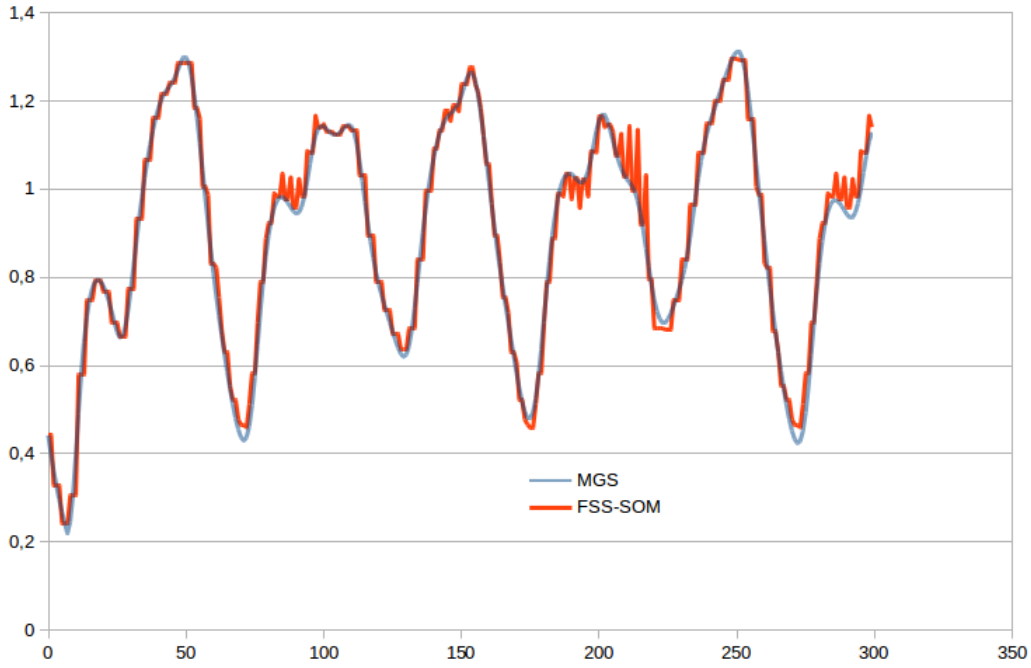


**Figure 5.4:** Gráfico da previsão temporal da série de Mackey-Glass utilizando uma rede FSS-SOM ao longo do tempo.

Com o intuito de ter comparações quantitativas, foram realizadas avaliações utilizando-se dois tipos de métricas. Em uma primeira abordagem, o desempenho dos mapas foi avaliado utilizando o erro quadrático médio (MSE) não-supervisionado e considerando apenas a capacidade dos mapas representarem os vetores de entrada $\mathbf{v}(t)$ em seus respectivos nodos, como em uma abordagem de agrupamento de um método com aprendizagem não-supervisionada. O desempenho foi avaliada calculando-se o MSE por época do treinamento para cada um dos mapas utilizados. Os resultados desta avaliação podem ser visualizados nas Figuras de 5.17, 5.18 e 5.19.

Uma segunda avaliação quantitativa foi conduzida utilizando as mesmas séries, mas considerando os pares entrada-saída $S_{tk}$, para o treinamento, e $W_{tk}$, para o teste. A avaliação do desempenho foi realizada também utilizando-se o MSE, mas calculado sobre a diferença dos valores de saída do sistema, já treinado, para os valores desejados do conjunto de teste ($\hat{y}(t+k)$). Neste caso, o sistema utiliza um mapa auto-organizável com um codebook funcionando como um previsor para a série temporal e como um sistema de aprendizagem supervisionada (MARTINETZ T., 1993). Este configuração foi baseada em (STRICKERT; HAMMER, 2005).

Na utilização de Mapas Auto-organizáveis como previsores, os mapas são treinados apenas com os padrões de entrada do sistema. Após esta etapa, os nodos dos mapas são rotulados com uma numeração inteira e um codebook é criado para realizar a associação dos nodos disparados com a apresentação de um padrão de entrada com o seu respectivo padrão de saída. Caso mais de um padrão diferente esteja associado a um mesmo nodo, a saída armazenada no codebook corresponderá a uma média das respectivas saídas desejadas para os padrões de entrada. A etapa de associação dos pares entrada-saída via codebook é realizada após a etapa de

treinamento não-supervisionada dos mapas.

Adicionalmente aos mapas GNG, RecSOM e SOM dois previsores simples de séries temporais (GOOIJER; HYNDMAN, 2006) foram utilizados para formar uma base de comparação de desempenho mais ampla para o FSS-SOM. O primeiro é um previsor ingênuo, que apresenta como previsão para o próximo valor da série um valor numericamente igual ao do último instante de tempo conhecido da série. Fazendo, $\hat{y}(t+k)$ (valor da previsão) igual ao valor em instante atual, $y(t)$. O segundo é um previsor aleatório ou Random, que faz $\hat{y}(t+k)$ igual a um valor proveniente de um sorteio. Este previsor tem como principal objetivo servir de linha de base para avaliar o quão melhor é um previsor em relação a um sistema que fornece resultados aleatoriamente.

Os resultados para o sistema funcionando como um previsor podem ser vistos nas Figuras 5.7, 5.13, 5.8, 5.14, 5.9, 5.15, 5.10, 5.16. As Figuras 5.5, 5.11, 5.6, 5.12 funcionam como uma referência para a solução trivial.

Nas simulações com objetivo de obter resultados que representassem o desempenho médio dos mapas e evitar a penalização de algum modelo por uma escolha ruim de parâmetros, foi utilizada uma técnica de variação paramétrica conhecida como *Latin Hyper Cube* ou LHC (SALTELLI A., 2009). Esta técnica permite que um conjunto de parâmetros com máxima independência possa ser gerado em uma determinada faixa paramétrica para o modelo (HELTON; DAVIS; JOHNSON, 2005). Nestes testes foi utilizada a técnica de LHC para selecionar 20 conjuntos diferentes de taxas de aprendizagem para os mapas utilizados nas comparações com o FSS-SOM. Outros parâmetros utilizados no GNG foram mantidos iguais aos do artigo publicado originalmente por Fritzke (FRITZKE, 1995). No SOM e no RecSOM, os parâmetro adicionais foram ajustados de acordo com o artigo publicado por VOEGTLIN (2002).

## 5.2 Resultados dos Testes com o FSS-SOM

O principal objetivo dos teste realizados foi comparar a desempenho do FSS-SOM com outros tipos de mapas de topologia fixa (SOM), variante no tempo (GNG) ou recorrentes (RecSOM). As Figuras 5.5 à 5.10 apresentam o *MSE* de cada sistema de previsão temporal considerando os dados da série de Mackey-Glass e as Figuras de 5.11 à 5.16 para a SOI.

Nos gráficos seguintes, a linha cheia central representa um resultado médio obtido nas simulações, as barras de erro apresentam o desvio padrão e as linhas externas os valores máximos e mínimos. No caso das Figuras 5.17 até 5.19 o desvio padrão, os valores máximos e mínimos são tão próximos ao valor médio que não aparecem no gráfico devido à escala utilizada.

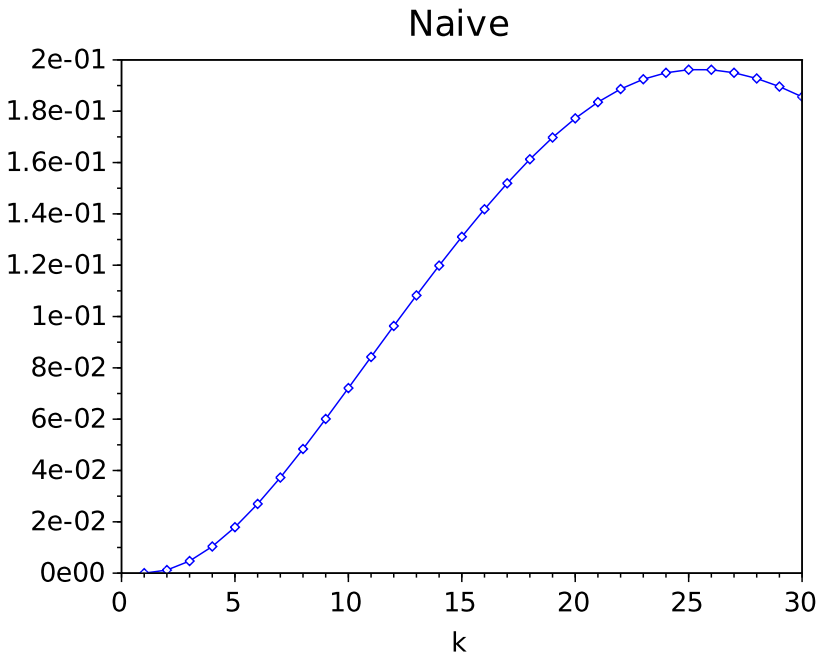


**Figure 5.5:** *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal ingenuo para a série de Mackey-Glass.

A Figura 5.5, apresenta o resultado do MSE da previsão da séria de Mackey-Glass utilizando um dos tipos mais simples que existem de previsor, o previsor trivial ou *naive*. Neste previsor, o valor da previsão é igual ao valor atual ou da entrada. É possível notar que ele apresenta um baixo MSE para valores de previsão em instantes de tempo próximos do atual, baixo $k$, mas o MSE cresce rapidamente com o aumento de $k$, horizonte da previsão. Este resultado é utilizado apenas como uma base de comparação para o desempenho dos Mapas Auto-organizáveis testados. Na Figura 5.11 é apresentado o mesmo previsor, mas aplicado a série temporal SOI.

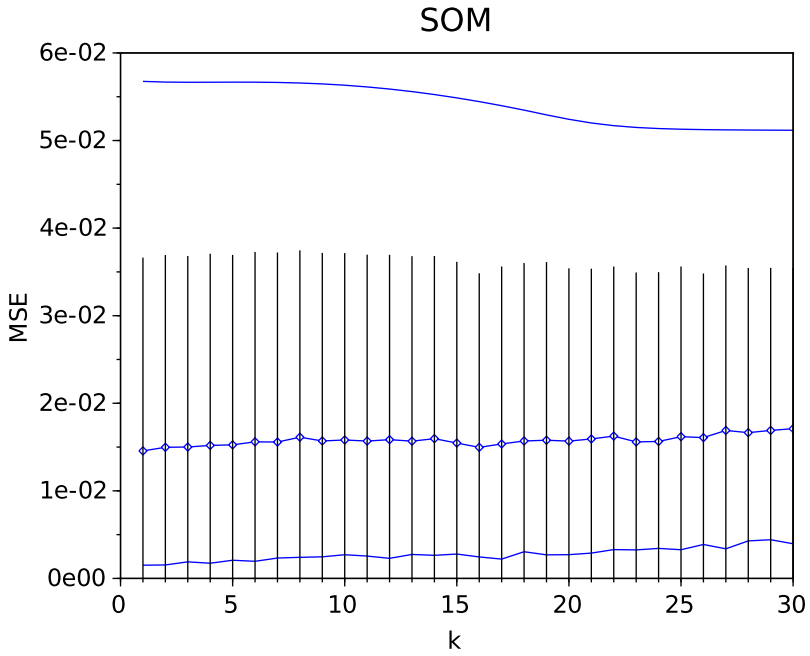


**Figure 5.7:** *MSE* variando de $1 \leq k \leq 30$ do SOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass.

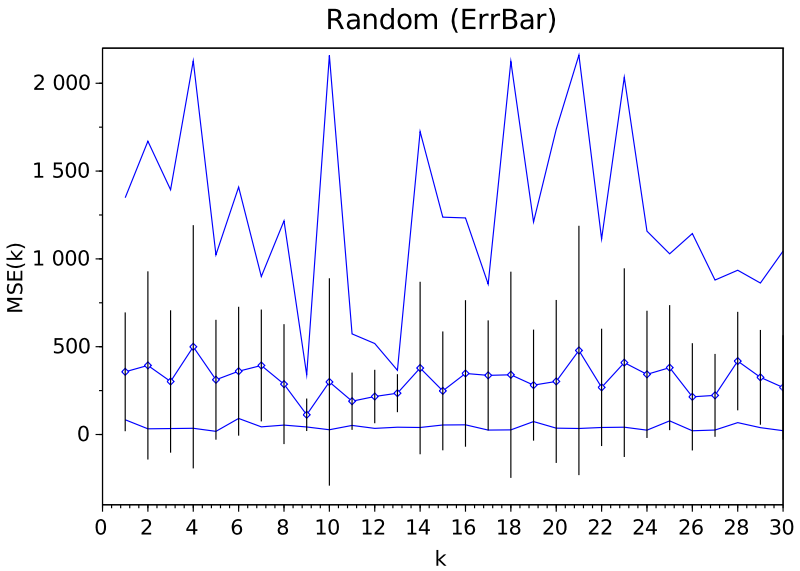


**Figure 5.6:** *MSE* variando de $1 \leq k \leq 30$ de um previsor do tipo *Random* para a série de Mackey-Glass.

Na Figura 5.6, o previsor do tipo *Random* é aplicado a série de Mackay-Glass. O valor da previsão é completamente aleatória e serve para estabelecer um limite superior para o erro de previsão de uma determinada série. A utilização deste previsor serve como linha de base para determinar o quanto distante está um modelo de previsão de um resultado completamente aleatório. Na Figura 5.12 é apresentado o mesmo previsor, porem aplicado a série temporal SOI.

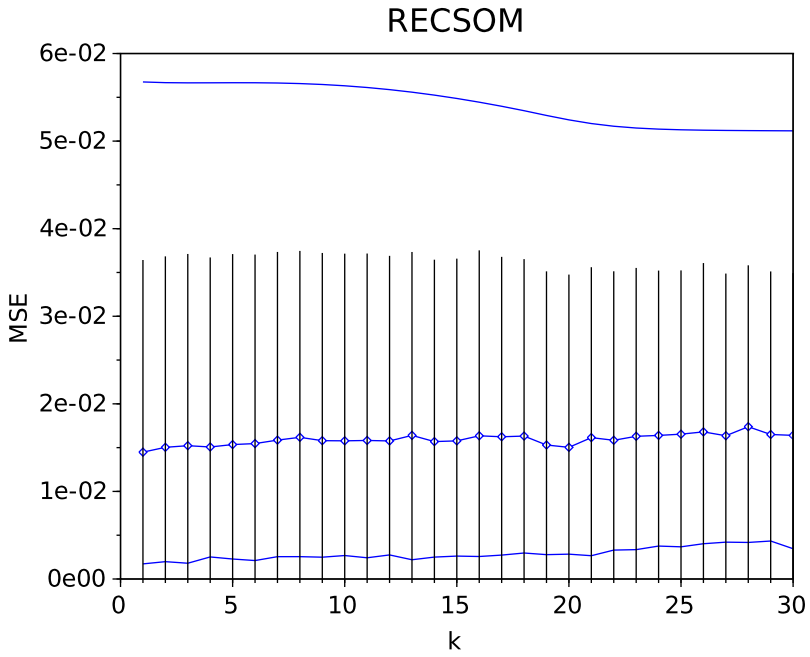


**Figure 5.8:** *MSE* variando de $1 \leq k \leq 30$ do RecSOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass.

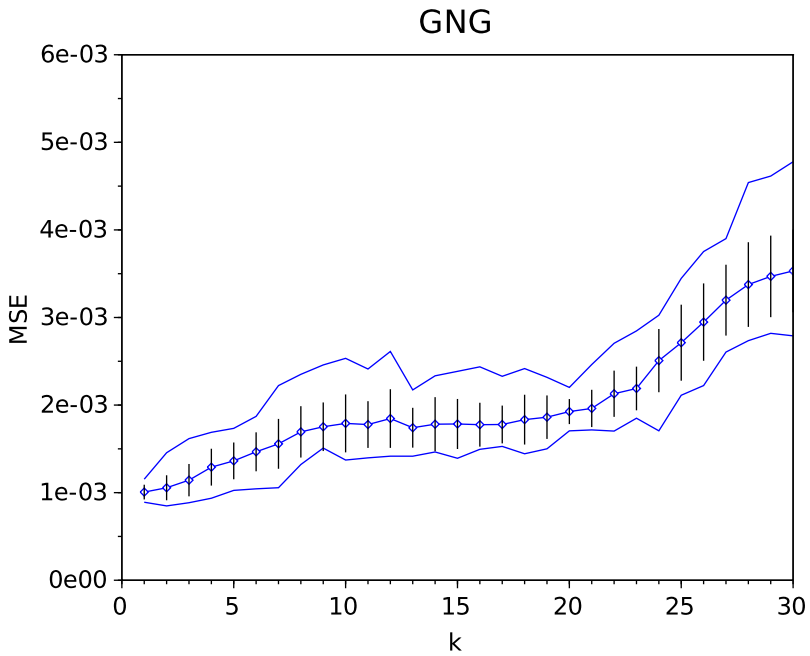


**Figure 5.9:** *MSE* variando de $1 \leq k \leq 30$ do GNG com um codebook de modo supervisionado como previsor para a série de Mackey-Glass.

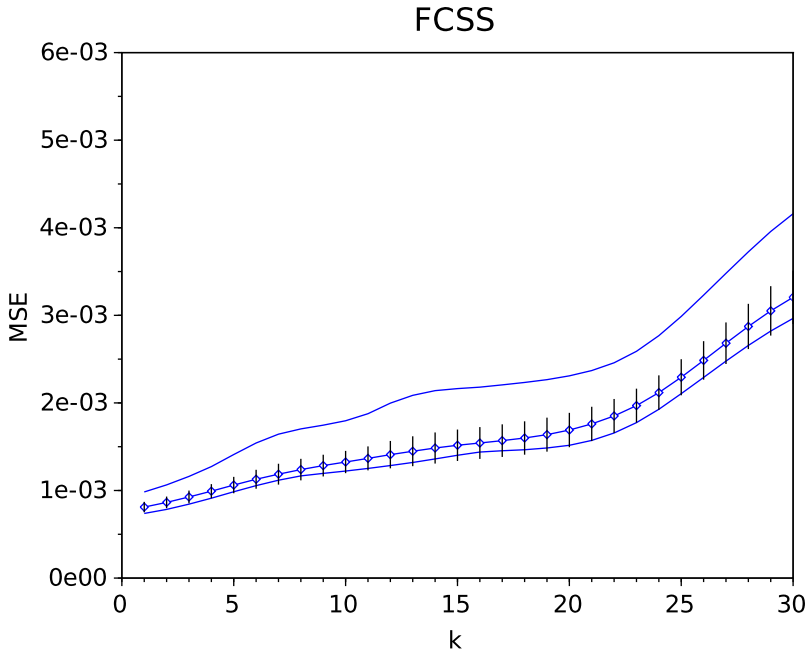


**Figure 5.10:** *MSE* variando de $1 \leq k \leq 30$ do FSS-SOM com um codebook de modo supervisionado como previsor para a série de Mackey-Glass.

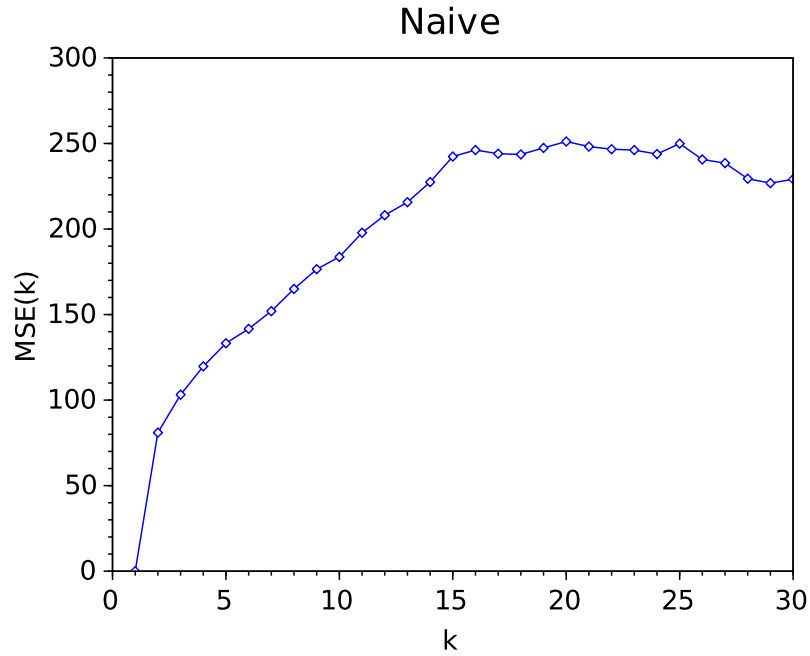


**Figure 5.11:** *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal ingenuo para a série SOI.

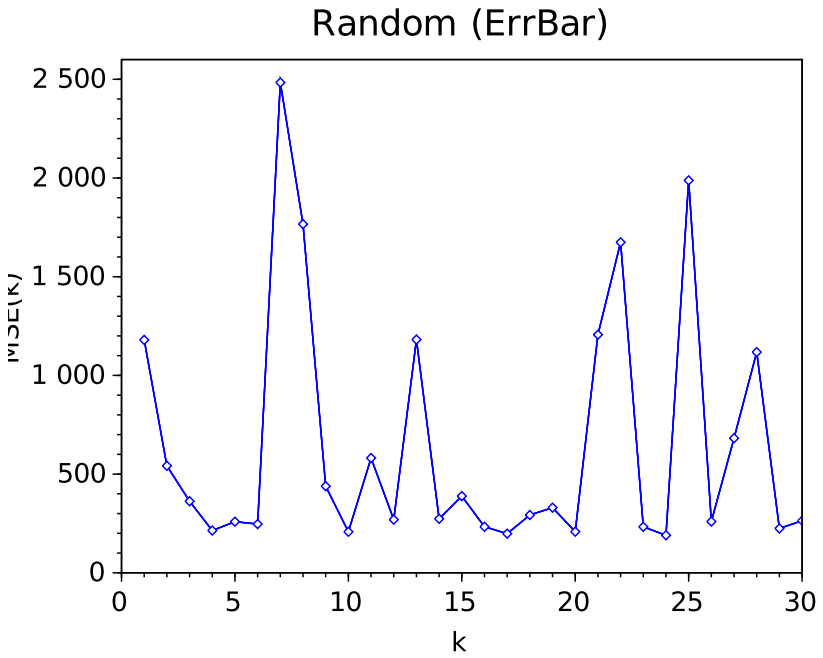


**Figure 5.12:** *MSE* variando de $1 \leq k \leq 30$ de um previsor temporal tipo Random para a série SOI.

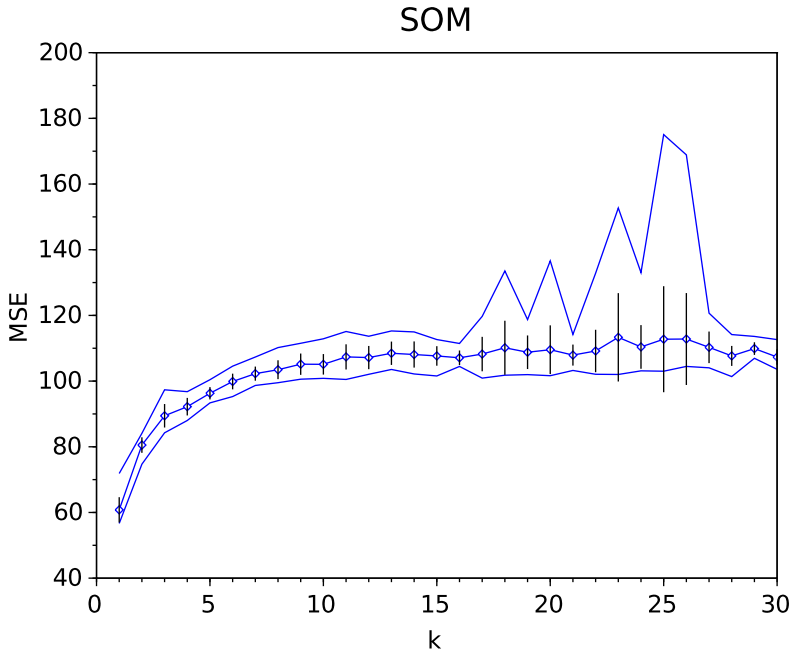


**Figure 5.13:** *MSE* variando de $1 \leq k \leq 30$ do SOM com um codebook de modo supervisionado como previsor para a série SOI.

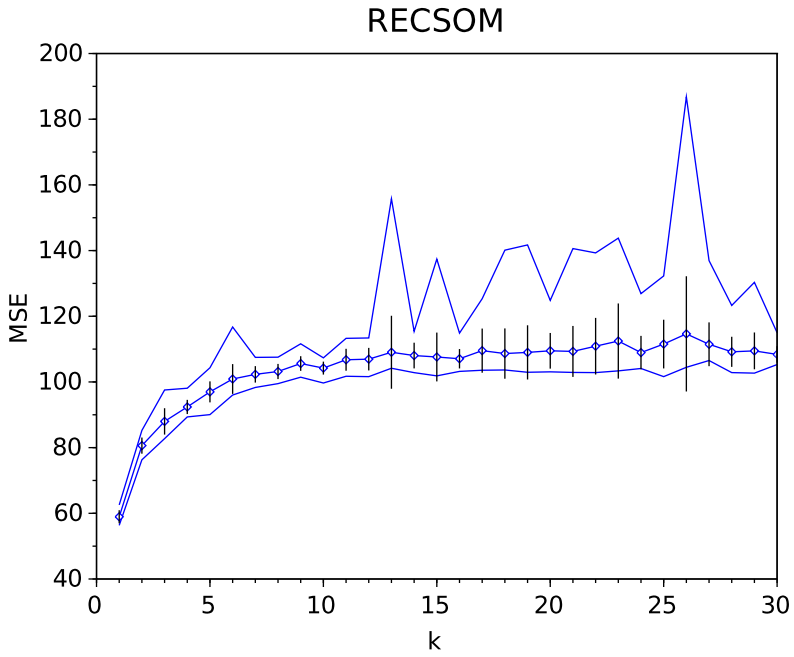


**Figure 5.14:** *MSE* variando de $1 \leq k \leq 30$ do RecSOM com um codebook de modo supervisionado como previsor para a série SOI.

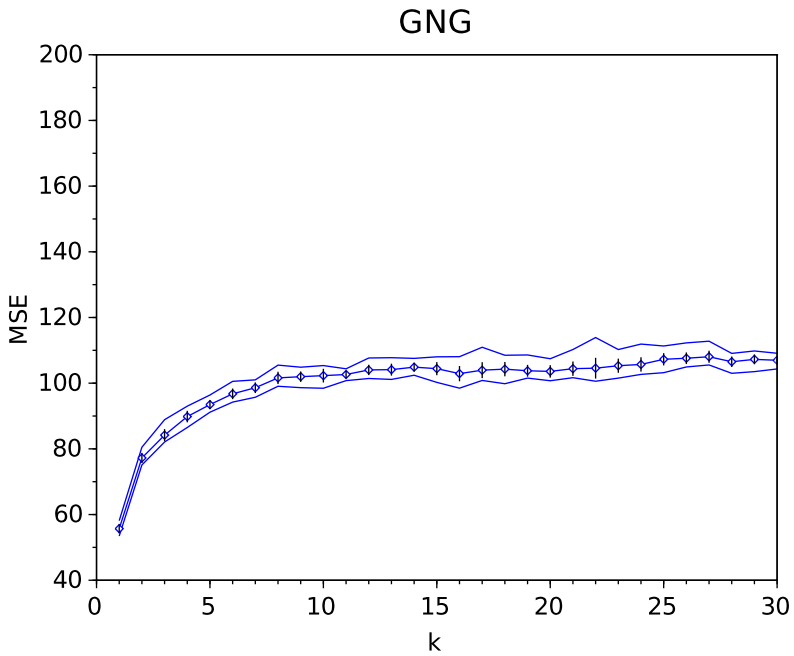


**Figure 5.15:** *MSE* variando de $1 \leq k \leq 30$ do GNG com um codebook de modo supervisionado como previsor para a série SOI.

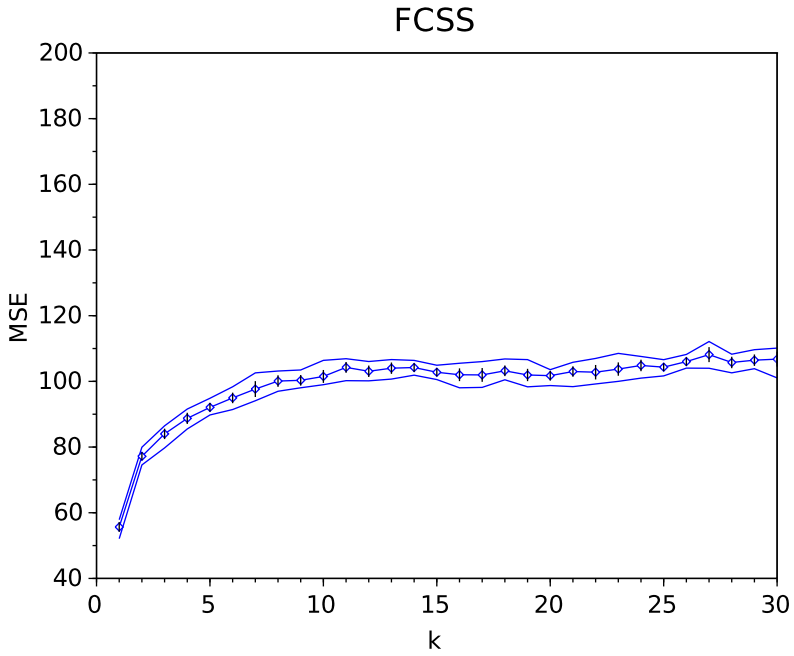


**Figure 5.16:** *MSE* variando de $1 \leq k \leq 30$ do FSS-SOM com um codebook de modo supervisionado como previsor para a série SOI.

Os resultados médios do MSE para a série temporal de Mackey-Glass com o SOM (Figura 5.7), RecSOM (Figura 5.8), GNG (Figura 5.9) e FSS-SOM (Figura 5.10) mostra o FSS-SOM com um desempenho numericamente melhor que a dos outros modelos avaliados, embora não seja possível afirmar que estatisticamente apresente um desempenho superior ao do GNG.

A previsão da série temporal SOI se mostrou uma tarefa bem mais complexa que a da série de Macky-Glass, principalmente devido às fortes e abruptas variações de amplitude que esta série apresenta e por possuir mais pontos de amostra por nodo que a de Mackey-Glass. Ainda assim, o FSS-SOM apresentou um desempenho médio do MSE numericamente melhor que a dos outros mapas testados.

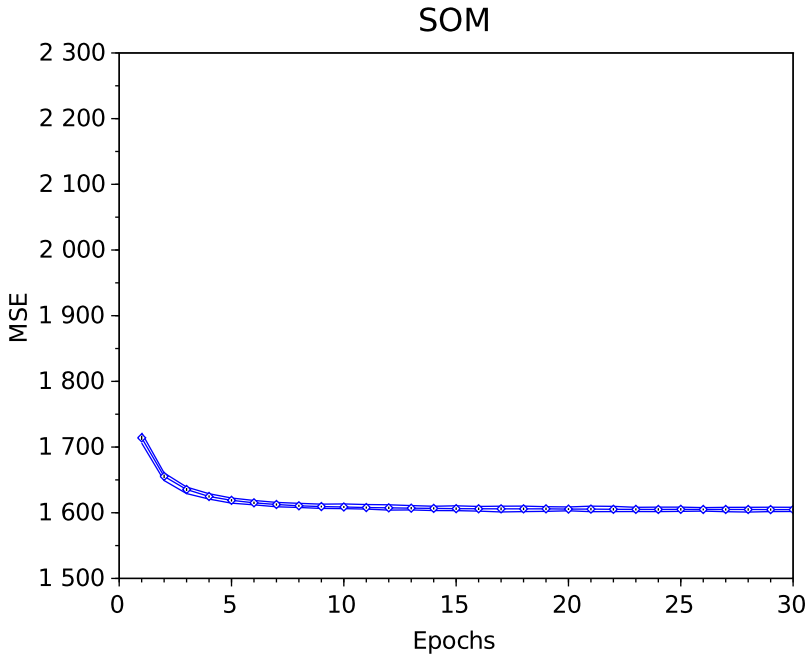


**Figure 5.17:** *MSE* por *Epoca* do SOM, os resultados utilizando o RecSOM são praticamente identicos ao do SOM e não estão incluídos no gráfico.

As Figuras 5.17 até 5.19 mostram que o FSS-SOM tem o menor MSE por época, mas também que é o mapa com convergência mais rápida entre os modelos testados. O FSS-SOM em 5 épocas apresenta um MSE menor que 1550, enquanto o GNG, por exemplo, apresenta um MSE de 1725. Os mapas SOM e RecSOM apresentam MSE de aproximadamente 1625 em 5 épocas. Estes gráficos mostram que o FSS-SOM claramente apresenta um crescimento e uma convergência mais eficiente que os outros modelos testados atingindo em poucas interações erros menores.

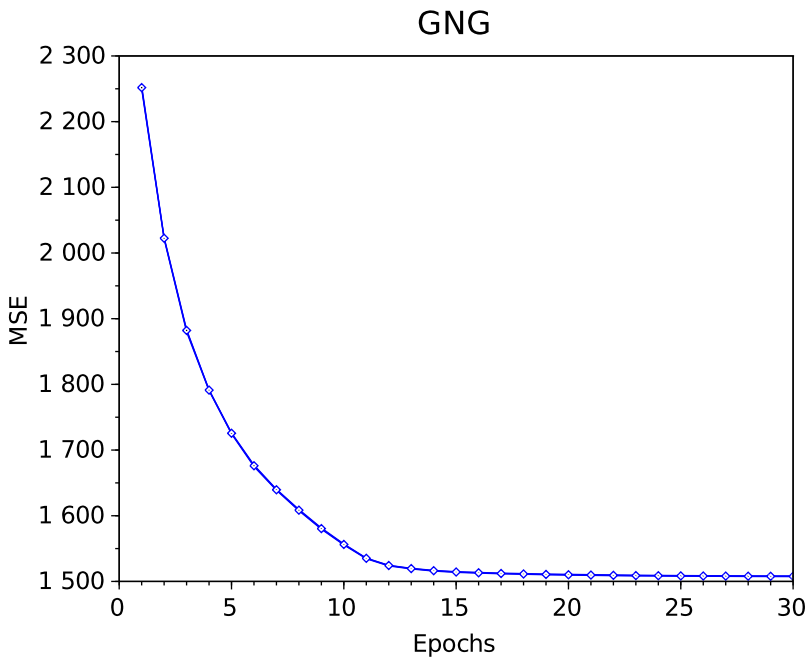


**Figure 5.18:** *MSE* por *Epoca* do GNG.

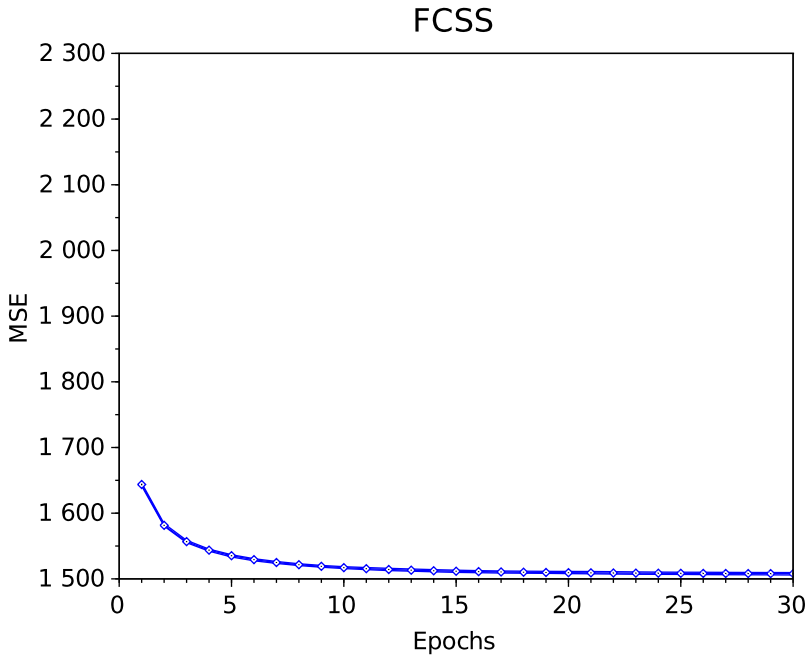


**Figure 5.19:** *MSE* por *Epoca* usando o FSS-SOM. Realizando uma comparação entre as Figuras 5.19,do FSS-SOM, e 5.18, do GNG, é interessante notar que o GNG com o mesmo número de nodos e conjunto de treinamento só consegue um MSE abaixo de 1550 após 12 épocas, enquanto o FSS-SOM atinge este valor em aproximadamente 5 épocas, evidenciando a característica de convergência rápida do mapa.

## 5.3 Metodologia Experimental para Validar o Sistema Proposto

O sistema de localização visual e mapeamento topológico foi testado em três partes. Um primeiro teste visava verificar se o critério de semelhança adotado pelo sistema, baseado em conjuntos, funcionaria adequadamente. Um segundo tipo de teste foi realizado para verificar a capacidade de localização visual do sistema. Por ultimo, foram realizados testes com localização visual e mapeamento topológico simultâneo.

Foram utilizados três tipos diferentes de conjuntos de dados: imagens de ambiente interno sem georreferenciamento e imagens de ambiente externo com e sem georreferenciamento. As aquisições visuais dos ambientes internos e externos adquiridos sem georreferenciamento foram obtidas, respectivamente, com: uma câmera de vídeo montada em uma plataforma robótica desenvolvida especialmente para realizar testes com o modelo proposto nesta tese (Capítulo A.1) e com um sistema de câmera de vídeo montado sobre um carro. As observações em ambientes internos foram realizadas nos corredores do Cin/UFPE e as observações externas foram realizadas no Campus da UFPE. As observações visuais em ambientes externos georreferenciadas foram obtidas a partir de dois bancos de teste para SLAM Visual (CUMMINS; NEWMAN, 2008b; GEIGER et al., 2013).

O sistema proposto realiza mapeamento topológico, assim não precisa necessariamente de imagens georreferenciadas. Nos testes onde esta informação estava disponível, os nodos do P-MAP armazenavam além de um rótulo numérico($i$), identificando o nodo, um vetor posição $\mathbf{x}_i$. A informação de posição, quando disponível, foi utilizada para verificar o desempenho do sistema ao se localizar e na verificação das vizinhanças do mapa topológico.

No caso dos testes que avaliaram o critério de semelhança adotado, foram utilizados as observações realizadas no interior do Cin/UFPE. Este conjunto fornece um ambiente mais simples e controlado, facilitando a verificar se o indicador de semelhança variava proporcionalmente a medida que um agente se aproxima ou se distancia de um local memorizado pelo P-MAP.

Nos testes de localização visual, todos os conjuntos de dados foram utilizados. Na localização visual, o que é avaliado é a capacidade do sistema conseguir associar corretamente novas imagens de locais, com locais já memorizados e selecionados previamente (SHEN; HU, 2006). Com as observações realizadas no interior do CIn/UFPE, foram testados as associação das imagens em relação a 17 pontos memorizados de uma trajetória linear e com as observações realizadas em ambiente externo, no campus da UFPE, foram testados as associações a 13 pontos memorizados de uma trajetória circular (Figura 5.24). Nestes dois casos, a verificação se o sistema relacionou corretamente a imagem de um local com um determinado ponto de referência memorizado foi realizada por inspeção visual. Devido a isto, não foram construídas curvas de Precison-Recall para estes dados. São apresentados apenas tabelas com as configurações dos mapas utilizados e o número de erros na localização.

Nos testes de localização visual utilizando as imagens georreferenciadas, a informação de distância entre a posição real das observações e os pontos de referência memorizados foi utilizada como métrica para construir curvas Precison-Recall. Utilizando as informações de georreferenciamento, associações entre um local inspecionado e um memorizado é considerada correta (um positivo verdadeiro) se a distância entre os locais for menor que um limiar $R$. Nos testes, um total de 6 valores diferentes variando entre 2 e 32 metros foram utilizados para construir as curvas Precison-Recall.

Todos os conjuntos de dados utilizados consideraram ambientes com componentes dinâmicos, como carros e pessoas em movimento e as informações utilizadas para realizar o treinamento da rede não foram utilizadas nos teste para avalização do desempenho.

Nos testes realizados com localização visual e mapeamento topológico simultâneo, foram utilizados os conjuntos de testes dos ambientes externos da UFPE, dos bancos de teste de SLAM Visual em Espaço de Aparências (CUMMINS; NEWMAN, 2008b).

Nestes testes, o objetivo foi produzidos mapas topológicos utilizando apenas as informações visuais disponíveis nos conjuntos de teste sem nenhum tipo de intervenção ou pré-seleção de regiões. O sistema de modo automático, realiza a detecção e memorização de novos locais não visitados, estabelecia a relação de vizinhança e detecta quando um local está sendo revisitado. Os resultados são apresentados na forma de grafos, com indicações visuais estabelecendo quando um local foi revisitado ou não.

Para os mapas topológicos construídos utilizando os bancos de teste de SLAM Visual em Espaço de Aparências, Figuras 5.33, 5.34 e 5.35, a informação de georreferenciamento dos locais foi utilizada apenas para organizar a disposição gráfica dos nodos dos mapas. Entretanto, as conexões e a detecção se um determinado local foi visitado mais de uma vez foram realizadas apenas utilizando informações visuais.

No caso do mapa topológico construído a partir do ambiente externo da UFPE, que não possui georreferenciamento, a disposição gráfica dos nodos do mapa e de suas conexões foi realizada manualmente a partir do conjunto de nodos e conexões apresentados pelo sistema, Figura 5.31.

## 5.4 Resultados do Sistema de Localização Visual e Mapeamento Topológico

### 5.4.1 Avaliação da Medida de Semelhança do Sistema

A medida de semelhança utilizado o I-MAP varia ao longo de um trecho inspecionado, à medida que o agente robótico se aproxima de um ponto já memorizado previamente, ou seja os locais de referência memorizados pelos I-MAP formam pontos de atração ao longo de uma trajetória. Nos Gráficos 5.20 e 5.23 é possível observar a evolução, frame-a-frame,

(a) (b) (c)

**Figure 5.21:** a) Frame 1, b) Frame 20 (cena memorizada) e c) Frame 36 do gráfico da Figura 5.20

da semelhança apresentada pelo sistema entre um frame do vídeo de teste em relação a um previamente memorizado. O critério de semelhança utilizado apresenta um máximo local exatamente na posição memorizada no treinamento. Os gráficos também mostram que as medidas de semelhança apresentam variações suaves. A semelhança aumenta na medida que o agente se aproxima do local memorizado como ponto de referência e diminuí quando se afasta. Figuras 5.21 e 5.22 observa-se os trechos dos frames memorizados (Figura central,b) e os frames anteriores e posteriores testados (Figuras a e c). Isto evidencia que existe uma relação que produz medidas de semelhanças maiores a medida que um agente se desloca na direção de um local já memorizado e produz um resultado inverso quando o agente se distância do local.

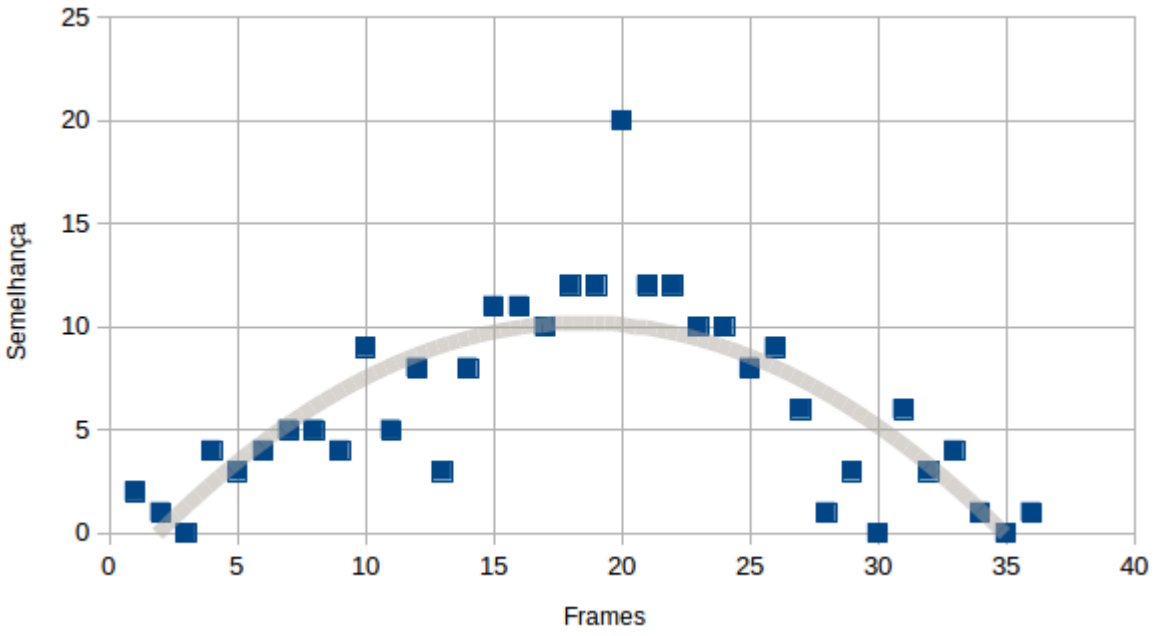


**Figure 5.20:** O gráfico apresenta a semelhança entre um determinado local, previamente memorizado(frame 20), e a sequência de frames ao longo do tempo.

(a) (b) (c)

**Figure 5.22:** a) Frame 1, b) Frame 12 (local memorizado) e c) Frame 19 do gráfico da Figura 5.20

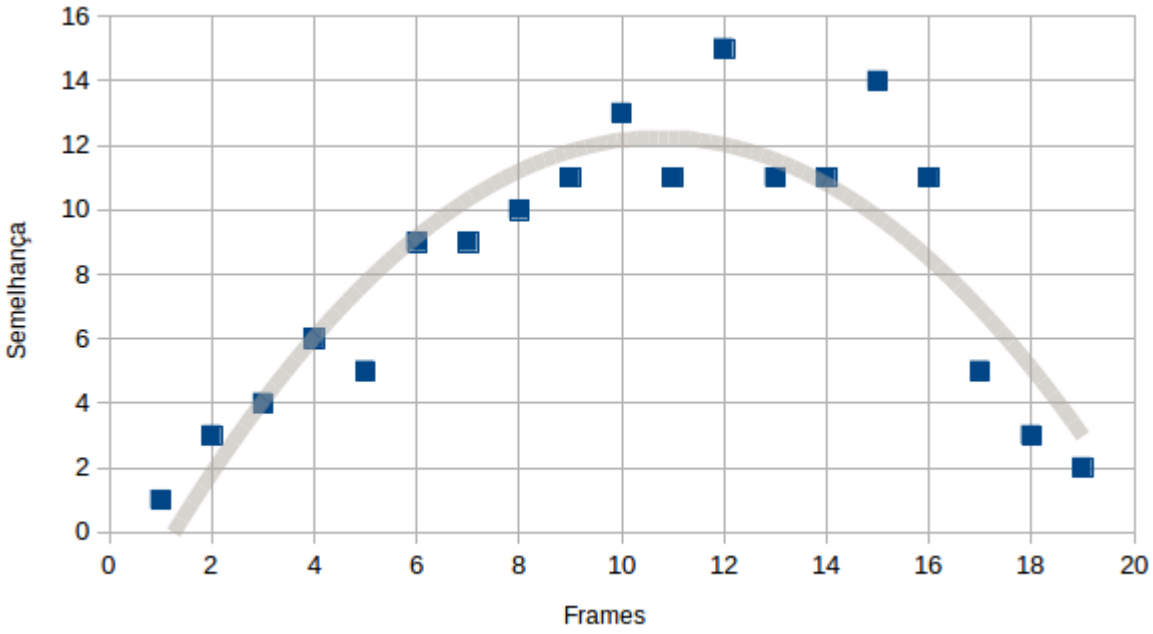


**Figure 5.23:** O gráfico apresenta a semelhança entre um determinado local, previamente memorizado(frame 12), e a sequência de frames ao longo do tempo.

### 5.4.2 Avaliação da Capacidade de Localização Visual

Nos testes de localização visual realizados no CIn/UFPE, com um conjunto de treinamento formado por 17 imagens de pontos de referência selecionados arbitrariamente de uma trajetória de 5598 inspecionada, houve apenas 27 identificações erradas de locais. Ou seja, em 5598 testes houve 27 erros onde o sistema identificou incorretamente a região onde o agente robótico estava localizado. Se for considerado o número total de testes realizados o erro é de 0.48%. Na Tabela 5.1 há resumo dos dados de erro e de mapas utilizados neste teste.

**Table 5.1:** Taxa de erros na localização dentro do Cin

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 5598 |
| Número de vetores SURF por frame: | 20 |
| Número mínimo de ativações no I-MAP para identificação do local: | 6 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 17 |
| Número de imagens utilizadas no conjunto de treinamento: | 17 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | 27 |

Um segundo experimento de localização visual foi realizado utilizando informações de uma filmagem a partir de uma câmera fixa em um carro se deslocando a 40km/h e realizando 7 voltas em um percurso fechado em ambiente externo no campus da UFPE (Figura 5.24) totalizando 26939 frames e apresentando apenas 3 erros de localização. Neste teste também não foi possível realizar georreferenciamento das imagens, sendo utilizado o mesmo procedimento de rotulação manual realizado no ambiente interno.

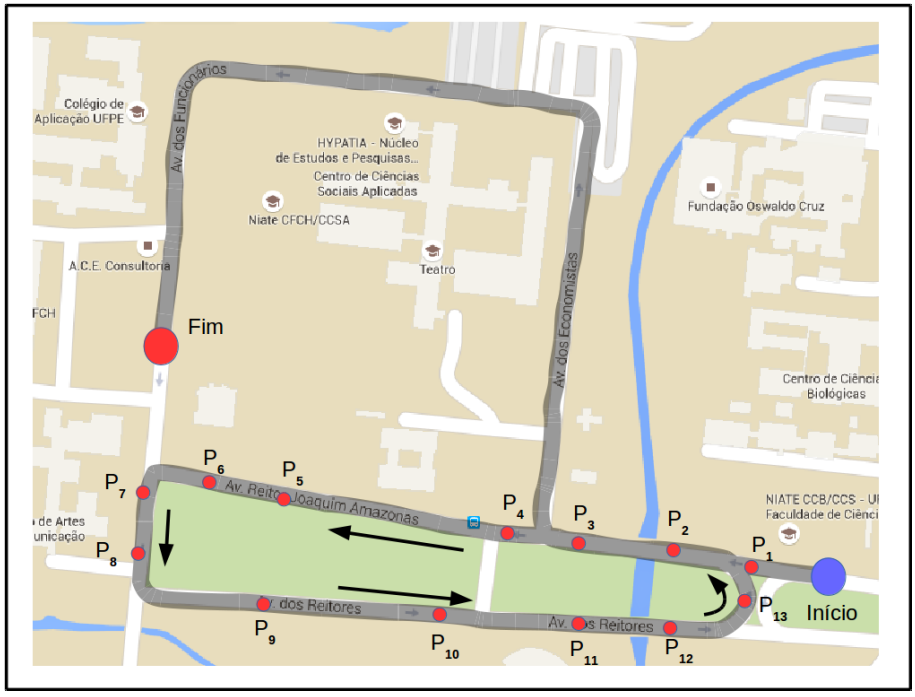


**Figure 5.24:** Vista superior da trajetória, em cinza, percorrida em ambiente externo no campus UFPE. (Mapa retirado do Google Maps)

A Tabela 5.2 apresenta um resumo do erro na identificação de locais no percurso externo

da UFPE e dos parâmetros utilizados.

**Table 5.2:** Taxa de erro na localização na Avenida dos Reitores, UFPE

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 26939 |
| Número de vetores SURF por frame: | 40 |
| Número mínimo de ativações no I-MAP para identificação do local: | 10 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 13 |
| Número de imagens utilizadas no conjunto de treinamento: | 4490 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | 3 |

O terceiro experimento realizado para avaliar a capacidade de localização visual foi utilizando *benchmarks*, utilizados em trabalhos de detecção visual de trajetórias fechadas e SLAM (CUMMINS; NEWMAN, 2008b; GEIGER et al., 2013), com imagens georreferenciadas e capturadas a cada trecho de 1,5 metros de distância percorrida. Nestes *benchmarks* foi possível utilizar a informação do georreferenciamento como um *ground truth* para uma métrica de erro. Na Figura 5.25 é mostrado o trecho visualizado no *benchmark* denominado *New College* e na Figura 5.27 o trecho visualizado no *benchmark* denominado *City*. Além do *City* e do *New College*, foram utilizadas bases de teste do Karlsruhe Institute of Technology (GEIGER et al., 2013), sem vista aérea disponível, denominadas de Kitti20110929 e Kitti20110926. Para estes trechos inspecionados e utilizando as informações de georreferenciamento fornecidas foram produzidos curvas tipo Precision-Recall considerando diferentes variações na tolerância da localização espacial variando de 1.5 a 32 metros. A curva da Figura 5.26 apresenta o gráfico Precison-Recall para o trecho *New College* que é formado por 1984 imagens. A Figura 5.28 apresenta curvas de Precision-Recall para o trecho *City* formado por 2474 imagens utilizando diferentes configurações do sistema. Na Figura 5.29 são apresentadas as curvas da base Kitti20110929 e Kitti20110926.

**Figure 5.25:** Vista aérea do trecho da base de dados *New College*. (Imagem retirada da tese de doutorado de CUMMINS (2009) )

**Table 5.3:** Configuração dos mapas utilizados no banco de testes *New College*

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 1984 |
| Número de vetores SURF por frame: | 40 |
| Número mínimo de ativações no I-MAP para identificação do local: | 10 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 330 |
| Número de imagens utilizadas no conjunto de treinamento: | 330 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | ver Figura 5.26 |

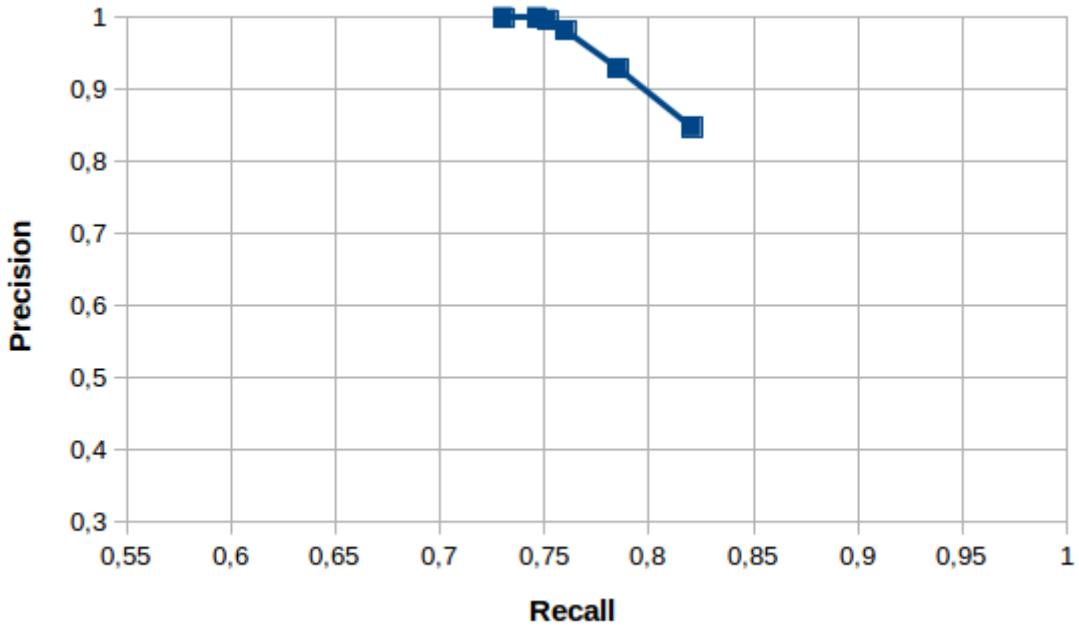


**Figure 5.26:** Curva Precision-Recall utilizando o modelo proposto com a base de dados *New College*.

**Figure 5.27:** Vista aérea do trecho da base da dados *City*. (Imagem retirada da tese de doutorado de CUMMINS (2009) )

**Table 5.4:** Configuração dos mapas utilizados no banco de testes *City*

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 2474 |
| Número de vetores SURF por frame: | 40 |
| Número mínimo de ativações no I-MAP para identificação do local: | 10 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 412 |
| Número de imagens utilizadas no conjunto de treinamento: | 412 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | ver Figura 5.28 |

Na curva Teste 1 da Figura 5.28 foram utilizados 5000 nodos no I-MAP e 1/6 do conjunto de imagens foi utilizado para realizar o treinamento do I-MAP. O Teste 2 foi realizado do mesmo modo que o Teste 1 só que utilizando 1/4 do conjunto de imagens para o treinamento. Em Teste 3, o I-MAP foi ajustado para crescer até 500 nodos e utilizar 1/6 das imagens para treinamento. Em Teste 4, foram utilizadas 4490 imagens da Avenida dos Reitores na UFPE para realizar a construção do I-MAP. Ou seja, o sistema utilizou mapa criado para armazenar a informação de um conjunto de treinamento completamente diferente do conjunto testado.

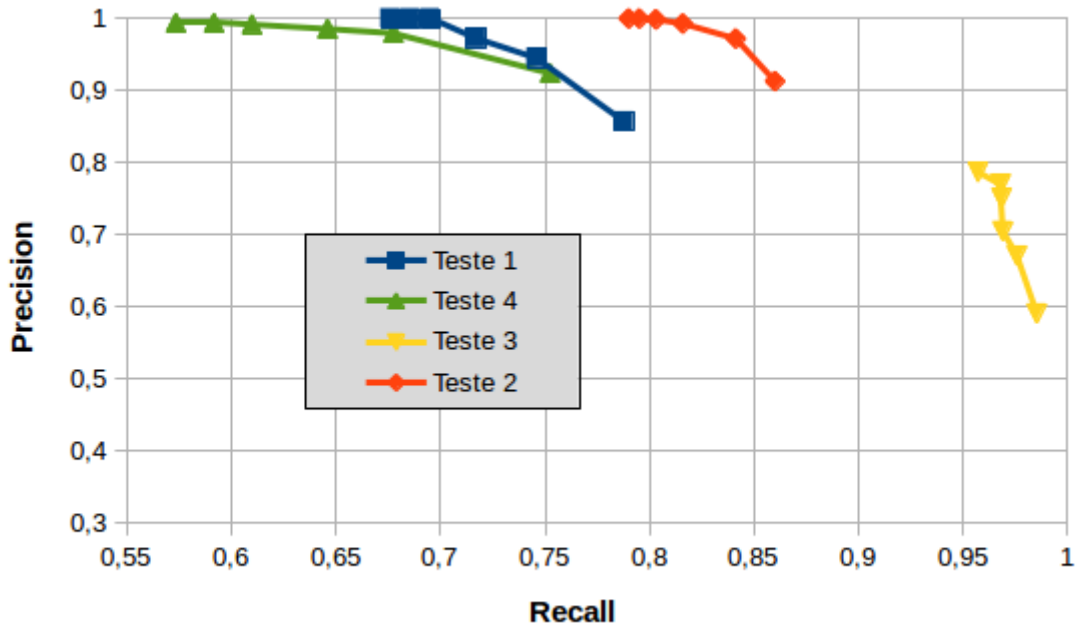


**Figure 5.28:** Curva Precision-Recall utilizando o modelo proposto com a base de dados City.

Abaixo seguem as Tabelas 5.5 e 5.6 com os parâmetros, respectivamente, do sistema utilizados nos testes com as bases Kitti20110926 e Kitti20110929.

**Table 5.5:** Configuração dos mapas utilizados no banco de testes Kitti20110926

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 474 |
| Número de vetores SURF por frame: | 40 |
| Número mínimo de ativações no I-MAP para identificação do local: | 10 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 79 |
| Número de imagens utilizadas no conjunto de treinamento: | 79 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | ver Figura 5.29 |

**Table 5.6:** Configuração dos mapas utilizados no banco de testes Kitti20110929

| **Parâmetros** | **Valor** |
|---|---|
| Número de frames avaliados: | 1058 |
| Número de vetores SURF por frame: | 40 |
| Número mínimo de ativações no I-MAP para identificação do local: | 10 |
| Número de nodos utilizado no I-MAP: | 5000 |
| Número de nodos no P-MAP: | 176 |
| Número de imagens utilizadas no conjunto de treinamento: | 176 |
| Tempo médio de execução por frame: | 54ms |
| Erros na localização: | ver Figura 5.29 |

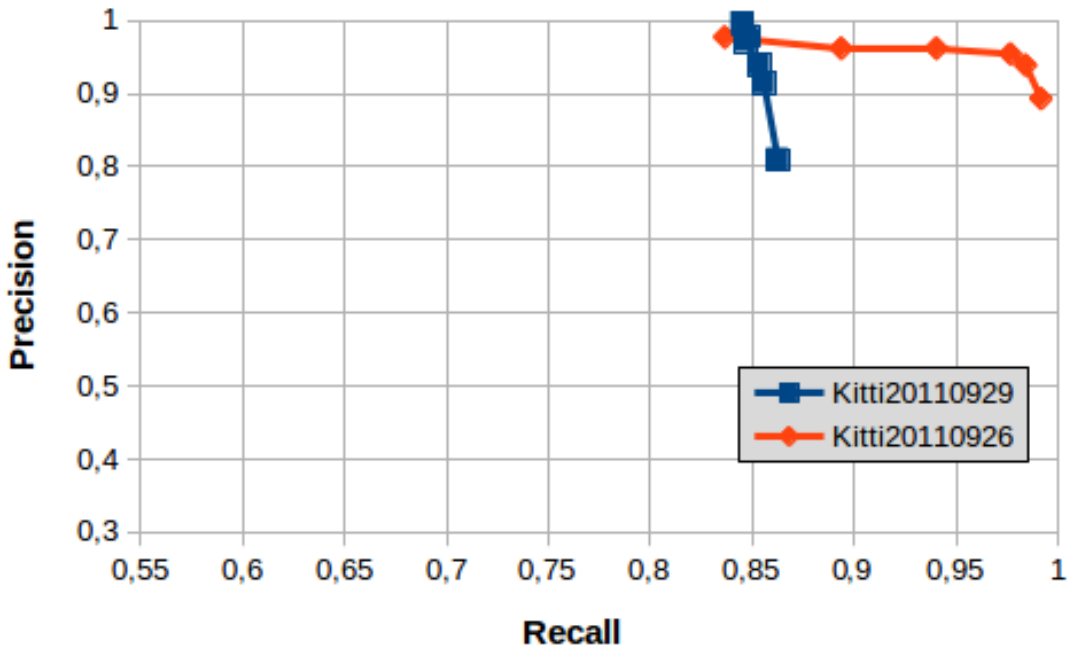


**Figure 5.29:** Curva Precision-Recall utilizando o modelo proposto com as bases de dados 2011-09-26 e 2011-09-29 do Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago (GEIGER et al., 2013).

Os parâmetros dos mapas I-MAP, como número de nodos ($N_{max}$) e os parâmetros de distância ($d_0$, $d_1$ e $d_2$), utilizados nos testes na UFPE e nos *benchmarks* foram os mesmos e baseados no desvio padrão médio dos vetores SURF dos testes realizados na UFPE.

Servindo como uma base de comparação, na Figura 5.30, é apresentado um gráfico com a curva Precision-Recall para um modelo proposto por CUMMINS (2009) para Slam visual em espaço de aparências. As bases de dados utilizada foram as *New College* e a *City*.

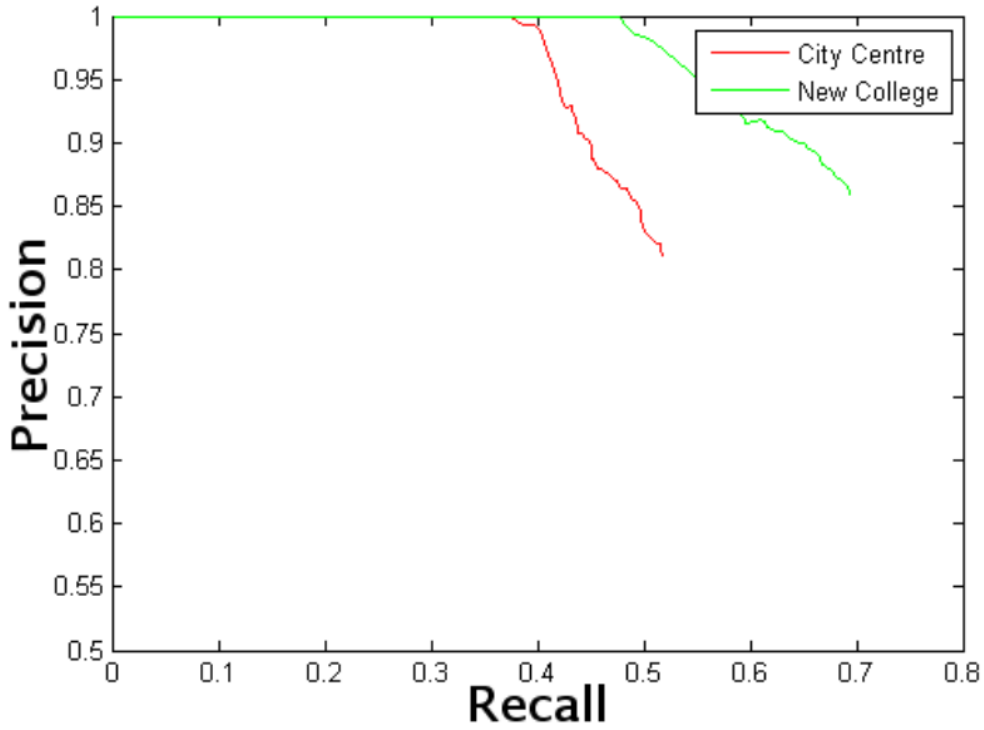


**Figure 5.30:** Como referência para comparação, resultados obtidos por CUMMINS (2009) utilizando os bancos *City* e *New College*. O gráfico foi retirado da Tese de Phd de Cummins.

### 5.4.3 Avaliação da Capacidade de Localização Visual e Mapeamento topológico

Na Figura 5.31 é apresentado o mapa topológico produzido pelo sistema de localização e mapeamento produzido a partir da inspeção em ambiente externo da UFPE, Figura 5.24. As cores dos nodos do grafo denotam o número de vezes que o sistema visitou o mesmo local, a rotulação numérica dos nodos corresponde ao número do frame utilizado como referencia para identificar o local visitado e a espessura das ligações denotam a frequência com que o agente transitou entre os nodos conectados pela ligação. É interessante notar na Figura 5.31 que os pontos iniciais e finais da trajetória inspecionada foram percorridos apenas uma vez e que o trecho central circular, na Av. dos Reitores, for percorrido 7 vezes. Por isso, os nodos iniciais e finais do mapa topológico só apresentam uma (início e fim das trajetores) ou dous conexões e estão sinalizados como visitados apenas uma vez (cor azul). É possível notar, também, que a parte central no mapa topológico apresenta a mesma forma circular da trajetória percorrida. Neste trecho, os nodos apresentam mais conexões formadas em decorrência de múltiplas passadas que não necessariamente seguiram exatamente os mesmos percurso. Na Figura 5.32, por exemplo, são exibidas as imagens da trajetória que originaram os nodos 601 e 8923. No mapa são nodos distintos e que foram inseridos em momentos não consecutivos, mas que estão conectados porque o sistema julgou o conteúdo das características visuais semelhante. Observando as imagens é possível notar que de fato são imagens de locais distintos, mas próximos fisicamente. Na Figura

5.31, a espessura das conexões é proporcional ao número de vezes que foram utilizadas. Desta forma é possível saber quais são os trechos com uma relação de vizinhança mais confiável ou que foi mais verificada durante a navegação.

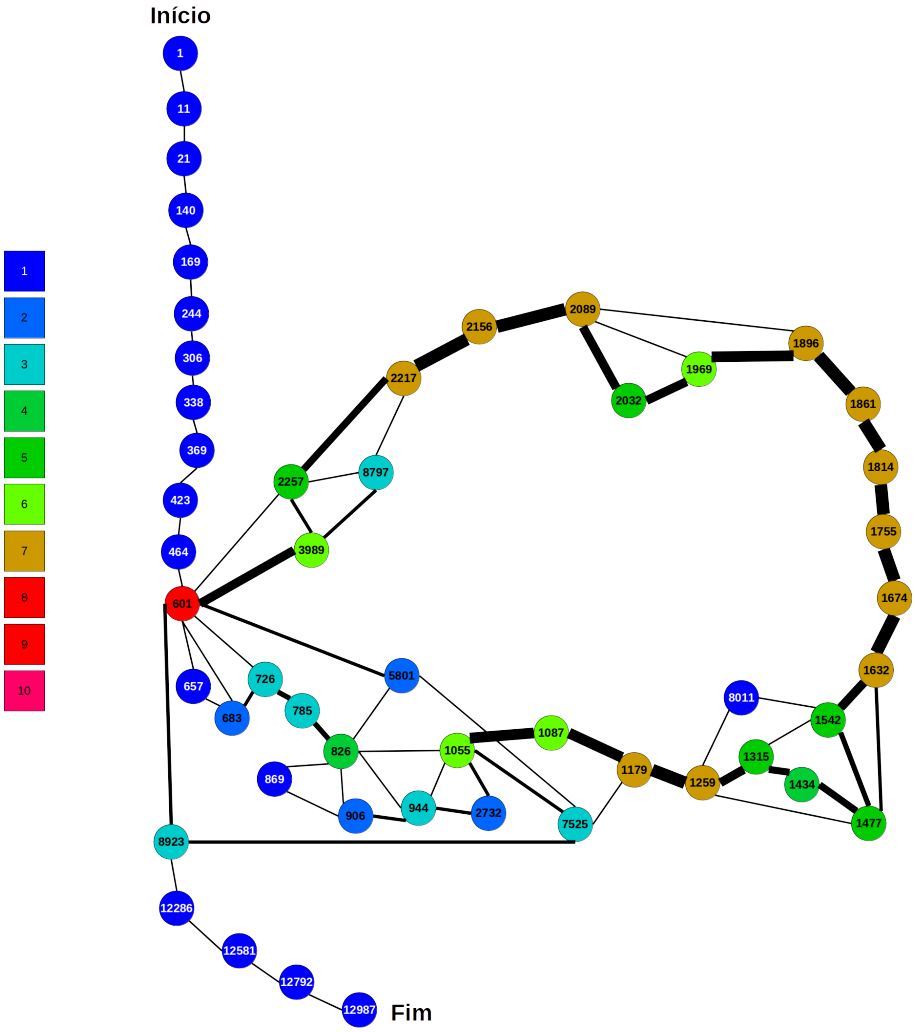


**Figure 5.31:** Vista do mapeamento topológico realizado pelo sistema de localização visual e mapeamento topológico proposto.

(a) Imagem nodo 601 (b) Imagem nodo 8923

**Figure 5.32:** Imagens associada aos nodo 601 (a) e 8923 (b) do mapa topológico da Figura 5.31

Na Figura 5.33, é apresentado o mapa topológico construído a partir do banco de teste *City*. Os pontos em azul, são os nodos do mapa que o sistema identificou como visitados apenas uma vez e os em vermelho são os que foram visitados duas vezes. O mapa construído apresentou *Precision* 100% e um *Recall* de 45.4% em relação a detecção de locais revisitados. Na Figura 5.33, todas as vezes que o sistema indicou um local como revisitado a classificação foi correta. Além disso, por inspeção visual é possível notar que as conexões topológicas do grafo estão coerentes com as posições espaciais dos nodos. Não há conexões de nodos que não são vizinhos próximos.

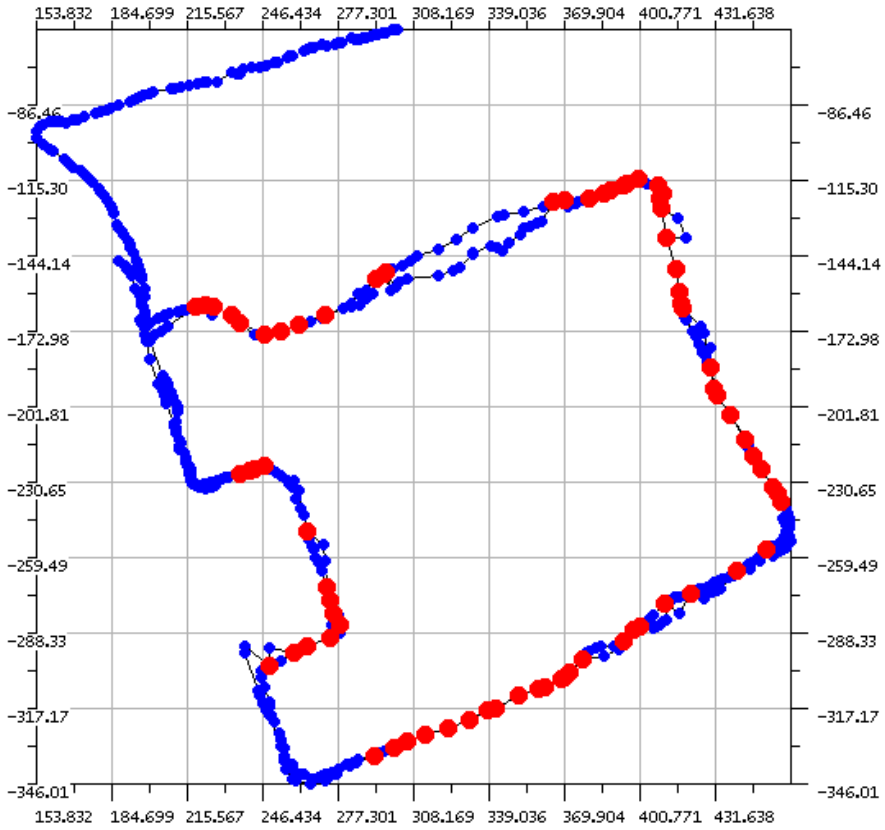


**Figure 5.33:** Mapa topológico construído a partir do banco de teste utilizando apenas as imagens monoculares e vetores de característica do banco *City*.

Na Figura 5.34, é apresentado o mapa topológico construído a partir do banco de teste *New College*. O padrão de cor é o mesmo utilizado para o mapa anterior, nodos azuis indicam uma visita e vermelhos duas. O mapa também apresentam *Precision* 100% e um *Recall* de 39.7% em relação a detecção de locais revisitados. Apresentando desempenho semelhante ao caso do banco *City*.

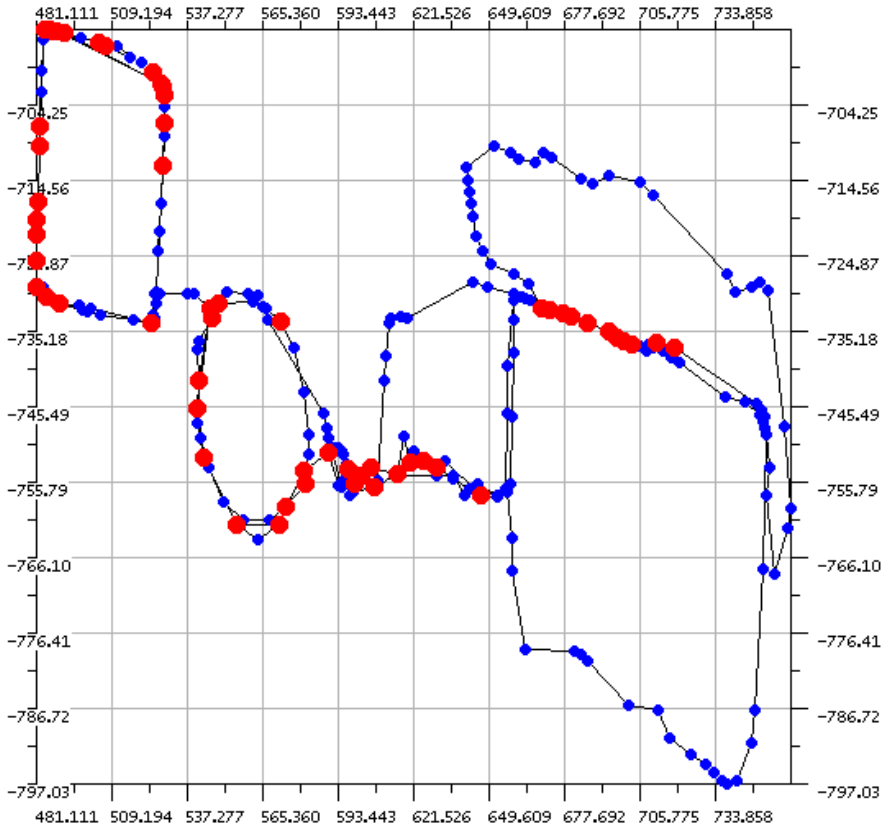


**Figure 5.34:** Mapa topológico construído a partir do banco de teste utilizando apenas as imagens monoculares e vetores de característica do banco *New College*.

Na Figura 5.35, é apresentado o mapa topológico construído a partir do banco *New College*, mas utilizando um dicionário visual treinado a partir do banco *City*. Ou seja, o I-MAP do sistema foi construído utilizando o banco *City* e o P-MAP construído com o banco *New College*. Este resultado apresentou uma performance menor que os casos anteriores, Figuras 5.33 e 5.34. As medidas de *Precison* e *Recall* obtidas, respectivamente, foram de 95.2% e 35.7% e foi possível notar um total de 5 ligações topológicas realizadas incorretamente. Entretanto, mostra claramente que não é necessário que o I-MAP seja construído com imagens do mesmo conjunto do inspecionado. Mostra que o I-MAP realmente tende a capturar microcaracterísticas gerais do banco de imagens de treinamento.

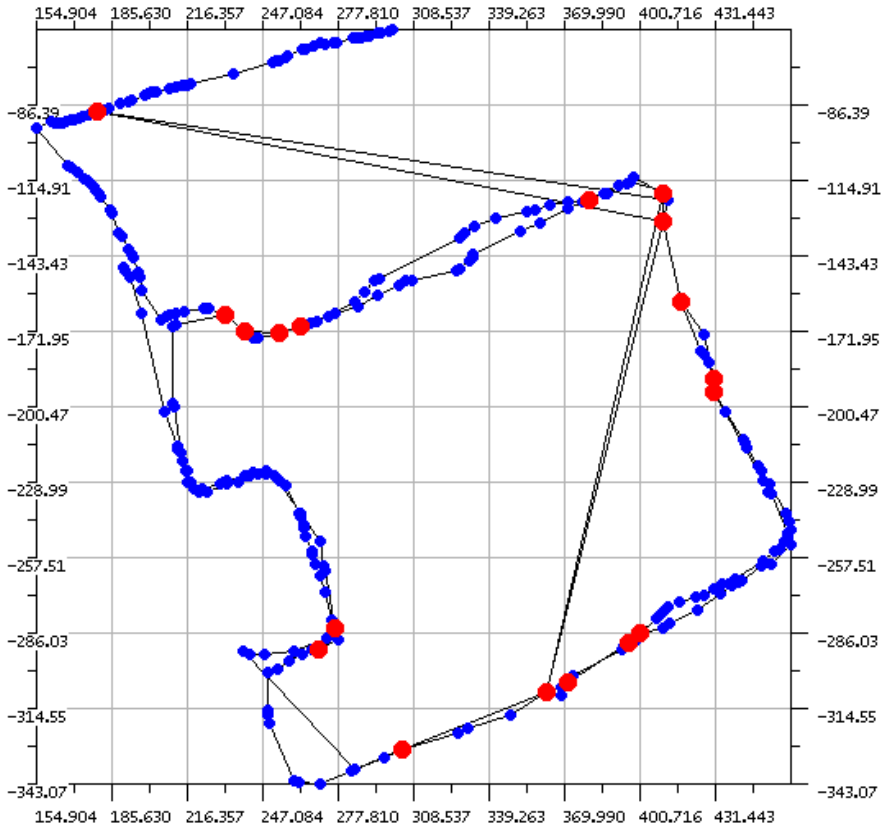


**Figure 5.35:** Mapa topológico construído a partir do banco de teste City utilizando apenas as imagens monoculares e vetores de característica do banco *New College*.

## 5.5 Discussões dos resultados

Neste capítulo foi apresentado um novo tipo de Mapa Auto-organizável com estrutura variante no tempo baseado em características dos mapas GNG e LARFSOM. O mapa foi proposto como uma ferramenta base para tratar problemas de redução de dimensionalidade e previsão temporal relacionados a área de SLAM Visual em Espaço de Aparências. Foram realizados testes e comparações ilustrativas do desempenho do mapa proposto com outros mapas de topologia fixa (SOM), topologia variável (GNG) e recursivos, como o RecSOM. Os testes consideraram problemas de agrupamento e previsão temporal. Os resultados das simulações sugerem que o FSS-SOM apresenta desempenho em problemas de agrupamento e previsão temporal compatível com o desempenho de um mapa GNG, mas com um número de interações menor.

O sistema completo de localização e mapeamento visual utilizando mapas FSS-SOM foi testado em relação a capacidade de se localizar visualmente e em realizar localização visual e mapeamento topológico simultâneo. Foram utilizados um total de três conjuntos

de dados diferentes: em ambiente interno sem georreferenciamento, em ambiente externo sem georreferenciamento e em ambiente externo com georreferenciamento. Os testes sem georreferenciamento foram realizados a partir de percurso realizados dentro do CIn e na Avenida dos Reitores da UFPE. Os testes realizados em ambiente externo com georreferenciamento foram realizados a partir de um conjunto de imagens e dados de GPS de bases de dados públicas (CUMMINS; NEWMAN, 2008b; GEIGER et al., 2013).

Nos teste de localização visual, os resultados foram apresentados em curvas Precison-Recall (Figuras 5.26, 5.28 e 5.29). As curvas Precision-Recall mostram, na maioria dos testes, foram obtidas taxas de Recall acima de 70% e precisão acima de 84%. Como um elemento de comparação, curvas Precision-Recall calculadas sobre as bases *City* e *New College* do trabalho de CUMMINS (2009) foram incluídas.

Nos testes de localização visual e mapeamento topológico os resultados foram apresentados na forma de mapas topológicos dos trechos inspecionados (Figuras 5.31, 5.33, 5.34 e 5.35). Os mapas produzidos são semelhantes a outras abordagem de mapeamento topológico (GARCIA-FIDALGO; ORTIZ, 2015).

Um dos principais intuitos deste trabalho foi propor um sistema inovador e factível utilizando Mapas Auto-organizáveis para resolver o problema de localização e mapeamento simultâneo em espaço de aparências. Esta é uma área dominada por abordagens estatísticas e que apresenta poucos modelos baseados em redes neurais. Os testes realizados com o modelo proposto não só mostram um sistema factível com Mapas Auto-organizáveis, mas também apresentam resultados com curvas e medidas *Precision*/*Recall* compatíveis com as melhores abordagens utilizando os *benchmarks* da área (GARCIA-FIDALGO; ORTIZ, 2015; CUMMINS, 2009).

# 6

# Conclusão

Este trabalho apresenta uma revisão bibliográfica sobre a área de navegação autônoma que cobre aproximadamente 20 anos de trabalhos e abordagens relevantes na área. Propõem e apresenta os resultados (D. F. GOMES, 2014), de um novo modelo de Mapa Auto-organizável de uso geral (Seção 4.2). O mapa proposto, FSS-SOM, apresenta como principais características crescimento rápido sobre o conjunto de treinamento e a possibilidade de controlar a distribuição mínima e máxima dos nodos conectados no mapa. Propõem, também, um novo sistema de localização visual e mapeamento topológico (Seção 4.3) utilizando o mapa desenvolvido. No Apêndice A, é apresentada uma plataforma robótica desenvolvida especialmente com a finalidade de servir de base de testes para o sistema de navegação e localização visual proposto neste trabalho. O Capítulo 5 apresenta os resultados do mapa FSS-SOM e do sistema de localização visual e mapeamento topológico utilizando o modelo proposto testado com seis tipos de base de dados de sequências de observações visuais, duas são sem georreferenciamento e quatro utilizam georreferenciamento.

Uma das motivações da tese foi oferecer solução e abordagem inovadora para o problema de localização e mapeamento simultâneos (SLAM) baseado em espaço de aparências. A maioria das soluções presentes na área se baseia fortemente em abordagens estatísticas, poucos são os modelos baseados redes neurais. Assim, o modelo proposto baseado em Mapas Auto-organizáveis é um esforço no sentido de demonstrar que o problema de SLAM Visual em Espaço de Aparências pode ser abordado de uma forma menos convencional.

Nos testes realizados o sistema proposto apresentou resultados equivalentes aos melhores modelos disponíveis atualmente em termos de curvas *Precision/Recall* de bases como *City* e *New College* (GARCIA-FIDALGO; ORTIZ, 2015). A geração de gráficos topológicos dos trechos inspecionados também apresentou alto grau de coerência. A partir dos testes realizados podem ser derivadas cinco conclusões sobre o mapa FSS-SOM e o modelo de localização e mapeamento:

- O Mapa Auto-organizável proposto, FSS-SOM, é capaz de realizar agrupamentos de dados de maneira similar ao mapa GNG, mas permitindo um controle sobre a formação e tamanho das conexões.

- O FSS-SOM apresentou uma convergência no seu treinamento, em número de épocas, menor que a de mapas como o de Kohonen, RecSOM e o GNG.
- O sistema proposto de localização visual e mapeamento topológico utilizando o FSS-SOM consegue construir mapas com a mesma topologia dos trechos inspecionados utilizando apenas informações visuais monoculares de uma câmera.
- O sistema utiliza uma métrica de semelhança baseada em conjuntos que evita a utilização de vetores esparso e que apresenta bons resultados para a classificação de imagens.
- Os resultados obtidos com as curvas e medidas de *Precision* e *Recall* do sistema utilizando os *benchmarks* sugerem que o sistema apresenta um desempenho no mínimo equivalente a outras abordagem que não utilizam Mapas Auto-organizáveis na sua solução.

## 6.1 Contribuições para a Ciência

- Nesta tese é proposto e testado um novo modelo de Mapa Auto-organizável, FSS-SOM, que apresenta um crescimento rápido sobre os conjuntos de treinamento e oferece a possibilidade de ter sua distribuição de nodos conectados limitada a uma densidade mínima e máxima.
- Um novo sistema de localização visual e mapeamento topológico baseado em Mapas Auto-organizáveis foi proposto, implementado e testado. O sistema desenvolvido apresenta alta taxa de acerto e apresenta um tempo de resposta pequeno. Apresenta um mecanismo local de busca e identificação de locais visitados, utilizando uma interação entre um Mapas Auto-organizáveis(I-MAP) e um mapa topológico(P-MAP), respectivamente, informação visual e de localidade, as buscas na memória visual do I-MAP podem ser limitadas pela estimativa do local dado pelo P-MAP possibilitando identificação de locais em tempos reduzidos, mesmo considerando grandes percursos inspecionados.

## 6.2 Trabalhos Futuros

O sistema atual de localização visual e mapeamento topológico proposto nesta tese utilizou como extrator de características salientes de um imagem o descritor SURF. Entretanto, uma possível linha a ser seguida nos próximos trabalhos é avaliar o desempenho do sistema utilizando outros tipos de descritor. Em especial, o descritor DIRD, proposto especificamente para SLAM Visual (LATEGAHN; BECK;

STILLER, 2014), parece ser uma opção interessante para testes futuros. A principal vantagem deste descritor sobre o SURF é presentar uma maior robustez a variações de sombra e luminosidade.

A plataforma robótica apresentada no Apêndice A é apenas uma versão inicial de um sistema de baixo custo que pode ser utilizado em testes de algoritmos e sistemas de navegação em pequena escala e em ambientes *indoor* e ainda pode ser bastante aperfeiçoada. A intensão é tornar esta plataforma opensource, com os projetos elétricos e o software disponíveis na internet para que possa ser utilizada como opção para teste padronizados de sistemas de navegação.

# Referências

ARAUJO, A. F. R.; REGO, R. L. M. E. Self-organizing Maps with a Time-varying Structure. **ACM Comput. Surv.**, New York, NY, USA, v.46, n.1, p.7:1–7:38, July 2013.

ARAúJO, A. R.; COSTA, D. C. Local adaptive receptive field self-organizing map for image color segmentation. **Image and Vision Computing**, [S.l.], v.27, n.9, p.1229 – 1239, 2009.

ARKIN, R. **BEHAVIOR-BASED ROBOTICS**. [S.l.]: Cambridge, MA: MIT Press, 1998.

ARLEO, A.; GERSTNER, W. **Modeling Rodent Head-direction Cells and Place Cells for Spatial Learning in Biomimetic Robotics**. 2000.

AUSTRALIAN GOVERNMENT, B. of Meteorology of the. **Southern Oscillation Index**. [Online; accessed 02-09-2014], `http://www.bom.gov.au/climate/glossary/soi.shtml`.

AUSTRALIAN GOVERNMENT, F. S. of Bureau of Meteorology of the. **Southern Oscillation Index Text Dataset**. [Online; accessed 02-09-2014], `ftp://ftp.bom.gov.au/anon/home/ncc/www/sco/soi/soiplaintext.html`.

BAY H. ESS, A. T. T. V. G. L. SURF: speeded up robust features. **Computer Vision and Image Understanding (CVIU)**, [S.l.], v.110, n.3, p.346–359, 2008.

BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual Navigation for Mobile Robots: a survey. **J. Intell. Robotics Syst.**, Hingham, MA, USA, v.53, n.3, p.263–296, nov 2008.

BUEHLER, M.; IAGNEMMA, K.; SINGH, S. **The DARPA Urban Challenge**. [S.l.]: Springer, 2009. (Springer Tracts in Advanced Robotics).

CABALLERO, F. et al. Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. **JOURNAL OF INTELLIGENT & ROBOTIC SYSTEMS**, VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS, v.54, n.1-3, p.137–161, MAR 2009. International Symposium on Unmanned Aerial Vehicles, Orlando, FL, JUN 23-25, 2008.

CASTELNOVI, M.; ARKIN, R.; COLLINS, T. Reactive speed control system based on terrain roughness detection. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), VOLS 1-4, 2005., 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2005. p.891–896. (IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION). IEEE International Conference on Robotics and Automation (ICRA), Barcelona, SPAIN, APR 18-22, 2005.

CHERIF, A.; CARDOT, H.; BONé, R. SOM time series clustering and prediction with recurrent neural networks. **Neurocomputing**, [S.l.], v.74, n.11, p.1936 – 1944, 2011. Adaptive Incremental Learning in Neural Networks Learning Algorithm and Mathematic Modelling Selected papers from the International Conference on Neural Information Processing 2009 (ICONIP 2009).

CHOSET, H. et al. **Principles of Robot Motion**: theory, algorithm, and implementation. [S.l.]: Bradford Book, 2005.

COSTA, D. C. **Mapa Auto-organizável com Campo Receptivo Adaptativo Local para Segmentação de Imagens**. 2007. Dissertação (Mestrado em Ciência da Computação) — Universidade Federal de Pernambuco Centro de Informática Pós-graduação em Ciência da Computação.

CUMMINS, M. **Probabilistic Localization and Mapping in Appearance Space**. 2009. PhD thesis — Robotics Research Group, Department of Engineering Science University of Oxford.

CUMMINS, M.; NEWMAN, P. Probabilistic Appearance Based Navigation and Loop Closing. In: ROBOTICS AND AUTOMATION, 2007 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2007. p.2042–2048.

CUMMINS, M.; NEWMAN, P. FAB-MAP: probabilistic localization and mapping in the space of appearance. **The International Journal of Robotics Research**, [S.l.], v.27, n.6, p.647–665, 2008.

CUMMINS, M.; NEWMAN, P. **Multimedia Extensions for FAB-MAP**: probabilistic localization and mapping in the space of appearance. International Journal of Robotics Research 2008 Dataset, Internet Site.

CUMMINS, M.; NEWMAN, P. Appearance-only SLAM at large scale with FAB-MAP 2.0. **The International Journal of Robotics Research**, [S.l.], v.30, n.9, p.1100–1123, 2011.

D. F. GOMES, A. F. R. A. Fast Coverage of N-Dimensional State Space Self-Organizing Map. In: INTELLIGENT SYSTEMS (BRACIS), 2014 BRAZILIAN CONFERENCE ON. **Anais...** [S.l.: s.n.], 2014. p.348–353.

DAUM, F. Nonlinear filters: beyond the kalman filter. **Aerospace and Electronic Systems Magazine, IEEE**, [S.l.], v.20, n.8, p.57–69, Aug 2005.

DAWOOD, M. et al. Harris, SIFT and SURF features comparison for vehicle localization based on virtual 3D model and camera. In: IMAGE PROCESSING THEORY, TOOLS AND APPLICATIONS (IPTA), 2012 3RD INTERNATIONAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2012. p.307–312.

DESOUZA, G.; KAK, A. Vision for mobile robot navigation: a survey. **IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE**, 10662 LOS VAQUEROS CIRCLE, PO BOX 3014, LOS ALAMITOS, CA 90720-1314 USA, v.24, n.2, p.237–267, FEB 2002.

DICKMANNS, E. et al. The seeing passenger car 'VaMoRs-P'. In: INTELLIGENT VEHICLES '94 SYMPOSIUM, PROCEEDINGS OF THE. **Anais...** [S.l.: s.n.], 1994. p.68–73.

DISSANAYAKE, M. et al. A solution to the simultaneous localization and map building (SLAM) problem. **IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION**, 345 E 47TH ST, NEW YORK, NY 10017-2394 USA, v.17, n.3, p.229–241, JUN 2001.

DOUCET, A. et al. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, 16., San Francisco, CA, USA. **Proceedings...** Morgan Kaufmann Publishers Inc., 2000. p.176–183. (UAI '00).

EURONCAP. **Lane-Assist**. [Online; accessed 05-01-2015], `http://www.euroncap.com/rewards/vw_lane_assist.aspx`.

FILLIAT, D. A visual bag of words method for interactive qualitative localization and mapping. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION. **Anais...** [S.l.: s.n.], 2007. p.3921–3926.

FRITZKE, B. A Growing Neural Gas Network Learns Topologies. In: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 7. **Anais...** MIT Press, 1995. p.625–632.

FUENTES-PACHECO, J.; RUIZ-ASCENCIO, J.; MANUEL RENDON-MANCHA, J. Visual simultaneous localization and mapping: a survey. **ARTIFICIAL INTELLIGENCE REVIEW**, VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS, v.43, n.1, p.55–81, JAN 2015.

GALINDO, C. et al. Multi-hierarchical semantic maps for mobile robotics. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, VOLS 1-4, 2005., 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2005. p.3492–3497. IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, CANADA, AUG 02-06, 2005.

GARCIA-FIDALGO, E.; ORTIZ, A. Vision-based topological mapping and localization methods: a survey. **Robotics and Autonomous Systems**, [S.l.], v.64, n.0, p.1 – 20, 2015.

GEIGER, A. et al. Vision meets Robotics: the kitti dataset. **International Journal of Robotics Research (IJRR)**, [S.l.], 2013.

GIBBENS, P.; DISSANAYAKE, G.; DURRANT-WHYTE, H. A closed form solution to the single degree of freedom simultaneous localisation and map building (SLAM) problem. In: PROCEEDINGS OF THE 39TH IEEE CONFERENCE ON DECISION AND CONTROL, VOLS 1-5, 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2000. p.191–196. (IEEE CONFERENCE ON DECISION AND CONTROL - PROCEEDINGS). 39th IEEE Conference on Decision and Control, SYDNEY, AUSTRALIA, DEC 12-15, 2000.

GLOVER, A. et al. OpenFABMAP: an open source toolbox for appearance-based loop closure detection. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2012. p.4730–4735. IEEE International Conference on Robotics and Automation (ICRA), St Paul, MN, MAY 14-18, 2012.

GOMES, L. **Hidden Obstacles for Google's Self-Driving Cars**. [Online; accessed 12-01-2015], `http://www.technologyreview.com/news/530276/hidden-obstacles-for-googles-self-driving-cars/`.

GONZALEZ, R. C.; WOODS, R. C. **Processamento Digital de Imagens**. 3..ed. [S.l.]: Pearson Prentice Hall, 2010.

GOOIJER, J. G. D.; HYNDMAN, R. J. 25 years of time series forecasting. **International Journal of Forecasting**, [S.l.], v.22, n.3, p.443 – 473, 2006. Twenty five years of forecasting.

GRAEFE, V. Vision for intelligent road vehicles. In: INTELLIGENT VEHICLES SYMPOSIUM (1993: TOKYO, JAPAN). PROCEEDINGS OF THE INTELLIGENT VEHICLES' 93 SYMPOSIUM. **Anais...** [S.l.: s.n.], 1993.

GRAEFE, V.; KUHNERT, K.-D. Vision-based autonomous road vehicles. In: **Vision-based Vehicle Guidance**. [S.l.]: Springer, 1992. p.1–29.

GUERRA, E.; MUNGUIA, R.; GRAU, A. Monocular SLAM for Autonomous Robots with Enhanced Features Initialization. **SENSORS**, POSTFACH, CH-4005 BASEL, SWITZERLAND, v.14, n.4, p.6317–6337, APR 2014.

PRENTICE-HALL (Ed.). **Neural Networks, A Comprehensive Foundation**. 2nd.ed. [S.l.]: Pearson Prentice Hall, 1999.

HELTON, J.; DAVIS, F.; JOHNSON, J. A comparison of uncertainty and sensitivity analysis results obtained with random and Latin hypercube sampling. **Reliability Engineering & System Safety**, [S.l.], v.89, n.3, p.305 – 330, 2005.

HOPFIELD, J. J. Neural Networks and Physical Systems With Emergent Collective Computational Abilities. **Proceedings of the National Academy of Sciences of the United States of America-biological Sciences**, [S.l.], v.79, n.8, p.2554–2558, 1982.

KOHONEN, T. The Self-organizing Maps. **Proceedings of the IEEE**, 345 E 47TH ST, NEW YORK, NY 10017-2394, v.78, n.9, p.1464–1480, SEP 1990.

KRIEGMAN D. TRIENDL, E. B. T. STEREO VISION AND NAVIGATION IN BUILDINGS FOR MOBILE ROBOTS. **IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION**, 345 E 47TH ST, NEW YORK, NY 10017-2394, v.5, n.6, p.792–803, DEC 1989.

LATEGAHN, H.; BECK, J.; STILLER, C. DIRD is an illumination robust descriptor. In: INTELLIGENT VEHICLES SYMPOSIUM PROCEEDINGS, 2014 IEEE. **Anais...** [S.l.: s.n.], 2014. p.756–761.

MACKEY, M. C.; GLASS, L. et al. Oscillation and chaos in physiological control systems. **Science**, [S.l.], v.197, n.4300, p.287–289, 1977.

MANDUCHI, R. et al. Obstacle detection and terrain classification for autonomous off-road navigation. **AUTONOMOUS ROBOTS**, VAN GODEWIJCKSTRAAT 30, 3311 GZ DORDRECHT, NETHERLANDS, v.18, n.1, p.81–102, JAN 2005.

MARTINETZ T., B. S. G. S. K. J. "Neural-Gas" Network for Vector Quantization and its Application to Time-Series Prediction. **IEEE Transactions on Neural Networks**, [S.l.], v.4, n.4, p.558–569, 1993. IEEE Press, New York.

MILFORD, M.; WYETH, G.; PRASSER, D. RatSLAM: a hippocampal model for simultaneous localization and mapping. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, VOLS 1- 5, PROCEEDINGS, 2004., 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2004. p.403–408. (IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION). IEEE International Conference on Robotics and Automation, New Orleans, LA, APR 26-MAY 01, 2004.

MILFORD, M.; WYETH, G.; PRASSER, D. Efficient goal directed navigation using RatSLAM. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), VOLS 1-4, 2005., 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2005. p.1097–1102. (IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION). IEEE International Conference on Robotics and Automation (ICRA), Barcelona, SPAIN, APR 18-22, 2005.

MILFORD, M.; WYETH, G.; PRASSER, D. **RatSLAM on the edge**: revealing a coherent representation from an overloaded rat brain. 2006. 4060–4065p.

MOLE, V. L. D. **Mapas Auto-Organizáveis para Agentes Robóticos Autônomos**. 2010. PhD thesis — Centro de Informática-UFPE.

MONTEMERLO, M. et al. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: EIGHTEENTH NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE (AAAI-02)/FOURTEENTH INNOVATIVE APPLICATIONS OF ARTIFICIAL INTELLIGENCE CONFERENCE (IAAI-02), PROCEEDINGS, FIVE CAMBRIDGE CENTER, CAMBRIDGE, MA 02142 USA. **Proceedings Paper...** M I T PRESS, 2002. p.593–598. 18th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence, EDMONTON, CANADA, JUL 28-AUG 01, 2002.

MURALI, V. N.; BIRCHFIELD, S. T. Autonomous exploration using rapid perception of low-resolution image information. **Autonomous Robots**, [S.l.], v.32, n.2, p.115–128, Feb. 2012.

DOUCET, A.; FREITAS, N.; GORDON, N. (Ed.). **Sequential Monte Carlo Methods in Pratice**. [S.l.]: Springer, 2001.

RAMANAN, A.; NIRANJAN, M. A review of codebook models in patch-based visual object recognition. **Journal of Signal Processing Systems**, [S.l.], v.68, n.3, p.333–352, 2012.

REBAI, K.; AZOUAOUI, O.; ACHOUR, N. Fuzzy ART-based place recognition for visual loop closure detection. **BIOLOGICAL CYBERNETICS**, 233 SPRING ST, NEW YORK, NY 10013 USA, v.107, n.2, p.247–259, APR 2013.

ROMERO, R. A. F. et al. **Robótica Móvel**. [S.l.]: Livros Técnicos e Científicos Editora Ltda (LTC)., 2014.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning Representations By Back-propagating Errors. **Nature**, [S.l.], v.323, n.6088, p.533–536, Oct. 1986.

RYAN, C. **Nevada issues Google first license for self-driving car**. 2012.

SALTELLI A., C. K. S. E. M. **Sensitivity Analysis**. [S.l.]: Wiley, West Sucssex (2009), 2009.

SHEN, J.; HU, H. Visual based Localization for mobile robots with Support Vector Machines. In: IEEE INDUSTRIAL ELECTRONICS, IECON 2006 - 32ND ANNUAL CONFERENCE ON. **Anais...** [S.l.: s.n.], 2006. p.4176–4181.

SIEGWART, R. (Ed.). **Introduction to Autonomous Mobile Robots**. 2nd.ed. Cambridge, Massachusetts: Cambridge, MA: MIT Press, 2011. (Intelligent Robotics and autonomous agents series).

SMITH, B. W. AUTOMATED VEHICLES ARE PROBABLY LEGAL IN THE UNITED STATES. **The Center for Internet and Society at Stanford Law School**, [S.l.], November 2012.

SMITH, R. C.; CHEESEMAN, P. On the Representation and Estimation of Spatial Uncertainly. **Int. J. Rob. Res.**, Thousand Oaks, CA, USA, v.5, n.4, p.56–68, Dec. 1986.

SMITH, T. **The Robot Operating System**. [Online; accessed 01-07-2015], http://www.ros.org/about-ros/.

STRASDAT, H.; MONTIEL, J.; DAVISON, A. J. Visual SLAM: why filter? **Image and Vision Computing**, [S.l.], v.30, n.2, p.65 – 77, 2012.

STRASDAT, H.; MONTIEL, J. M. M.; DAVISON, A. J. Real-time Monocular SLAM: why filter? In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2010. p.2657–2664. (IEEE International Conference on Robotics and Automation ICRA). IEEE International Conference on Robotics and Automation (ICRA), Anchorage, AK, MAY 03-08, 2010.

STRICKERT, M.; HAMMER, B. Merge SOM for temporal data. **Neurocomputing**, [S.l.], v.64, n.0, p.39 – 71, 2005. Trends in Neurocomputing: 12th European Symposium on Artificial Neural Networks 2004.

THRUN, S. **Google's Driverless Car**. [Online; accessed 07-01-2015], http://www.huffingtonpost.com/2011/12/06/self-driving-car_n_1130568.html.

THRUN, S. **Stanford University**. [Online; accessed 01-07-2015], http://robots.stanford.edu/.

THRUN, S. et al. Stanley: the robot that won the darpa grand challenge. **JOURNAL OF FIELD ROBOTICS**, 111 RIVER ST, HOBOKEN, NJ 07030 USA, v.23, n.9, p.661–692, SEP 2006.

THRUN, S. et al. Stanley: the robot that won the darpa grand challenge. **Journal of Field Robotics**, [S.l.], v.23, n.9, p.661–692, 2006.

THRUN. SEBASTIAN BURGARD. WOLFRAM, F. D. **Probabilistic robotics**. [S.l.]: Cambridge, MA: MIT Press, 2006. (Intelligent robotics and autonomous agents series).

TORII, A.; SIVIC, J.; PAJDLA, T. Visual localization by linear combination of image descriptors. In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCV WORKSHOPS), 2011., 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2011. IEEE International Conference on Computer Vision (ICCV), Barcelona, SPAIN, NOV 06-13, 2011.

URMSON, C. et al. **Tartan Racing**: a multi-modal approach to the darpa urban challenge. http://archive.darpa.mil/grandchallenge/: Robotics Institute, 2007. (CMU-RI-TR-).

VOEGTLIN, T. Recursive self-organizing maps. **Neural Networks**, [S.l.], v.15, n.8–9, p.979 – 991, 2002.

WANG, J.; CIPOLLA, R.; ZHA, H. Vision-based global localization using a visual vocabulary. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION (ICRA), 345 E 47TH ST, NEW YORK, NY 10017 USA. **Proceedings Paper...** IEEE, 2005. p.4230–4235. (IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, v.1-4). IEEE International Conference on Robotics and Automation (ICRA), Barcelona, SPAIN, APR 18-22, 2005.

WHISHAW, I. Q.; HINES, D. J.; WALLACE, D. G. Dead reckoning (path integration) requires the hippocampal formation: evidence from spontaneous exploration and spatial learning tasks in light (allothetic) and dark (idiothetic) tests. **Behavioural Brain Research**, [S.l.], v.127, n.1-2, p.49 – 69, 2001.

# Apêndice

# A

# Plataforma Robótica

## A.1 Plataforma Robótica

Para se obter resultados experimentais consistentes, sistemas de localização visual e mapeamento necessitam de entradas visuais realistas, ambientes detalhados e dinâmicos e estas características são de difícil implementação nos simuladores e ambientes virtuais disponíveis atualmente. Deste modo, para testar a proposição inicial deste trabalho foi implementada uma plataforma robótica, Figuras A.7, A.8 e A.9, que pode ser controlada remotamente via joystick ou por um programa desenvolvido em linguagem c/c++.

O agende robótico desenvolvido é um conjunto de hardware e interface de software que pode ser utilizado para testes de sistemas inteligentes e de navegação robótica. A plataforma de hardware é centrada em uma placa de um Arduino Mega e um conjunto de 3 servo motores. Dois servo motores são utilizados para a movimentação da plataforma e um terceiro pode ser utilizado para a realização da movimentação de algum aparato sensorial, por exemplo, um sistema de sonar.

O sistema possui uma interface de software desenvolvida em c++ que pode ser integrada a qualquer programa em c/c++ para controlar remotamente o hardware do robô. A única dependência do sistema é a lib pthread, padrão em sistemas baseados no GNU/Linux.

O objetivo principal foi obter uma plataforma robótica para testes em pequena escala que não necessitasse bibliotecas com muitas dependências ou de um sistema operacional específico, como o ROS (SMITH, 2015). Assim, um usuário interessado em realizar pequenos testes com rotinas de navegação e controle pode utilizar esta plataforma diretamente a partir do computador desktop utilizado no desenvolvimento dos software. Não há a necessidade da instalação de muitas bibliotecas, de computadores adicionais para embarcar na plataforma ou recompilar os programas desenvolvidos para um outro sistema operacional ou CPU específica para o agente robótico. Além disto, todos os componentes utilizados no desenvolvimento são de baixo custo e facilmente encontrados em sites de comercio eletrônico.

Um diagrama esquemático do sistema robótico é apresentado na Figura A.1. Os comandos enviados do computador para a plataforma móvel podem utilizar interface USB ou Bluetooth.

A comunicação é estabelecida utilizando um microprotocolo desenvolvido especificamente para esta plataforma. O tempo de latência na execução dos comandos enviados a partir do computador podem variar entre 300uS e 5mS dependendo do tipo de comando executado.

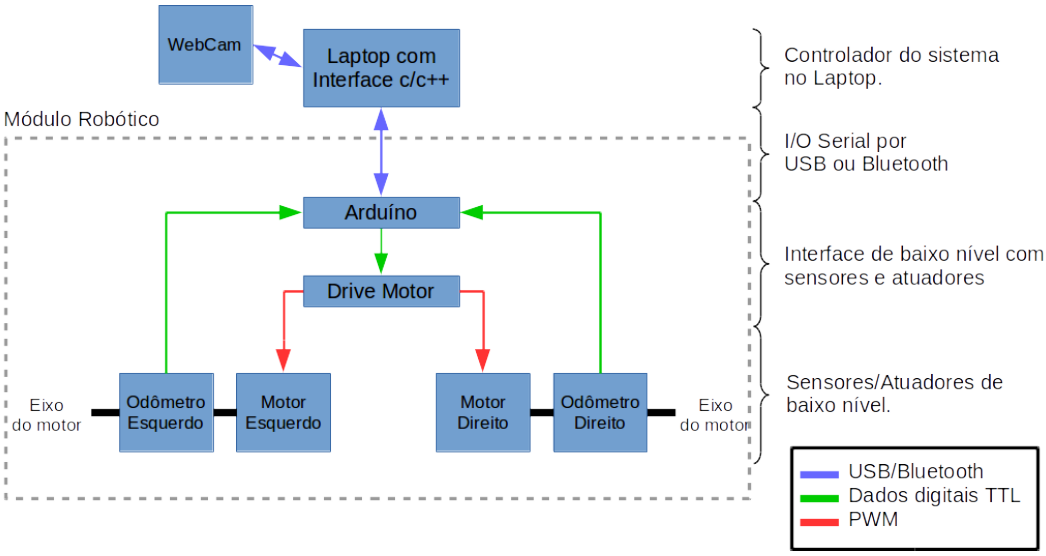


**Figure A.1:** Diagrama do sistema robótico

O sistema é modular e pode ser facilmente adaptado para a utilização de diversos tipos de sensores e atuadores. Na Figura A.2, há uma representação esquemática de como os sensores e atuadores da plataforma robótica estão mapeados na interface de controle de sensores e atuadores do sistema. A plataforma robótica é controlada/monitorada através de escritas e leituras nesta estrutura de memória utilizando um microprotocolo e uma comunicação serial entre o PC e a plataforma. O microprotocolo utilizado está descrito na seção A.2 e na seção A.3 há imagens da plataforma montada utilizando comunicação serial por Bluetooth com um laptop(não mostrado nas imagens).

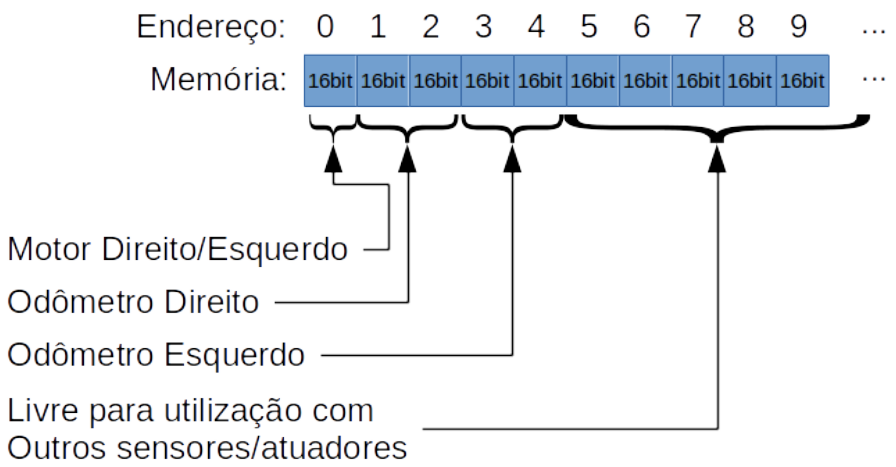


**Figure A.2:** Mapeamento da memória do sistema robótico

A codificação para o acionamento de um dos motores da plataforma robótica está representada esquematicamente na Figura A.3.

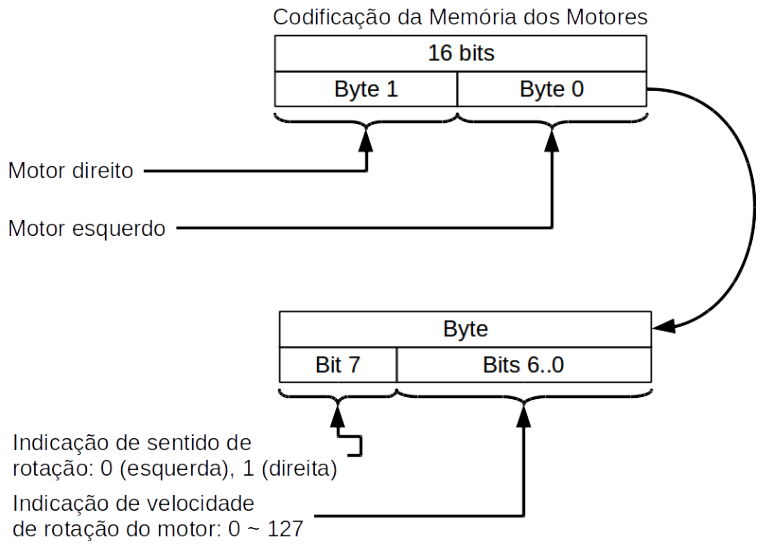


**Figure A.3:** Codificação do acionamento dos motores

A primeira posição da memória do sistema é utilizada para armazenar as informações sobre o estado de funcionamento do motor esquerdo e o direito. Como mostrado na Figura A.3, o byte mais significativo ($Byte_1$) da memória de 16bits, armazena o estado do motor direito e o byte menos significativo ($Byte_0$) o estado do motor esquerdo. A opção de armazenar os estados dos motores em uma mesma posição de memória esta relacionado com o tamanho do campo de dados do micropotocolo utilizado, que possui 16bits por pacote. Assim, para garantir que os motores do sistema sejam acionados simultaneamente e que o protocolo seja utilizado com eficiência, a codificação dos motores foi feita de forma a utilizar apenas 16bits ou um pacote. Para cada um dos bytes $Byte_1$ e $Byte_0$, o bit 7 ,mais significativo, é utilizado para indicar a direção de rotação do motor e os bit 6 ao 0 para codificar o ciclo ativo do sinal de PWM enviado para o motor. Os motores utilizados no sistema, por simplicidade, são motores DC. Assim, o sinal do ciclo ativo do PWM funciona como um tipo de controle de velocidade dos motores, quando maior o ciclo ativo mais rápido os motores irão giram.

## A.2 Protocolo de comunicação PC-Plataforma Robótica

O microprotocolo foi inspirado no protocolo Modbus. Existem implementações já prontas do Modbus para a plataforma Arduino, mas as implementações disponíveis apresentam um tempo de comunicação elevado, aproximadamente 200ms, para os propósitos desta plataforma robótica. O protocolo desenvolvido visa permitir comunicações rápidas de um ou dois bytes utilizando correção de erro e verificação de integridade. São utilizados três tipos básicos de pacotes de dados com tamanhos distintos, variando de 1 a 4 bytes (Figuras A.4,A.5 e A.6 ). A arquitetura

é do tipo mestre escravo ponto-a-ponto. Onde o dispositivo mestre é um computador PC e o escravo uma placa Arduino. O procedimento de comunicação é sempre inicializado pelo dispositivo mestre que pode, essencialmente, ler ou gravar em um determinado endereço de memória do Arduino. Esses endereços de memória são mapeados nos pinos de entrada e saída do Arduino, permitindo o comando de dispositivos de baixo nível, como: motores e sensores.

Um procedimento de escrita envolve o envio de um pacote e o recebimento de um pacote de confirmação. O pacote enviado pelo mestre é do tipo 1 formado por 4 bytes, Figura A.4, onde o *start_byte* é o dígito $0xF1$, o campo de endereço é um número binário de 0 a 10 que indica qual a posição da memória que será gravada e os campos de dados são preenchidos com um número binário de 16bits a ser gravado. O mestre aguarda por até $\Delta t$ milissegundos por um pacote de confirmação de recebimento do tipo 3 de 1 byte com o valor $0xF7$. Caso a confirmação de gravação não seja recebida, o mestre volta a enviar o pacote de escrita. Este procedimento pode ser repetido por até $N$ vezes antes de ser gerada uma exceção de falha na comunicação no dispositivo mestre. Na implementação o $\Delta t$ utilizado foi de 100ms e $N = 5$.

Para ler uma posição de memória no dispositivo escravo, o dispositivo mestre envia um pacote do tipo 2, Figura A.5, com *start_byte* igual a $0xF3$ e com o endereço da memória que se deseja ler. O mestre aguarda por até $\Delta t$ milissegundos pelo recebimento de um pacote tipo 1 enviado pelo dispositivo escravo com *start_byte* também igual a $0xF3$ e com o endereço igual ao solicitado e com os campos de dados preenchidos. Caso o mestre não receba nenhuma resposta do dispositivo escravo, o procedimento é reiniciado por até $N$ tentativas. Se não houver uma resposta, é gerada uma exceção por falha de comunicação no dispositivo mestre. Assim como no procedimento de escrita, na implementação foram utilizados os valores de $\Delta t = 100ms$ e $N = 5$.

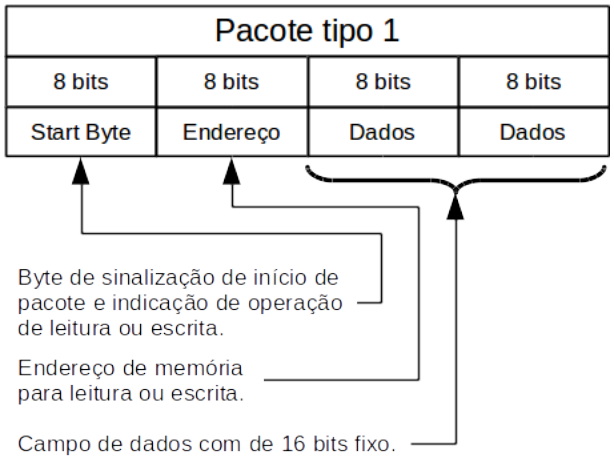


**Figure A.4:** Pacote tipo 1

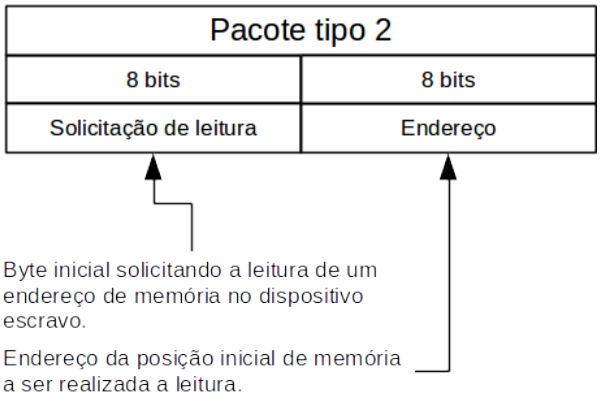


**Figure A.5:** Pacote tipo 2

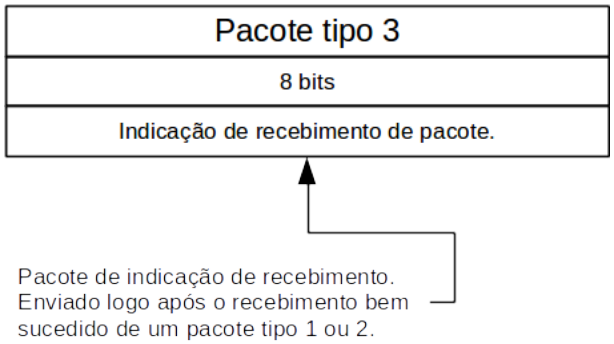


**Figure A.6:** Pacote tipo 3

## A.3 Vistas da Plataforma Robótica

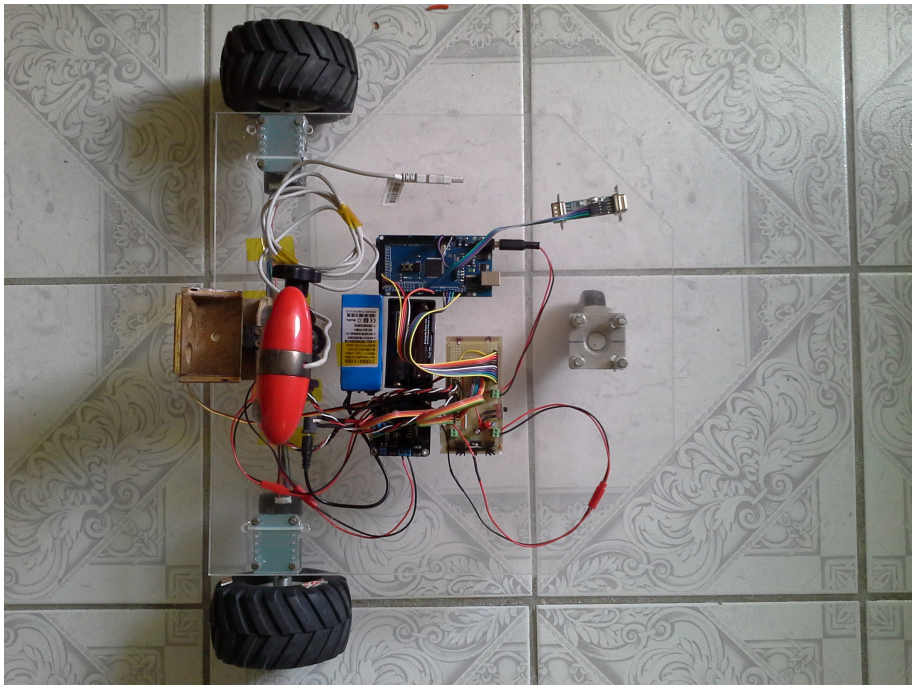

**Figure A.7:** Vista superior da plataforma robótica desenvolvida para os testes.

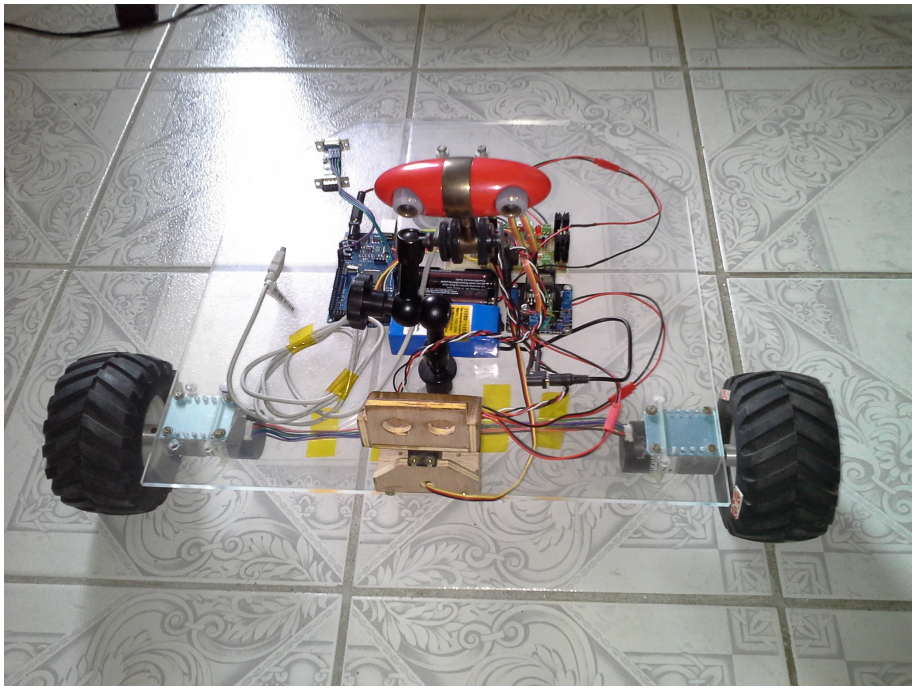

**Figure A.8:** Vista frontal da plataforma robótica desenvolvida para os teste do sistema.

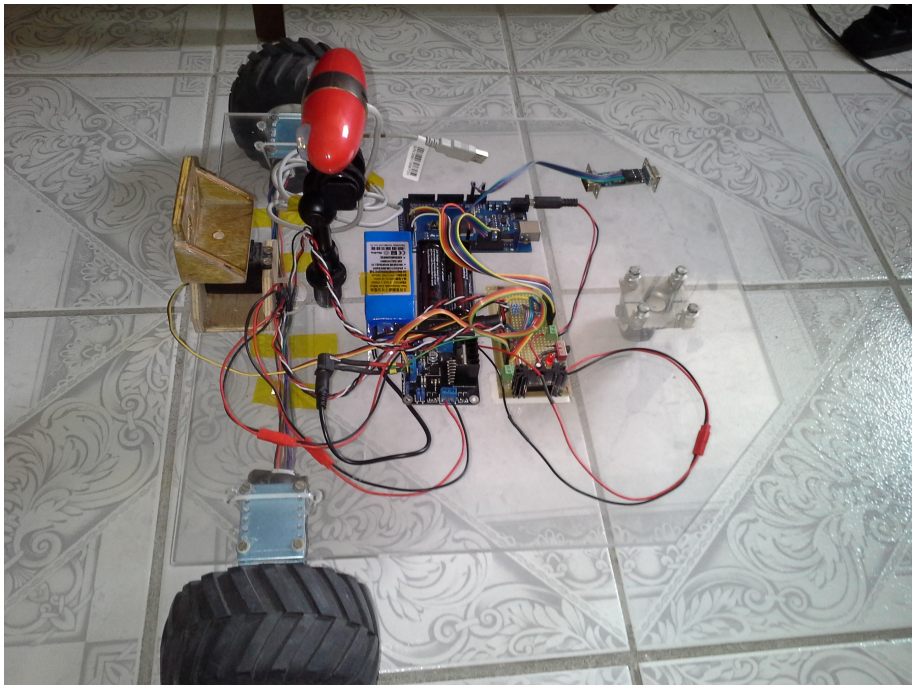

**Figure A.9:** Vista lateral da plataforma robótica desenvolvida para os teste do sistema.