


Luan Silveira

Sistema Bioinspirado para Mapeamento e Localização de Robôs Móveis em Ambientes Subaquáticos

Rio Grande, RS, Brasil

mar - 2015

Luan Silveira

Sistema Bioinspirado para Mapeamento e Localização de Robôs Móveis em Ambientes Subaquáticos

Dissertação de Mestrado apresentada ao Programa de Pós Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtensão do grau de Mestre em Engenharia de Computação

Universidade Federal do Rio Grande - FURG

Centro de Ciências Computacionais - C3

Programa de Pós Graduação em Computação

Curso de Mestrado em Engenharia de Computação

Orientadora: Prof. Dra. Silvia Silva da Costa Botelho

Rio Grande, RS, Brasil

mar - 2015

Luan Silveira
Sistema Bioinspirado para Mapeamento e Localização de Robôs Móveis em Ambientes Subaquáticos/ Luan Silveira. – Rio Grande, RS, Brasil, mar - 2015-
97 p. : il. (algumas color.) ; 30 cm.

Orientadora: Prof. Dra. Silvia Silva da Costa Botelho

Dissertação (Mestrado) – Universidade Federal do Rio Grande - FURG
Centro de Ciências Computacionais - C3
Programa de Pós Graduação em Computação
Curso de Mestrado em Engenharia de Computação, mar - 2015.

1. Robótica Subaquática. 2. SLAM. I. Prof. Dra. Silvia Silva da Costa Botelho. II. Universidade Federal do Rio Grande. III. Centro de Ciências Computacionais. IV. Sistema Bioinspirado para Mapeamento e Localização de Robôs Móveis em Ambientes Subaquáticos

CDU 02:141:005.7

Luan Silveira

Sistema Bioinspirado para Mapeamento e Localização de Robôs Móveis em Ambientes Subaquáticos

Dissertação de Mestrado apresentada ao Programa de Pós Graduação em Computação da Universidade Federal do Rio Grande, como requisito parcial para a obtensão do grau de Mestre em Engenharia de Computação

Trabalho aprovada. Rio Grande, RS, Brasil, 05 de março de 2015

Prof. Dra. Silvia Silva da Costa Botelho
Orientadora - FURG

Prof. Dr. Edson Prestes
Convidado 1 - UFRGS

Dr. Ney Robinson Salvi dos Reis
Convidado 2 - Cenpes - Petrobrás

Rio Grande, RS, Brasil
mar - 2015

Dedico este trabalho às pessoas que estão sempre ao meu lado,
apoiando-me em minhas decisões:

Minha mãe Elaine;
Minha namorada Iana;
Minha irmã Letícia;
Meu sobrinho César;
E meu cachorro Bob Balotelli.

Agradecimentos

Há exatos sete anos atrás eu entrava na faculdade. Queria me formar um engenheiro. Entre as opções, estava o curso de Engenharia de computação, uma área bastante desafiadora, onde aparecia no currículo a Robótica. Naquela época, não tinha muita ideia do que se tratava, mas me imaginava construindo um robô daqueles que aparecem em filmes.

Aos poucos fui aprendendo o que era a grande área da computação, e ao mesmo tempo entristecendo-me por utilizar cada vez menos o que aprendia nas disciplinas de engenharia. Em meados do segundo ano de faculdade, entrei para o NAUTEC, um laboratório incrível, onde os alunos realmente trabalhavam com robótica. Não o que eu imaginava por robótica lá no começo da faculdade, mas algo mais próximo do mundo real.

Foi uma longa caminhada até o final na graduação, vários projetos de iniciação científica, muitas noites sem dormir. Nessas horas, nos damos conta da importância de ter amigos, pessoas com quem podemos contar nos momentos difíceis. Agradeço a minha família, meus eternos amigos, pelo constante apoio. E aos amigos da faculdade, os irmãos que escolhemos, que levarei para toda a vida.

Acabando a faculdade, decido continuar no universo acadêmico, fazer um mestrado, permanecendo na robótica. Mais dois anos de"sofrimento", pelas inúmeras frustrações que a pesquisa nos proporciona. A cada dez passos para trás, conseguimos avançar um passo. Como pessoas, o crescimento é enorme.

Agradeço o grande apoio recebido de minha namorada Iana durante grande parte de minha caminhada, desde o final da graduação até o final do mestrado. Sem ele, tudo seria mais difícil.

Agradeço ao apoio de minha família, especialmente a minha mãe, sem ela eu nada seria. À minha irmã e sobrinho pela companhia familiar.

Agradeço aos colegas do mestrado, inicialmente estranhos, que viraram amigos. Cada um com sua formação, um conhecimento diferente para compartilhar. Agradeço ao Felipe Guth, nosso mestre, pela parceria de pesquisa. Agradeço ao Felipe Codevilla, pelos trabalhos e experiências compartilhados. Agradeço ao Steffens, pelas discussões sobre linguagens de programação. Agradeço ao Telmo, por suas discussões sobre os mais diversos assuntos, incluindo política. Nessa o Steffens tava sempre junto. Agradeço ao Machado, pela parceria, juntamente com o "altímetro", nas expedições subaquáticas. Ao Alex, único colega que sabe(tenta) jogar futebol. E ao Ricardo, meu segundo irmão, que apesar de não ser do mestrado, estava sempre no laboratório, "estudando".

Agradeço aos colegas de laboratório. Em especial ao Ballester, pela constante ajuda nos códigos e artigos. Ao Sidnei, amigo de longa data, sempre disponível para discussões acerca dos problemas encontrados. E a toda a galera do NAUTEC.

Não menos importante, agradeço a minha orientadora Silvia Botelho, essencial durante toda a minha caminhada acadêmica. Sempre com pouco tempo, mais muita vontade de revolucionar a pesquisa. É uma pessoa fora do padrão.

Por fim, agradeço o apoio financeiro da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis – ANP – , da Financiadora de Estudos e Projetos – FINEP – e do Ministério da Ciência e Tecnologia – MCT por meio do Programa de Recursos Humanos da ANP para o Setor Petróleo e Gás – PRH-ANP/MCT. Um agradecimento especial aos professores responsáveis pelo PRH - 27, Maria Isabel e Gilberto Griep. Ainda, agradeço ao apoio financeiro da Fapergs por meio do Programa de Internacionalização da Pós Graduação do Rio Grande do Sul.

A todos, Meu Muito Obrigado!

"When the sea was calm,
all ships alike showed mastership in floating..."
(William Shakespeare)

Resumo

A exploração de ambientes subaquáticos é um cenário comum nos dias atuais, principalmente em aplicações comerciais, como a extração de petróleo e a mineração. Esses ambientes apresentam novos desafios tecnológicos, buscando na robótica móvel soluções para os problemas encontrados, criando uma demanda por algoritmos que possibilitem a atuação dos robôs nesses ambientes. Aqui será apresentado um novo método para localização e mapeamento simultâneos de robôs em ambientes subaquáticos, inspirado em neurônios existentes no cérebro dos mamíferos. O método proposto baseia-se nos algoritmos RatSLAM e FAB-MAP, aplicando essas técnicas ao ambiente subaquático. O trabalho apresentará os principais sensores utilizados por robôs subaquáticos e os métodos desenvolvidos nos últimos anos relacionados ao problema de mapeamento e localização. Ao final, serão apresentados três estudos de caso para validação da proposta, demonstrando suas vantagens e limitações. **Palavras-chaves**: Robótica Subaquática. SLAM.

Abstract

The exploitation of underwater environments is a common scenario these days, especially in commercial applications, such as oil drilling and mining. These environments present new technological challenges, seeking mobile robotics solutions to encountered problems. With this scenario, new algorithms have to be developed for mobile robots. Here, we propose a new method for simultaneous localization and mapping in underwater environments, inspired by neurons found in mammalian brain. The proposed method extends the RatSLAM and FAB-MAP algorithms, seeking to apply these techniques to the underwater environment. We also present the sensors used by underwater robots and the main methods developed to solve the SLAM problem. In the end, we validate our algorithm in three different scenarios and demonstrate its advantages and limitations.

Key-words: Underwater Robotics. SLAM.

Lista de ilustrações

Lista de tabelas

Lista de Abreviaturas e Siglas

AUV Veículo Subaquático Autônomo

CANN Rede Neural de Atração Contínua

DGPS Sistema de Posicionamento Global Diferencial

DVL Doppler Velocity Log

EIF Filtro de Informação Estendido

EKFSLAM SLAM baseado em Filtro de Kalman Extendido

EKF Filtro de Kalman Estendido

GPS Sistema de Posicionamento Global

IMU Unidade de Medição Inercial

LASER Light Amplification by Stimulated Emission of Radiation

LBL Long Baseline Positioning System

MIT Massachussets Institute of Technology

ROV Veículo Operado Remotamente

SBL Short Baseline Positioning System

SLAM Localização e Mapeamento Simultâneos

SURF Speed-Up Robust Features

USBL Ultra Short Baseline Positioning System

UUV Veículo Subaquático Não-Tripulado

VAN Visual Augmented Navigation

Sumário

1 Introdução

O avanço tecnológico permitiu a aplicação da robótica nos mais diversos ambientes, desde o fundo dos oceanos até a superfície de outros planetas. Seu constante desenvolvimento torna possível a utilização dos robôs em tarefas autônomas, como fabricação de produtos industriais, limpeza de residências, inspeção de ambientes inóspitos, análise e desmonte de explosivos, entre outros.

No Brasil, a indústria petrolífera recebeu vultosos investimentos nos últimos anos, aumentando a demanda por tecnologias específicas aplicadas aos ambientes de extração e distribuição de óleo e gás. Dentre os cenários mais desafiadores para a indústria estão a instalação e inspeção de equipamentos em águas ultraprofundas, onde não é possível a presença humana. As operações de controle e supervisão desses equipamentos é realizada com a utilização de computadores, sensores e robôs (BRASILIS, 2012). Em 2011, o custo do aluguel de um robô subaquático, juntamente com seu navio de controle, era de aproximadamente 500 mil reais por dia (STAUFFER, 2011a).

Atualmente, os principais robôs aplicados em missões subaquáticas são os Veículos Operados Remotamente (ROVs), os quais são controlados remotamente de uma estação base instalada em um navio de operação. O navio deve acompanhar o robô durante toda a operação. Os ROVs são conectados à estação base por meio de um cabo umbilical, responsável pela transmissão de dados e energia. Esse cabo pode conter alguns quilômetros de comprimento (STAUFFER, 2011b), dificultando a navegação do robô pelo aumento do arraste sofrido na água. Além disso, podem ocorrer entrelaçamentos no cabo umbilical, acarretando a ruptura da fibra óptica e consequente interrupção no controle do robô. Ainda, em casos extremos, o cabo pode romper-se, fazendo com que o robô perca-se no ambiente.

Outra classe de robôs, ainda pouco aplicada à indústria, é composta pelos Veículos Subaquáticos Autônomos (AUVs). Esses robôs realizam tarefas de forma autônoma, não sendo necessário o acompanhamento humano durante a operação, reduzindo os custos associados ao aluguel de um navio de apoio. No entanto, sua aplicação ainda é limitada pela inexistência de inteligência artificial confiável para a realização de atividades complexas. Ainda, a baixa autonomia de suas baterias reduz o campo de atuação desses robôs. Atualmente, as principais aplicações dos AUVs estão relacionadas a missões de monitoramento ambiental.

A expansão do uso de AUVs em missões subaquáticas mais complexas e, com isso, a presença de inteligência artificial a bordo desses veículos apresenta para a robótica uma série de novos desafios, relacionados à localização, percepção, comunicação, mapeamento,

controle e navegação. O meio físico apresenta-se como um meio participativo, apresentando restrições na utilização de sensores e limitações à navegabilidade do sistema robótico. Primeiramente, não existe um sistema de posicionamento global abaixo da superfície da água, devido à atenuação das ondas de rádio no meio líquido. As alternativas existentes para a localização de um sistema robótico limitam o ambiente de atuação do veículo e necessitam, para uma maior precisão, da instalação de equipamentos georreferenciados no fundo dos oceanos.

Em relação à comunicação sem fio, faz-se necessário o envio de dados por ondas sonoras. Porém, os sistemas existentes atualmente não são capazes de enviar dados a uma alta frequência, o que limita a velocidade de envio. Devido à perda de sinal e atrasos na transmissão, não é viável sua utilização na teleoperação de sistemas robóticos.

A navegação de uma plataforma robótica na água sofre o efeito das correntes e do arraste do meio, pois é necessário o deslocamento de fluído para a movimentação do veículo. Dessa forma, os veículos devem ter propulsão suficiente para superar o arraste e, ao mesmo tempo, precisão na modelagem de tarefas complexas, como por exemplo, o fechamento de uma válvula de controle em um poço de extração de petróleo.

A percepção do ambiente, de forma similar, apresenta algumas peculiaridades referentes ao meio. Sensores baseados em luz, como câmeras e lasers, amplamente utilizados na robótica terrestre, estão sujeitos aos efeitos da turbidez da água, causada pelas partículas em suspensão no fluído. Essa turbidez é responsável por uma alta atenuação das ondas luminosas. Com isso, esses sensores tem sua aplicabilidade limitada a operações de curta distância, isto é, onde o robô navega próximo aos objetos a serem monitorados.

Além disso, o ambiente subaquático é inerentemente não estruturado, composto por elementos naturais ambíguos, difíceis de serem percebidos e rastreados, havendo demanda por algoritmos robustos para a detecção de pontos discriminantes na cena. Em virtude da limitação dos sensores baseados em luz, aplicações de longa distância requerem a utilização de sonares. Esses sensores são caracterizados pela sensibilidade a ruídos e formação de ecos acústicos, trazendo novos desafios relacionados ao tratamento do sinal recebido.

Por todo o exposto, percebe-se a existência de um novo cenário para a robótica móvel, o qual requer o desenvolvimento de algoritmos para a solução dos problemas básicos, como a localização no ambiente e a criação de uma representação computacional para os objetos encontrados pelo robô. Esses dois problemas, tratados de forma concorrente, são conhecidos na robótica como Localização e Mapeamento Simultâneos (SLAM).

Considerado um problema resolvido em ambientes terrestres, o SLAM ainda está em aberto no ambiente subaquático, principalmente pela mudança sensorial existente e pela característica não estruturada do ambiente. Os métodos clássicos são sensíveis à detecção de pontos discriminativos robustos e possuem alto custo computacional no

mapeamento de ambientes vastos.

Dessa forma, novas abordagens para a resolução do problema são necessárias. Como opção, busca-se inspiração na natureza. Os animais, muitas vezes com sensores precários, resolvem o problema de localização e mapeamento de forma eficaz. Os mamíferos, por exemplo, utilizam mecanismos cerebrais especiais, criando um sistema de representação do ambiente capaz de estimar o melhor caminho entre dois pontos no espaço, mesmo na ausência de informações perceptivas do ambiente (MCNAUGHTON et al., 2006).

As pesquisas em neurociência animal culminaram na descoberta de classes de neurônios relacionados a tarefas de navegação no cérebro dos mamíferos. As principais, relacionadas à localização do mamífero, são as *Place Cells* (O'KEEFE; DOSTROVSKY, 1971) e as *Grid Cells* (HAFTING et al., 2005). Para orientação, esses animais utilizam principalmente as *Head Direction Cells* (TAUBE; MULLER; JR, 1990). Esse conjunto de células é utilizado em um sistema de navegação nos mamíferos e suas interconexões são objeto de estudo da neurociência moderna.

1.1 Objetivos

O presente trabalho tem com objetivo geral propor uma solução para o problema de mapeamento e localização de robôs móveis em ambientes subaquáticos. Em virtude da diversidade encontrada nos ecossistemas marinhos, a presente proposta será focada em ambientes de extração e distribuição de óleo e gás *offshore*.

Para definição do escopo da proposta, a seguir serão elencados os objetivos específicos:

- Realizar um estudo dos principais sensores utilizados na robótica subaquática e o que os diferenciam da robótica terrestre. Elencar os desafios encontrados e a necessidade algorítmica para o tratamento dos dados sensoriais.
- Desenvolver uma metodologia inspirada no cérebro dos mamíferos, realizando um estudo dos modelos computacionais dos neurônios espaciais desses animais e sua viabilidade para a robótica.
- Estender o algoritmo RatSLAM, inicialmente desenvolvido para a resolução do problema de SLAM visual em ambientes terrestres, para aplicação em ambientes subaquáticos. Buscar-se-á extrair as principais vantagens do algoritmo RatSLAM ligadas ao mapeamento de ambientes amplos e monótonos.
- Desenvolver um método capaz de mapear ambientes com a utilização de multiplos sensores, baseados em imagens ópticas e acústicas, tornando sua utilização possível em tarefas de curta e longa distância entre o robô e a cena.

- Utilizar o algoritmo FAB-MAP, estado da arte na criação de mapas no espaço de aparências, com o objetivo de garantir maior robustez no reconhecimento de locais.

1.2 Estrutura da Dissertação

A presente dissertação está dividida em sete capítulos, organizados da seguinte maneira: O Capítulo 2 apresentará uma definição formal para o problema de Localização e Mapeamento Simultâneos (SLAM) clássico da robótica, apresentando as principais classificações de acordo com o tipo de mapa criado e método de resolução. Atualmente, esse problema é considerado resolvido em ambientes terrestres estruturados. Ainda, serão apresentadas as diferentes abordagens utilizadas na resolução do problema: probabilística e bioinspirada. Os principais métodos desenvolvidos em cada abordagem serão detalhados ao final do capítulo.

Após a definição do problema, o Capítulo 3 focará nos principais aspectos dos ambientes subaquáticos importantes para a resolução do problema. Serão explicados os principais sensores utilizados na aquisição de informações do ambiente e do estado do próprio robô. Ao final, serão apresentados os algoritmos que compõem o estado da arte em SLAM subaquático e uma análise das vantagens e desvantagens de cada método.

O Capítulo 4 apresentará os algoritmos utilizados como base para a presente proposta. Inicialmente, será apresentado o RatSLAM, desenvolvido por Milford e Wyeth (2008). Sua inspiração é detalhada na subseção 4.1.1 com a explicação dos principais mecanismos cerebrais encontrados nos mamíferos. Na sequência, os detalhes do método RatSLAM são definidos, finalizando com a apresentação dos principais trabalhos de mapeamento com a utilização do sistema.

No mesmo capítulo, será apresentado o método FAB-MAP (CUMMINS; NEWMAN, 2008), estado da arte em SLAM baseado em aparências. Esse método utiliza uma representação das imagens como um conjunto de palavras, de acordo com o algoritmo Bag of Words (SIVIC; ZISSERMAN, 2003), o qual também será explicado na subseção 4.2.1. Na sequência serão apresentados os principais trabalhos que empregam esse método, além de trabalhos que integram o RatSLAM e o FAB-MAP em ambientes terrestres.

Por sua vez, no Capítulo 5 será apresentado o método proposto no presente trabalho, o qual fará integração do RatSLAM com o FAB-MAP, modificando os algoritmos para viabilizar sua aplicação em ambientes subaquáticos. Serão detalhadas a integração dos métodos e as modificações realizadas nos algoritmos.

O capítulo Capítulo 6 apresentará três estudos de caso realizados com o método proposto. Primeiramente o método foi testado em um ambiente simulado utilizando câmera óptica. No segundo, foi utilizado um dataset real de uma piscina, onde o robô

estava equipado com uma câmera óptica e sensores de movimentação. Por fim, utilizou-se um cenário de uma marina, onde a captura de dados foi realizada com um sonar de imageamento.

O capítulo termina com uma discussão sobre o funcionamento do método nos três diferentes cenários, apresentando as vantagens e limitações da abordagem. Ao final, o 7 apresenta as conclusões obtidas com o desenvolvimento do método.

2 Definição do Problema

A localização, definida como a capacidade de inferência de posição no ambiente com base em um referencial fixo, é uma das principais aptidões necessárias a um sistema robótico móvel autônomo. Em um primeiro cenário, essa habilidade pode ser alcançada pela utilização de sensores de posicionamento, como o Sistema de Posicionamento Global (GPS). O GPS é capaz de retornar a posição do robô em um sistema de coordenadas georreferenciado, juntamente com uma incerteza de medição. No entanto, esses sistemas nem sempre estão disponíveis ou sua precisão é insuficiente para a aplicação.

Em um segundo cenário, pode-se utilizar um mapa representativo do ambiente como base para um algoritmo de localização. Neste caso, o mapa é conhecido *a priori* e o objetivo passa a ser a localização do robô no mapa, utilizando dados sensoriais para detecção de *landmarks* no ambiente.

Como terceiro cenário, o robô não possui conhecimento sobre o ambiente e tampouco possui acesso a sistemas de medição de sua posição absoluta no ambiente. Em um mundo hipotético, pode-se pensar na utilização de sensores de movimentação proprioceptivos, como os *encoders*, e estimar a posição do robô desde o início de seu deslocamento. O problema, nesse caso, está na inexistência de sensores 100% precisos. Os sistemas baseados apenas na leitura desses dispositivos, também chamados de sensores de odometria, possuem como característica um erro crescente na localização do agente, tornando as posições inferidas dessa forma cada vez mais longe da posição real. Isto deve-se ao acúmulo de erro a cada leitura sensorial. (THRUN, 2008).

Dessa forma, faz-se necessário o desenvolvimento de algoritmos que utilizem os sensores existentes e, ao mesmo tempo, eliminem o crescimento ilimitado do erro de localização. Passamos, dessa forma, à definição do problema tratado no presente trabalho, conhecido como mapeamento e localização simultâneos, um problema clássico na robótica móvel.

2.1 Mapeamento e Localização Simultâneos

O problema de SLAM é definido como a capacidade de um robô navegar em um ambiente desconhecido e, por meio de leituras sensoriais, construir um mapa representativo do ambiente navegado. De forma simultânea, o algoritmo localiza o robô no mapa criado (DURRANT-WHYTE; BAILEY, 2006a). Esse processo é feito de forma incremental, aumentando a certeza de posição do robô ao mesmo tempo em que ocorre um refinamento no mapa representativo do ambiente.

Com posse de um algoritmo de SLAM, o robô torna-se capaz de realizar tarefas em um ambiente desconhecido. Por exemplo, ele pode realizar expedições de inspeção em ambientes de difícil acesso humano, como o ambiente subaquático. Ainda, uma vez em posse do mapa do ambiente, este pode ser utilizado para análise posterior, realizada por operadores humanos sobre os dados sensoriais adquiridos de forma autônoma.

Normalmente, um algoritmo de SLAM utiliza dois tipos de entradas: uma entrada motora própria (proprioceptiva) representando o deslocamento relativo do robô e uma entrada sensorial externa (exteroceptiva) , esta última responsável pela detecção de pontos característicos do ambiente. Entre os principais sensores aplicados em SLAM estão as câmeras ópticas, lasers 2D, sonares, kinect, entre outros. Com isso, um algoritmo para SLAM é um processo de estimativa da localização do robô baseado na observação de características do ambiente. À medida que o agente navega, essas características são reobservadas e a posição do robô é atualizada. O erro de medição dos *landmarks* do ambiente também deve ser levado em consideração e minimizado sempre que possível.

2.1.1 Formulação clássica

Em sua formulação clássica, o problema de SLAM é definido como a criação de um mapa métrico, em que cada localização do robô está associada à localização anterior e, por sua vez, a posição das pistas sensoriais está conectada com a posição do robô onde essas informações foram adquiridas, como mostrado na Figura 1.

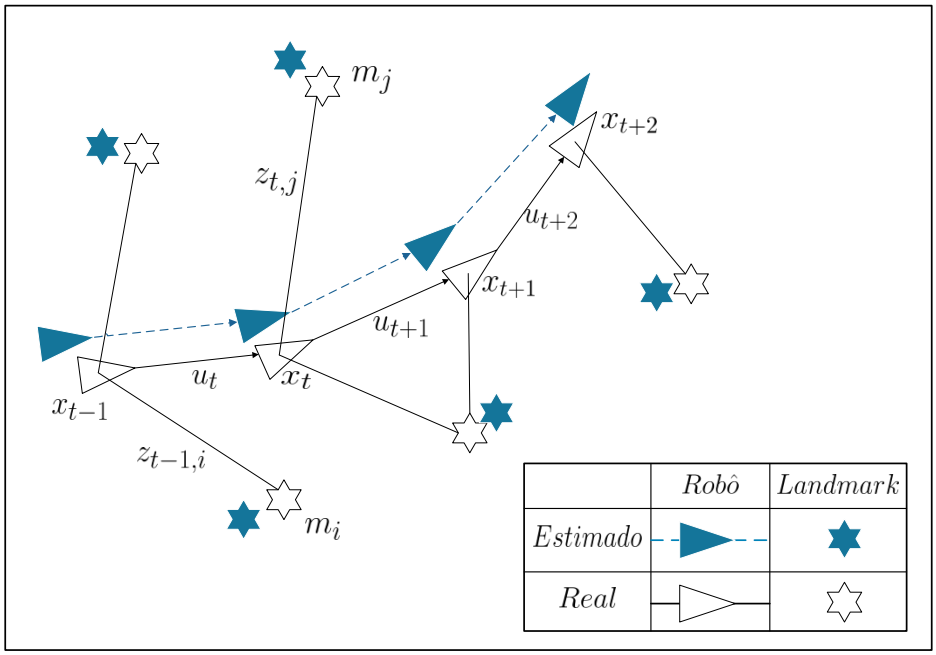

Figura 1: Formulação clássica para a resolução do problema de SLAM

Nessa figura, cada posição colocada no mapa está relacionada com um instante de

tempo t. Dessa forma, podemos definir as seguintes variáveis envolvidas: (DURRANT-WHYTE; BAILEY, 2006a)

- x_t: Vetor de estados descrevendo a posição e orientação do veículo
- u_t: Vetor de controles, aplicado no tempo $t-1$ para levar o veículo até o estado x_t no instante t
- m^i: Um vetor descrevendo a localização do i-ésimo ponto de referência (*landmark*, contendo a posição deste, sendo invariante no tempo(ambiente estático).
- $z_{t,i}$: Uma observação tomada pelo veículo sobre a localização do i-ésimo ponto de referência no tempo t

Para robôs que se movimentam em ambientes planos, x_t é normalmente um vetor tridimensional, compreendendo as coordenadas 2D do plano acrescido do valor de orientação do robô. No caso de robôs que se movimentam no espaço tridimensional, x_t compreende as três variáveis do espaço, mais a orientação do robô, a qual pode ser apenas sobre o eixo z ou as três orientações. Neste último caso, x_t possui seis graus de liberdade.

A sequência de posições do robô, ou seja, o caminho percorrido por este, é definida como:

$$X_T = \{x_0, x_1, x_2, ..., x_T\}, \tag{2.1}$$

onde T é o tempo final, podendo ser infinito dependendo da aplicação. A posição inicial x_0 é conhecida e as posições subsequentes sequer podem ser observadas, devendo ser estimadas.

As informações de controle são oriundas dos sensores de movimentação própria, como *encoders* [1] nos motores do robô ou sensores de medição inercial [2] Esses sensores são conhecidos por sensores de odometria, ou proprioceptivos, pois proveem informações relativas entre duas posições consecutivas. O processo de integrar as informações oriundas dos sensores motores para criação do mapa é conhecido no contexto de SLAM como *Dead Reckoning*.

A sequência de movimentação realizada pelo robô é definida por:

$$U_T = \{u_1, u_2, u_3, ..., u_T\} \tag{2.2}$$

[1] Encoder é um dispositivo óptico que conta ou reproduz pulsos elétricos a partir do movimento rotacional de seu eixo.

[2] Unidade de Medição Inercial é um dispositivo eletrônico de medição de aceleração, orientação e força gravitacional usando uma combinação de acelerômetros, giroscópios e magnetômetros.

Com o objetivo de corrigir os erros inerentes a odometria, o robô observa o ambiente por onde navega. Considerando m o mapa real do ambiente, tipicamente estático, composto por objetos, superfícies e pontos de referência, cada pista encontrada nesse ambiente possui uma posição real, representada por m_i, onde i representa o i-ésimo landmark nesse ambiente.

Considerando que o robô observa o ambiente em cada instante de tempo e, sem perda de generalização, adquire apenas uma observação a cada instante, a sequência de todas as medições tomadas pelo robô dos pontos de referência encontrados no ambiente é dada por:

$$Z_T = \{z_0, z_1, z_2, ..., z_T\} \tag{2.3}$$

A partir dessa notação, a literatura define duas formas principais para a resolução do problema: SLAM Offline e SLAM Online. (THRUN, 2008)

SLAM Offline

Na resolução do problema offline, também conhecida por SLAM completo, calcula-se a probabilidade da estimativa de posição de todos os lugares percorridos pelo robô, juntamente com o mapa, representada pela seguinte equação:

$$P(X_T, m|Z_T, U_T) \tag{2.4}$$

Escrita dessa forma, essa abordagem baseia-se em calcular a probabilidade posterior conjunta sobre X_T e m com base em todos os dados adquiridos até o momento atual. Nota-se que as variáveis da direita da barra condicional são diretamente observáveis pelo robô.

Os algoritmos para SLAM Completo são executados normalmente em lote, processando todos os dados ao mesmo tempo. Em vista disso, possuem um custo computacional maior comparado com a versão online.

SLAM Online

Igualmente importante ao anterior, procura descobrir apenas a posição atual do robô, ao invés de todo o caminho percorrido por ele, sendo definido pela seguinte equação:

$$P(x_T, m|Z_T, U_T) \tag{2.5}$$

Essa definição online baseia-se na concepção do problema de SLAM como uma Cadeia de Markov, como representado na Figura 2, onde o estado atual representa todas as posições anteriores percorridas pelo robô bem como as ações por ele executadas.

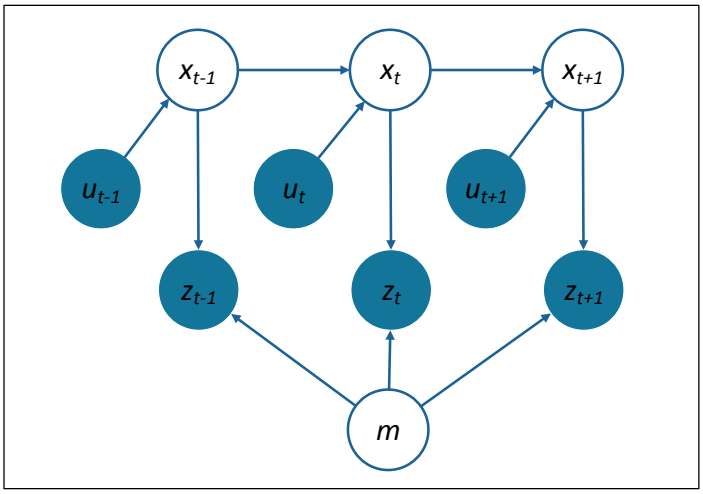

Figura 2: Formulação Gráfica para o problema de SLAM *online*

Independente da forma escolhida para o tratamento do problema, *online* ou *offline*, partindo dessa formulação clássica é necessária a definição de um modelo de observação do mapa e de um modelo de movimentação do robô.

O modelo de observação relaciona a leitura sensorial obtida com a posição atual do robô. Seguindo a mesma notação, podemos definí-lo como:

$$P(z_t|x_t, m), \tag{2.6}$$

onde busca-se calcular a probabilidade da medição z_t dado a posição atual e o mapa.

De forma semelhante, define-se um modelo de movimentação que relaciona a posição atual do robô com a posição anterior, dado que o robô executou uma determinada ação:

$$P(x_t|x_{t-1}, u_t), \tag{2.7}$$

onde busca-se estimar a posição atual dada a posição anterior e a último controle executado.

Tipos de mapa

Existem dois tipos de mapas gerados por um sistema de SLAM (THRUN, 2008): mapa topológico e mapa métrico.

Mapas topológicos armazenam apenas uma descrição qualitativa do ambiente, caracterizando as relações de vizinhança entre os locais do mapa (O local A é adjacente ao local B).

Mapas métricos proveem uma informação métrica de ligação entre os locais (O local A está 10m à frente do local B).

Abordagens probabilísticas para a resolução do problema de SLAM normalmente descrevem o ambiente como um mapa métrico. Por outro lado, há evidências que nós humanos utilizamos informações topológicas para navegação.

2.1.2 Formulação baseada na trajetória do robô

Diferentemente da formulação clássica descrita anteriormente, é possível uma formulação alternativa para o problema, onde sua resolução baseia-se na construção de um mapa da trajetória percorrida pelo robô. A cada instante de tempo t, armazena-se a leitura sensorial capturada no ambiente, juntamente com a posição onde esta foi adquirida (DURRANT-WHYTE; BAILEY, 2006b).

Formulado dessa forma, o sistema não fornece o mapa do ambiente diretamente, as sim estima todo o caminho percorrido pelo robô, onde cada posição registra uma representação qualitativa do ambiente, formando um mapa topológico. Por exemplo, cada posição do mapa está relacionada com uma imagem capturada no tempo t e ligada a posição percorrida no tempo $t-1$, formando ligações topológicas no mapa.

Essa formulação é particularmente adequada para ambientes onde a identificação de pistas discretas é uma tarefa complexa, tornando o alinhamento direto dos dados capturados mais simples e robusto. O ambiente subaquático é um bom exemplo onde a detecção de *landmarks* robustos nem sempre é possível, devido a alta ambiguidade dos dados capturados no ambiente. Além disso, essa formulação torna viável o mapeamento de ambientes extensos, com quilômetros de distância percorrida pelo robô.

2.2 Abordagens para resolução do problema

Historicamente, o problema de SLAM foi tratado por abordagens probabilísticas, devido às incertezas relacionadas à movimentação do robô e aos sensores exteroceptivos, além da dificuldade de associação dos landmarks do ambiente. Dentre as abordagens probabilísticas, pode-se citar o SLAM baseado em Filtro de Kalman Extendido (EKFSLAM), o FastSLAM, entre outros. Essa classe de algoritmos probabilísticos parte das seguintes premissas:

- A próxima posição do robô só depende da posição atual e do controle aplicado ao

robô (THRUN, 2008).:

$$P(x_t|x_{t-1}, u_t), \tag{2.8}$$

- A leitura sensorial atual só depende da posição do robô e do mapa:

$$P(z_t|x_t, m) \tag{2.9}$$

A solução do problema transforma-se na resolução iterativa dessas equações, motivo pelo qual os algoritmos dessa classe são chamados de Filtros (THRUN, 2008)

Como alternativa à abordagem clássica surgiu a abordagem bioinspirada, foco do presente trabalho. Nela, busca-se modelar a capacidade de localização e mapeamento dos animais com o objetivo de desenvolver algoritmos robustos para a resolução do problema de SLAM.

As próximas subseções apresentarão essas duas abordagens para resolução do problema, probabilística e bioinspirada, juntamente com os principais algoritmos desenvolvidos em cada área.

2.2.1 Abordagens probabilísticas

Partindo da definição clássica para o problema de SLAM, desenvolveu-se várias classes de algoritmos probabilísticos. Aqui serão detalhados dois algoritmos que servem como base para grande parte das variações que surgiram com o desenvolvimento do problema. São eles: O Filtro de Kalman Estendido e o Filtro de Partículas.

2.2.1.1 Filtro de Kalman Extendido

O Filtro de Kalman Estendido (EKF) é uma extensão do Filtro de Kalman desenvolvido por Kalman (1960). Historicamente, foi o algoritmo pioneiro aplicado à resolução do problema de SLAM.

Introduzido por Smith e Cheeseman (1986), o EKFSLAM foi o primeiro algoritmo a propor o uso de um único vetor de estados para estimar a posição do robô e de um conjunto de *features* no ambiente, associado a uma matriz de covariância relacionada à incerteza na estimativa, incluindo a correlação entre os dados do vetor de estados.

O EKFSLAM representa o vetor de estados por uma distribuição gaussiana multivariada:

$$p(x_t, m|Z_T, U_T) \sim \mathcal{N}(\mu_t, \Sigma_t), \tag{2.10}$$

onde o vetor μ_t contém a melhor estimativa da posição do robô e das features do ambiente. A matriz Σ_t é a matriz de covariâncias relacionada a estimativa de μ_t. (THRUN, 2008)

O modelo de movimentação é, de modo similar, aproximado por uma distribuição gaussiana multivariada. A posição do robô no tempo t, após a aplicação da ação de controle u_t sobre o estado x_{t-1} é calculada pela função $g(x_{t-1}, u_t)$. A nova estimativa de posição é então aproximada por:

$$p(x_t|x_{t-1}, u_t) \sim \mathcal{N}(g(x_{t-1}, u_t), R_t). \tag{2.11}$$

A incerteza na movimentação é representada pela matriz de covariância R_t.

Por sua vez, o modelo de observação, o qual estima a posição das features no mapa de acordo com a posição do robô, é estimado por:

$$p(z_t|x_t, m) \sim \mathcal{N}(h(x_t, m), Q_t), \tag{2.12}$$

onde a função $h(x_t, m)$ retorna a posição do landmark no ambiente e a matriz de covariância Q_t representa o erro associado à medição.

À medida que novos landmarks são observados, novos estados são adicionados ao vetor de estados μ_t, fazendo com que a matriz de covariância cresça quadraticamente ao número de features existentes no mapa. Em geral, a abordagem probabilística assume uma representação métrica do ambiente, baseada em features pontuais. (THRUN, 2008).

A Figura 3 ilustra o funcionamento do algoritmo EKFSLAM em um ambiente composto por oito landmarks pontuais (representados por estrelas). O robô inicia seu deslocamento pelo ambiente (linha tracejada na figura). Simultaneamente, o mesmo observa as features dispostas no ambiente, alocando posições para elas no vetor de estados. A incerteza na movimentação do robô é responsável por uma estimativa cada vez mais longe da realidade(elipses cinzas na Figura 3.b). Da mesma forma, as posições dos landmarks são cada vez mais imprecisas (elipses brancas na Figura 3.c). Ao final de seu deslocamento (Figura 3.d), o robô reobserva landmarks conhecidos, ocorrendo o processo conhecido por fechamento de loop. Nesse instante, o algoritmo corrige o erro na estimativa de posição do robô, propagando a correção para todas as features previamente observadas pelo robô. Com isso, a confiabilidade do mapa é aumentada.

A aplicação do Filtro de Kalman para a resolução do problema de SLAM já é bem definida pela literatura e herda os mesmos benefícios e problemas das soluções com EKF para problemas de localização. Entre esses problemas estão a convergência, custo computacional, associação de dados e a não linearidade dos sistemas (DURRANT-WHYTE; BAILEY, 2006a).

O Filtro de Kalman Estendido aplica uma linearização nos modelos de movimentação e observação, os quais são inerentemente não lineares. Por isso, a quantidade de incerteza deve ser consideravelmente pequena, senão o processo de linearização tenderá a

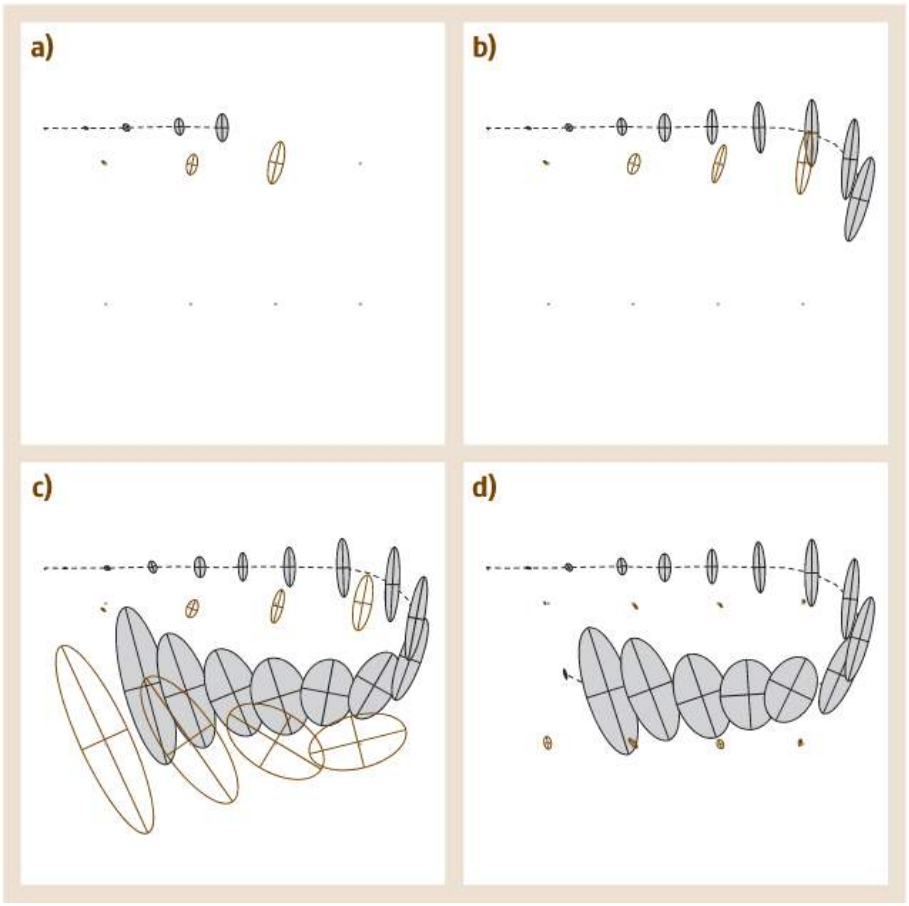

Figura 3: Exemplo de operação do EKF SLAM. As elipses demonstram a incerteza ligada à localização do robô e dos pontos de referência. Quando o robô encontra novamente um marcador conhecido (d), cuja posição ele sabe com um grau aceitável, ele atualiza sua posição atual e a posição dos outros marcadores, levando a uma melhor estimativa de sua posição e da posição de todos os pontos de referência previamente observados no ambiente. (THRUN, 2008)

introduzir erros intoleráveis para o sistema. A convergência e consistência só são garantidas no caso linear.

A escalabilidade do método precisa levar em conta a natureza quadrática da matriz de covariância, limitando normalmente o número de *landmarks* criados. Além disso, o método apresenta melhores resultados em ambientes com pouca ambiguidade entre os *landmarks*. Em caso de uma associação de dados incorreta o mapa pode chegar a um estado inconsistente.

2.2.1.2 Filtro de Partículas

O Filtro de Partículas é uma técnica utilizada para aproximar distribuições de probabilidade arbitrárias por um conjunto de amostras. Reside numa abordagem não paramétrica para implementação do Filtro Bayesiano (METROPOLIS; ULAM, 1949).

A idéia principal do Filtro pode ser vista na Figura 4, onde um conjunto de amostras

é utilizado para a aproximação de uma função gaussiana.

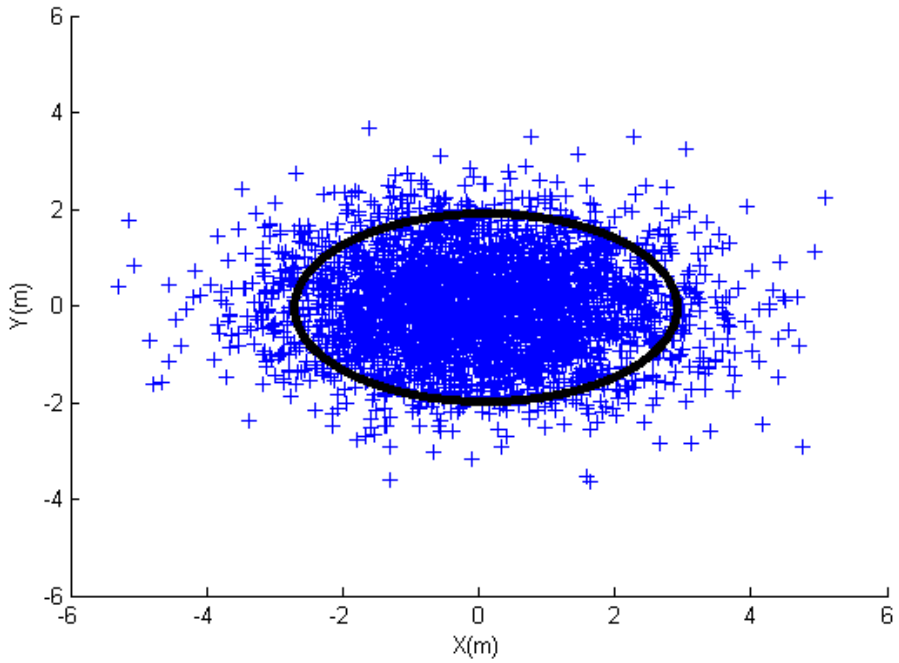

Figura 4: Exemplo de um conjunto de partículas para aproximação de uma Função Gaussiana

A principal vantagem do Filtro de Partículas, frente ao EKF, reside na capacidade de representação de estimativas multimodais. No contexto de SLAM, cada amostra fará a estimativa de todo o caminho percorrido pelo robô e da posição dos *landmarks* do ambiente. Como limitação do Filtro de Partículas padrão, para uma correta estimativa do estado atual do sistema é necessário um conjunto muito grande de partículas. Dessa forma, o crescimento do algoritmo torna-se exponencial ao número de *features* armazenadas no mapa.

A aplicação desse método para a resolução do problema de SLAM só foi possível após os trabalhos de Doucet et al. (2000) e Montemerlo et al. (2002), tornando o algoritmo computacionalmente possível de ser implementado. Essa implementação ficou conhecida como FastSLAM.

A cada instante de tempo t, o algoritmo mantém K amostras. Sendo k o índice da amostra genérica, cada amostra será composta por:

$$X_t^{[k]}, \mu_{t,1}^{[k]}, ..., \mu_{t,N}^{[k]}, \Sigma_{t,1}^{[k]}, ..., \Sigma_{t,N}^{[k]}, \tag{2.13}$$

Dessa forma, cada partícula conterá:

- $X_t^{[k]}$ Uma estimativa de todo o caminho percorrido pelo robô:

- N gaussianas representando a posição dos landmarks no ambiente, representadas pela média $\mu_{t,n}^{[k]}$ e uma variância $\Sigma_{t,n}^{[k]}$, sendo n $(0 \leq n \leq N)$ o índice do landmark.

A partir dessa definição pode-se notar que K partículas irão estimar K possíveis caminhos, além de KN Gaussianas, representando K estimativas diferentes para a posição dos landmarks no ambiente.

Para tornar o filtro computacionalmente viável, o FastSLAM aplica três técnicas: Rao-Blackwellization, independência de todo o caminho para a estimativa de posição do landmark e reamostragem.

O FastSLAM resolve o SLAM completo, onde cada partícula tem uma estimativa de todo o caminho, e o SLAM online, pois cada atualização utiliza apenas a posição mais recente.

A principal vantagem da utilização do FastSLAM está na possibilidade de manter múltiplas hipóteses na associação de dados, obtendo convergência mesmo quando a associação de dados é desconhecida. Além disso, o algoritmo pode ser implementado de forma eficiente, explorando métodos de armazenamento da estimativa do mapa em uma árvore. Com isso, a atualização pode ser executada em tempo logarítmico no tamanho do mapa N e linear ao número de partículas K (THRUN, 2008).

Como se pode notar, tando o EKFSLAM quanto o FastSLAM possuem vantagens e limitações, não havendo uma melhor opção para todo tipo de ambiente a ser mapeado. O Filtro de Kalman possui a limitação de seu custo computacional ser quadrático ao número de *landmarks* alocados no ambiente, dificultando sua escalabilidade para grandes ambientes. No entanto várias implementações obtiveram sucesso na aplicação do EKF SLAM (DISSANAYAKE et al., 2002). Há ainda várias variações do EKF SLAM, focando principalmente na escalabilidade, como a criação de submapas locais interconectados (ESTRADA; NEIRA; TARDÓS, 2005) (RIBAS et al., 2008), tornando o tempo de atualização do filtro dentro de um limite aceitável.

O FastSLAM, por outro lado, é naturalmente indicado para soluções onde a associação de dados seja desconhecida (MONTEMERLO; THRUN, 2003). Ele possui uma atualização eficiente quando comparado com o EKF, executando em tempo linear-logarítmico. Como ponto negativo, o número necessário de partículas para uma correta aproximação do mapa pode crescer muito, especialmente para trajetos com múltiplos *loops* entrelaçados, tornando inviável sua utilização.

Como uma alternativa aos métodos probabilísticos, na busca por soluções para suas limitações, surgem algoritmos inspirados na natureza, que serão explorados na próxima seção.

2.2.2 Abordagens Bioinspiradas

Mais recentemente surgiram novas abordagens para a resolução do problema de SLAM, afastando-se das abordagens probabilísticas. Entre elas, ganham destaque abordagens inspiradas na natureza, as quais buscam aplicar modelos comportamentais dos animais, criando-se o que se pode chamar de *animats*, uma mistura das palavras inglesas "*animals*"com "*robots*". Aliados às descobertas da neurociência, surgem métodos inspirados no comportamento cerebral de mamíferos, como por exemplo o RatSLAM.

O algoritmo RatSLAM busca aplicar modelos cerebrais do cérebro dos ratos para resolver o problema de SLAM. Criado por Milford e Wyeth (2008), o algoritmo modela sistemas cerebrais relacionados à tarefas de localização descobertos pelos neurocientistas. Entre esses neurônios, pode-se citar as *Place Cells*, células cuja atividade sináptica aumenta sempre que o rato passa em uma determinada região do ambiente.

Esse algoritmo servirá como base para o método aqui proposto, sendo melhor explicado no Capítulo 4

Como principal vantagem do RatSLAM pode-se citar a escalabilidade. A etapa de atualização da estimativa de posição é independente ao número de nós criados no mapa, dependendo apenas do número de neurônios existentes na rede neural. Além disso, a rede é capaz de trabalhar com múltiplas estimativas de posição, similar ao Filtro de Partículas, as quais competem entre si de acordo com a dinâmica da rede. Milford também demonstrou a capacidade de utilização do método na criação de mapas de longa duração (*lifelong maps*), devido à capacidade de atualização dos nodos criados no mapa. Esse método serve de base para o presente trabalho e será explicado em detalhes nas próximas seções.

Outro método bioinspirado para a resolução do problema de SLAM foi proposto por Barrera e Weitzenfeld (2008). Diferente do RatSLAM, o foco do trabalho é o desenvolvimento de um sistema cognitivo totalmente bioinspirado, servindo de base para testes de modelos da neurosciência. Os resultados obtidos foram comparados com experimentos clássicos da navegação de ratos em ambientes simples, obtendo resultados similares.

Atualmente o problema de SLAM é considerado um problema resolvido em ambientes estruturados e estáticos, porém continua um problema em aberto em ambientes mais complexos. O ambiente subaquático apresenta-se como um cenário pouco estruturado, onde a aplicação direta dos métodos existentes torna-se inviável. Os problemas variam desde a dificuldade de detectar *landmarks* robustos até a necessidade de novos algoritmos quando os sensores são diferentes.

O próximo capítulo apresentará as características dos ambientes subaquáticos,

analisando os desafios encontrados nesses ambientes e os sensores utilizados pelos robôs para navegação.

3 SLAM em Ambientes Subaquáticos

As próximas seções apresentarão os desafios encontrados no ambiente subaquático, seguido dos principais sensores utilizados nesses ambientes.

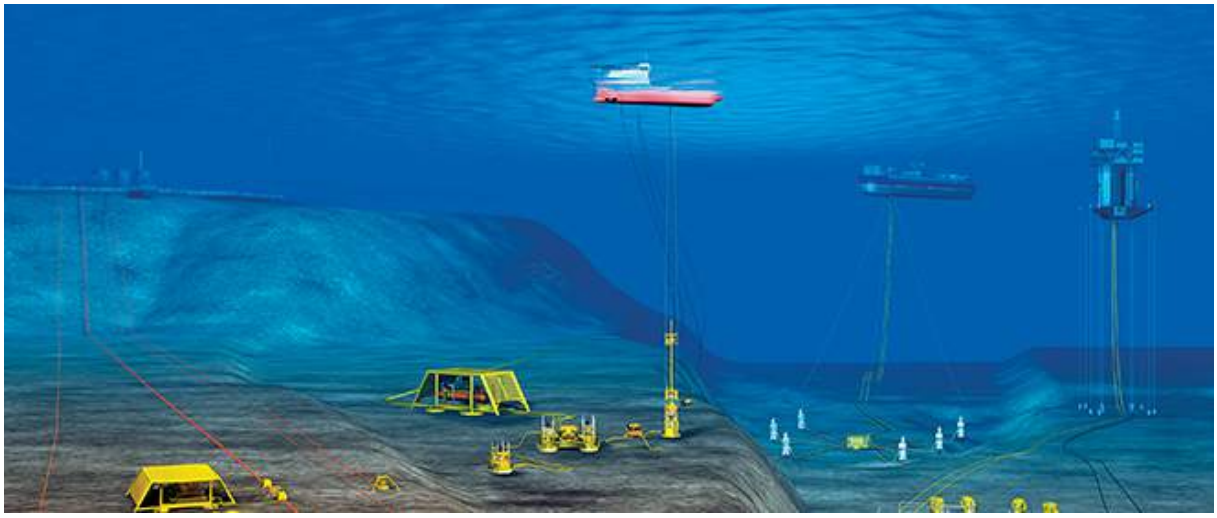

Figura 5: Típico ambiente subaquático de extração de petróleo. Fonte: (AMBIENTE..., 2014)

3.1 Desafios encontrados

O ambiente subaquático possui uma gama de novos desafios para a robótica, começando com os sensores utilizados nesses ambientes. Devido às características do meio aquático, grande parte dos sensores utilizados em robótica terrestre não são aplicáveis ou são de difícil utilização nesse cenário.

O principal motivo pela inviabilidade na utilização dos sensores nesse ambiente diz respeito às características físicas do meio. Sensores baseados em ondas de rádio, como o GPS(MASUMOTO, 1993) sofrem com a atenuação do sinal e corrente do meio.

Ainda, a aplicação de sensores baseados em ondas de luz, como câmeras e Light Amplification by Stimulated Emission of Radiation (LASER) também possuem restrições, devido a alta atenuação da luz no meio. Esses sensores são apropriados apenas para aplicações que não necessitem grande alcance(< 5m). Além disso, dependendo da profundidade de operação, há necessidade de uma fonte de iluminação acoplada ao sistema robótico.

A principal alternativa adotada nesses ambientes é a utilização de sensores baseados em ondas sonoras, as quais são facilmente propagadas no meio aquático, possuindo grande alcance e baixa atenuação.

A presença do meio líquido e sua implicação nas características perceptivas do veículo traz novos desafios para a robótica, sendo necessário o desenvolvimento de algoritmos para tratamento desses dados sensoriais, explorando as peculiaridades de cada sensor.

Para o problema em questão, o principal desafio que os algoritmos devem enfrentar diz respeito à detecção de pontos característicos robustos no ambiente. Há uma dificuldade na detecção de pistas salientes na cena, isto é, pontos característicos que possam ser identificados toda vez que estejam no campo de visão do robô. Essa questão deve ser enfrentada criando-se algoritmos mais eficientes para detecção desses pontos característicos.

3.2 Sensores utilizados

Aqui serão apresentados os principais sensores utilizados no ambiente subaquático, explicitando a informação adquirida com a utilização de cada um deles bem como suas vantagens e desvantagens. Mais informação sobre sensores utilizados nesses ambientes pode ser encontrada em (PAULL et al., 2014)

Os sensores serão divididos em três categorias: sensores de posicionamento, sensores de percepção e sensores de movimentação.

3.2.1 Sensores de Posicionamento

Os sensores de posicionamento são capazes de prover ao robô sua pose em um sistema de coordenadas fixo, bem como a orientação que o mesmo se encontra. Eles subdividem-se em sensores de localização por som, sensores de profundidade e altitude e sensores de orientação.

Os sensores de localização por som (subseção 3.2.1.1) buscam preencher a lacuna existente na robótica subaquática devido à inexistência do GPS. Seu funcionamento baseia-se na triangulação de sinais sonoros enviados por *transponders*, os quais podem ser instalados no fundo do mar, na superfície da água ou no casco de navios.

Os sensores de profundidade e altitude (subseção 3.2.1.2) são responsáveis pela localização do robô no eixo vertical. Nessa classificação existem sensores para medir a distância do robô até a superfície e a distância do robô até o fundo do mar.

Os sensores de orientação (seção 3.2.1.3) são responsáveis por uma estimativa de orientação do robô no ambiente, podendo ser absoluta, quando baseada no eixo de rotação da terra, ou relativa, no caso dos giroscópios.

3.2.1.1 Sensores de Localização por Som

Devido a alta atenuação das ondas de rádio abaixo da superfície da água, sistemas de posicionamento global como o GPS não podem ser utilizados. Como alternativa, existe

a possibilidade de implantação de um sistema de posicionamento no local da operação. Abaixo serão detalhadas as principais tecnologias existentes para estimar o posicionamento subaquáticos.

USBL

Ultra Short Baseline Positioning System (USBL) é um método de posicionamento acústico subaquático, mostrado na Figura 6. É composto por um *transciever*, normalmente montado no casco de um navio e um *transponder*, instalado na plataforma robótica. Cada *transciever* possui pelo quatro *transducers* para captação do sinal de resposta do robô. A posição do robô é então calculada com base no tempo da propagação da onda sonora entre o robô e o navio.

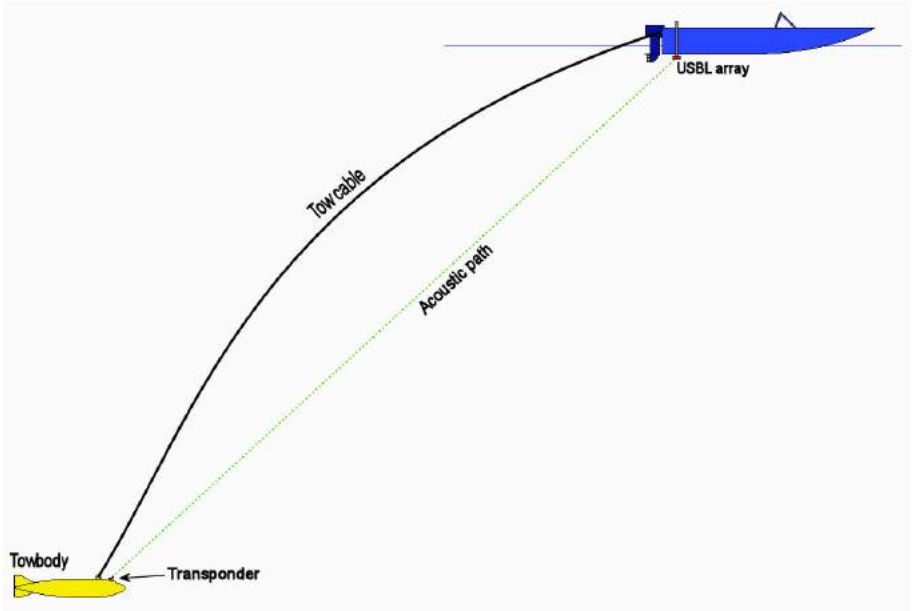

Figura 6: Sistema USBL. O transponder fixado ao navio envia um sinal sonoro, que é respondido pelo robô e captado pelos transdutores no navio. A localização é calculada com base no tempo de resposta. Fonte: (USBL. . . , 2014)

Essa configuração é normalmente utilizada por ROVs, em que a posição deve ser conhecida pela unidade central de controle, localizada no navio. Para a utilização desse sistema por AUVs, os equipamentos são invertidos, de modo que o *transciever* esteja acoplado ao robô, permitindo ao mesmo a descoberta de sua localização.

O nome USBL deve-se à configuração dos *transdutores*, os quais estão separados por uma curta distância (baseline), de aproximadamente $10cm$. Esse sistema possui uma precisão de aproximadamente $1m$, não sendo indicado para tarefas de alto risco, como operações abaixo do gelo e instalação de poços de extração de petróleo.

SBL

Short Baseline Positioning System (SBL) é outro método de posicionamento acústico subaquático, mostrado na Figura 7. Esse método utilizada três ou mais *transponders* distribuídos na superfície da água, ou presos ao casco do navio. A posição do veículo é calculada com base no tempo de resposta entre a onda emitida por cada *transponder* e o veículo, utilizando uma técnica de triangularização, similar a um sistema de GPS.

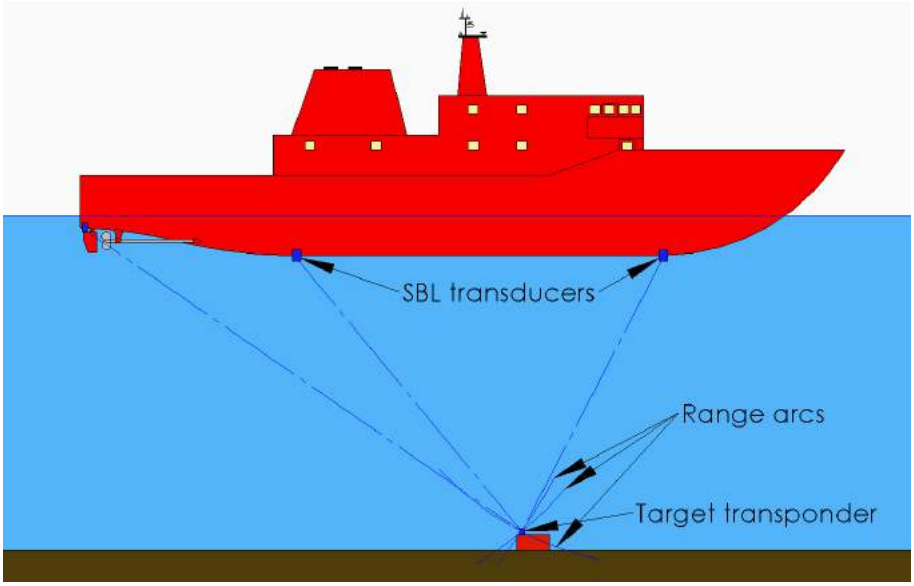

Figura 7: Sistema SBL. Vários transponders são fixados ao casco do navio, os quais enviam sinais sonoros, que são respondidos pelo robô e captados pelos transdutores no navio. A localização é calculada com base no tempo de resposta. Fonte: (USBL. . . , 2014)

A baseline(distância entre os *transponders*) é maior que a utilizada pela USBL, limitando-se normalmente a algumas dezenas de metros, principalmente quando presas ao casco do navio. A precisão da localização calculada por esse sistema não é fixa e aumenta com o número de *transponders* utilizados e a distância entre eles. Dependendo da configuração utilizada, esse sistema pode ser utilizado para tarefas que exijam precisão.

LBL

Uma terceira tecnologia para localização amplamente utilizada na robótica subaquática é a Long Baseline Positioning System (LBL), mostrada na Figura 8. Consiste em um conjunto de *transponders* espalhados no fundo do oceano, cercando a área onde o sistema robótico realizará tarefas. Nesse sistema, a distância entre os *transponders* é maior que a distância entre o robô e cada um dos *transponders*.

Sua precisão é maior quando comparado com os anteriores, cerca de $10cm$, por isso sua ampla aplicação pela indústria na instalação de equipamentos no fundo do oceano.

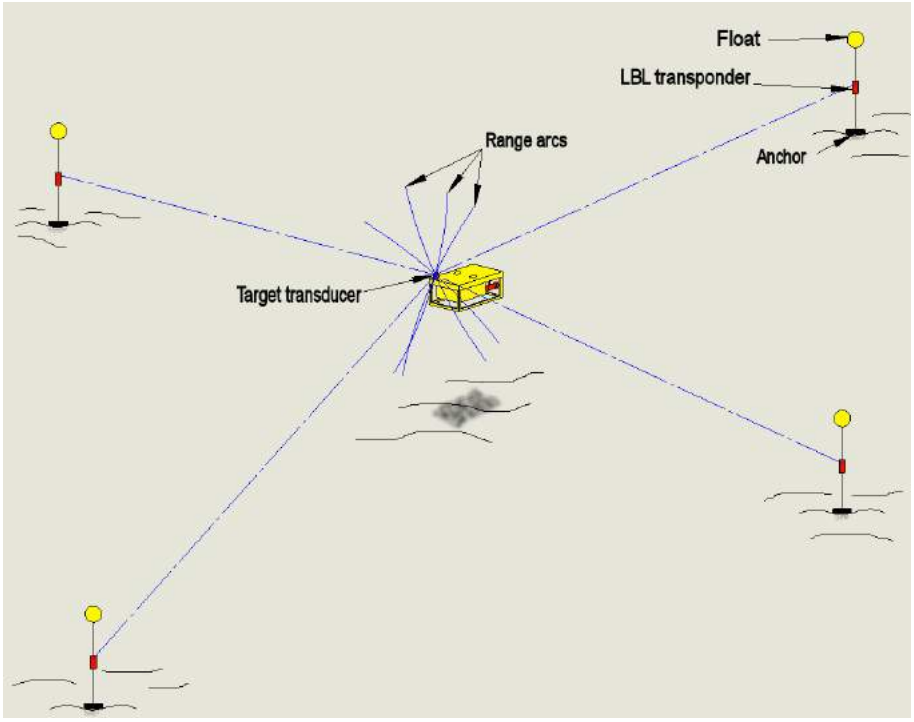

Figura 8: Sistema LBL. Vários *transponders* são colocados no fundo do mar, os quais respondem as ondas sonoras enviadas pelo robô, permitindo que o robô calcule sua posição com base no tempo de resposta de cada um dos transdutores. Fonte: (USBL. . . , 2014)

No entanto, esse sistema possui uma maior dificuldade de instalação, pois necessita a colocação de *transponders* no fundo do mar, seguido de uma etapa de georreferenciamento de cada *transponder*.

3.2.1.2 Sensores de Profundidade e Altitude

A seção anterior listou as principais tecnologias para localização absoluta de robôs no ambiente aquático. Esses sistemas provêem uma localização nos três eixos coordenados. No entanto, existem soluções com custo menos elevado para a descoberta da profundidade e altitude que o robô encontra-se. Essas soluções são listadas a seguir.

Altímetro

O altímetro é um sensor sonoro, instalado no robô de modo a enviar uma onda em direção ao fundo do oceano (Figura 9). Esse equipamento mede o tempo de retorno da onda a fim de estimar a distância do robô até o fundo do mar.

Esse equipamento possui fácil instrumentação, não necessitando a colocação de *transponders* no ambiente.

Figura 9: Exemplos de altímetros comerciais para utilização em robótica subaquática

Sensor de pressão

O sensor de pressão é um equipamento de baixo custo para medir a profundidade do robô, isto é, a quantidade de coluna d'água acima dele. As tecnologias mais comuns para leitura da pressão da coluna d'água são medidores de tensão e cristais de quartzo (KINSEY; EUSTICE; WHITCOMB, 2006).

As principais vantagens desse dispositivo são seu baixo custo e o aumento linear da pressão no meio aquático de acordo com a profundidade do robô.

3.2.1.3 Sensores de Orientação

Os sensores de orientação são responsáveis por calcular a orientação absoluta do robô. São também chamados de sensores de atitude.

Bússola

A Bússola, ou magnetômetro, é um equipamento magnético capaz de medir o vetor do campo magnético da terra. Esse equipamento possui precisão superior a 1^0. Como restrição, esse sistema sofre interferência de forças magnéticas geradas pelo próprio robô ou por equipamentos que estejam operando nas proximidades, o que acarreta em erros de medição. A Figura 10 apresenta um exemplo de bússola.

Giroscópio

Similar a bússola, esse dispositivo também mede a orientação do robô. Essa orientação pode ser medida em relação a um, dois ou três eixos coordenados. Existem três tecnologias comerciais capazes de fazer a leitura da orientação, cada uma com suas vantagens e desvantagens.

O giroscópio rotacional utiliza um disco em alta rotação e a rotação da Terra para para estimar a orientação. Não é afetado por forças magnéticas externas, no entanto está

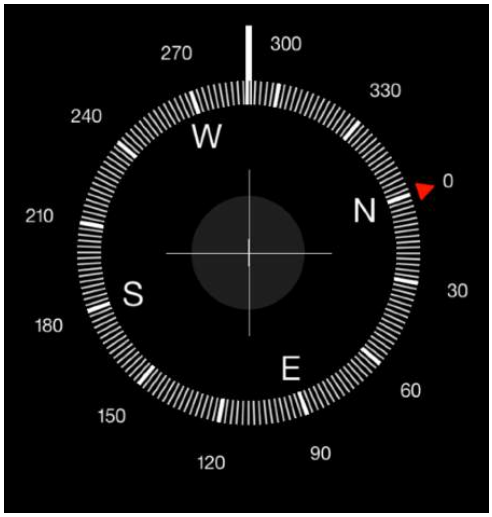

Figura 10: Imagem ilustrativa de uma bússola digital de um eixo.

sujeito a um erro de medição crescente com o tempo, havendo a necessidade de *reset* após certo período de operação.

O segundo giroscópio utiliza a vibração de estruturas mecânicas de modo a obter uma leitura de orientação. Esse sistema é facilmente encontrado comercialmente, podendo ser produzido em tamanho reduzido devido a ausência de partes móveis, porém ainda há a existência de erros devido ao tempo de operação.

O terceiro tipo consiste em um giroscópio ótico, o qual não contém nem partes móveis, não sendo suscetível de erros por tempo de operação. Sua precisão depende apenas da constância da velocidade da luz. Por esse motivo esse dispositivo é amplamente utilizado na robótica, estando presente na maioria dos robôs móveis subaquáticos.

3.2.2 Sensores Proprioceptivos

Como já citado anteriormente, a falta de um sistema de posicionamento global no meio subaquático faz com que se busquem soluções alternativas. Os sistemas de localização por som desenvolvidos limitam a operação do robô a pequenas áreas ou a curtas distâncias do navio de acompanhamento. Como alternativa, existem dispositivos capazes de adquirir informações sobre a movimentação do robô, como aceleração e velocidade. Esses dispositivos serão detalhados abaixo.

3.2.2.1 Unidade de Medição Inercial

Uma unidade de medição inercial (Fig. 11), conhecida por Unidade de Medição Inercial (IMU) é um dispositivo capaz de prover informações acerca da aceleração linear do robô e de sua velocidade angular. Uma IMU é normalmente composta por um conjunto de acelerômetros, giroscópios e, em alguns casos, magnetômetros.

Figura 11: IMU Xsens, uma opção comercial para unidade de medição inercial.

Esse dispositivo pode ser utilizado para localizar o robô em um sistema fixo na posição inicial da trajetória, integrando as informações de aceleração de acordo com a orientação instantânea do robô, adquirida pelos giroscópios e magnetômetros. A principal desvantagem desse sistema está no erro inerente da integração, crescendo ao longo do deslocamento do robô.

3.2.2.2 Doppler Velocity Log

Doppler Velocity Log (DVL) é um equipamento capaz de medir a velocidade que o veículo está se deslocando, por meio da propagação de ondas sonoras. Esse dispositivo emite ondas em quatro ou mais direções, e mede o retorno da onda. Esse retorno pode ser provocado a partir de partículas na água ou utilizando o fundo do oceano. A última opção possui uma maior precisão, no entanto necessita que o robô desloque-se a uma distância máxima de $100m$ do fundo do oceano.

Figura 12: Exemplo de uma DVL comercial. (DVL. . . , 2014)

Esse sensor possui uma precisão superior à IMU para a localização do robô, no entanto também possui um erro crescente ao longo do deslocamento. A DVL e a IMU

são normalmente utilizadas em conjunto para a criação de um sistema de navegação para AUVs.

3.2.3 Sensores Exteroceptivos

3.2.3.1 Câmera

Um dos sensores mais utilizados em aplicações robóticas inteligentes é a câmera, um instrumento ótico que captura imagens formadas por feixes de fótons na frequência visível do ambiente observado. Como ponto forte no emprego desse sensor está o baixo custo na sua aquisição, havendo vários modelos disponíveis comercialmente, o que facilita sua utilização na robótica subaquática.

A principal desvantagem na utilização desses sensores é seu baixo alcance em meio subaquático, devido à alta atenuação das ondas de luz nesse meio físico. Com isso, torna-se possível apenas para utilização em tarefas em que a distância entre os objetos observados e o robô seja curta, como por exemplo em inspeções de equipamentos na indústria de óleo e gás e para captura da vida marinha existente no fundo dos oceanos. No entanto, frisa-se que sua utilização por robôs autônomos está atrelada à capacidade do mesmo processar as informações de forma online, pois não é simples o envio de vídeo sem fio entre o robô e uma unidade de controle na superfície.

Normalmente os robôs são equipados com uma ou mais câmeras, com diferentes campos de visão. Quando há a existência de apenas uma lente por câmera, trata-se de uma câmera monocular (Figura 13). Quando possui mais de uma lente, separadas por uma distância lateral entre elas, trata-se de uma câmera estereoscópica (Figura 13). Essa última é capaz de inferir a distância dos objetos existentes na cena quando previamente calibrada. Um câmera estereoscópica também pode ser formada por duas câmeras monoculares com sincronia na captura (Figura 13).

Figura 13: Exemplos de cameras empregadas em robótica subaquática.

3.2.3.2 LASER

O LASER é um equipamento capaz de captar distâncias dos objetos existentes ao redor do robô. Apesar de ser amplamente utilizado em robótica terrestre, sua aplicação em ambientes subaquáticos é limitada devido ao uso de ondas de luz em sua operação. Essa característica limita o alcance de detecção dos sensores.

Existem equipamentos específicos para o ambiente subaquático, os quais fazem um tratamento dos ruídos inerentes ao fenômeno de *backscaterring* existente na propagação da onda de luz na água, sendo capaz de reduzir o erro na medição causados pelo meio. No entanto, a aplicação desses sensores é limitada para tarefas cuja distância dos objetos seja de poucos metros.

Sua utilização não é tão difundida devido ao alto custo de aquisição desses equipamentos. Um exemplo de um laser existente no mercado é mostrado na Fig. 14.

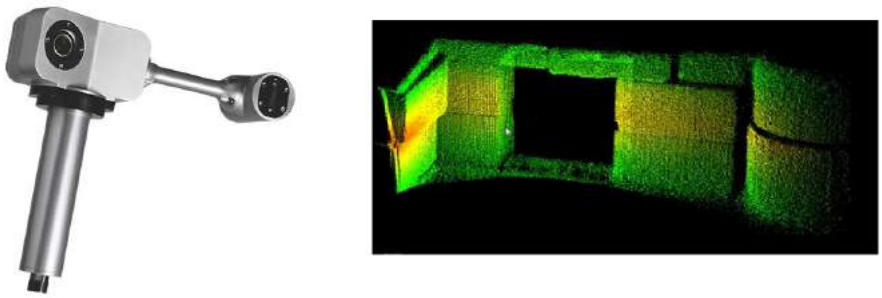

Figura 14: Laser Subaquático. Imagem extraída de (LASER..., 2014)

3.2.3.3 Sonar

Inúmeras tecnologias foram desenvolvidas para aquisição de informações do ambiente subaquático navegado com o uso de ondas sonoras. Essas tecnologias variam desde a descoberta de obstáculos a frente do veículo até a criação de mapas batimétricos, com uma completa definição dos tipos de sedimentos encontrados no ambiente. No contexto da robótica subaquática, foram criados equipamentos a serem integrados nos veículos, desde sensores específicos para navegação até sensores para realização de análises oceanográficas.

Esses sensores subdividem-se em dois grupos: Sonares de Distância e Sonares de Imageamento. A primeira categoria provê informações sobre a distância dos objetos a partir da leitura da onda emitida. A segunda categoria é capaz de criar uma imagem do ambiente escaneado, a partir do envio de múltiplas ondas.

A seguir serão apresentados as principais tecnologias comerciais existentes.

Single Beam Sonar

Esse tipo de sonar emite apenas um pulso sonoro por vez. A partir do tempo entre o envio da onda e seu recebimento é calculada a distância entre o robô e os objetos. A Figura 15 apresenta o funcionamento do dispositivo. O altímetro, apresentado na subseção 3.2.1.2, utiliza essa tecnologia para calcular a distância que o veículo encontra-se do fundo do oceano.

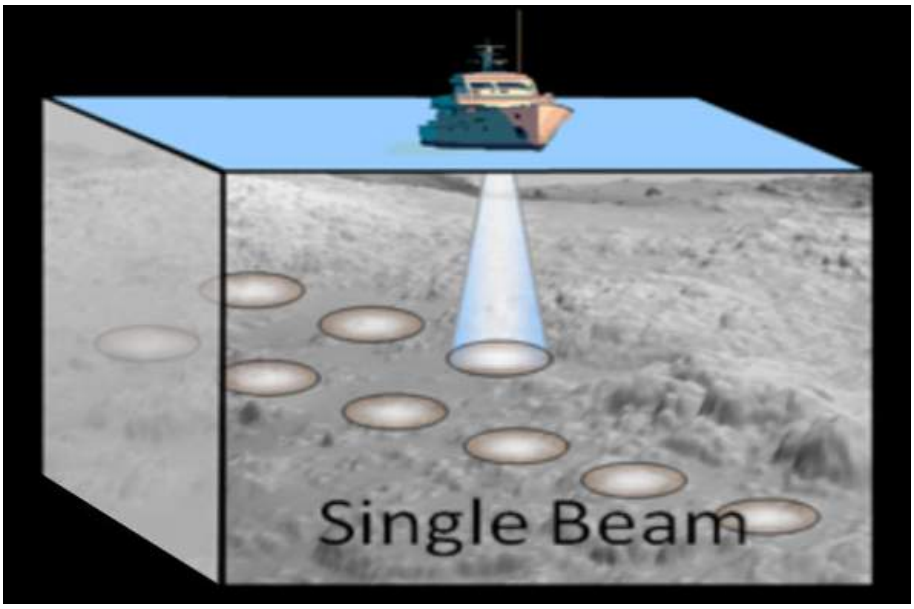

Figura 15: Single Beam Sonar. Imagem extraída de (SONARES. . . , 2014)

Multi Beam Sonar

Esse tipo de sonar emite vários pulsos sonoros simultaneamente, em direção ao fundo do mar. Cada pulso é emitido com um ângulo de inclinação diferente, possibilitando a aquisição de todo o relevo abaixo do robô em uma única leitura. Esse dispositivo é amplamente utilizado para criação do perfil do fundo do mar e pode ser visto na Figura 16.

Side Scan Sonar

Esse tipo de sonar emite duas ondas sonoras, uma de cada lado do equipamento, de modo a fazer uma leitura do fundo do oceano. Esse sonar possui a característica de detectar o tipo de solo existente no fundo do oceano de acordo com a intensidade da onda retornada. Ele pode ser visto na Figura 17. Esse sonar é amplamente utilizado para analisar o tipo de solo ou objeto existente no fundo do oceano.

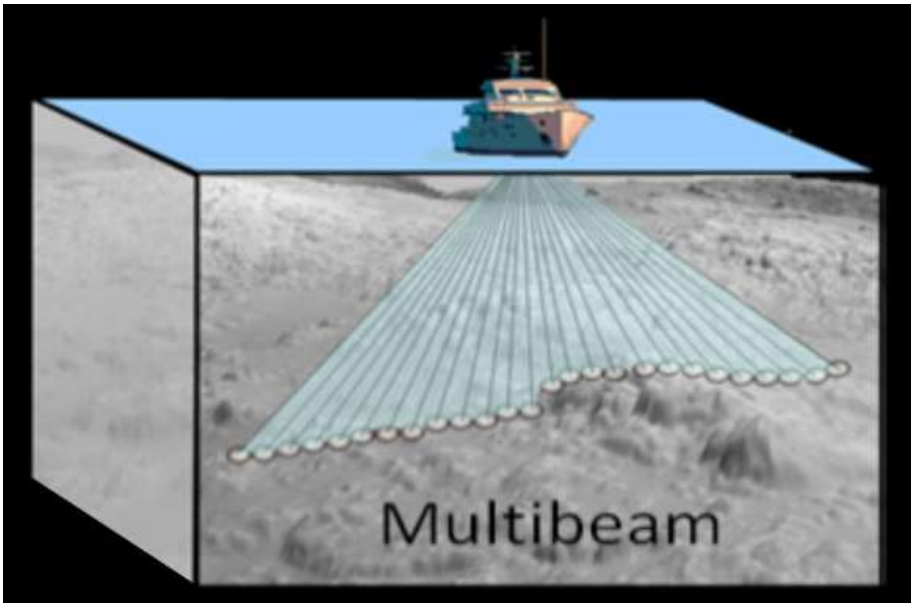

Figura 16: Multi Beam Sonar. Imagem extraída de (SONARES. . . , 2014)

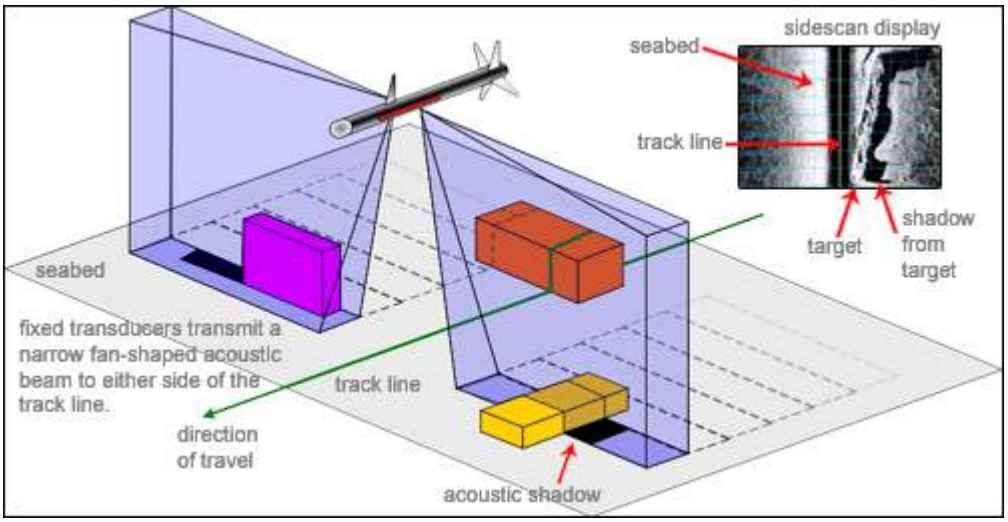

Figura 17: Funcionamento do Sonar do tipo Sidescan. O sonar envia dois feixes sonoros laterais para fazer uma varredura do fundo do mar. Imagem extraída de (SONAR. . . , 2014b)

Mechanically Imaging Sonar

Esse sonar é capaz de criar um perfil dos objetos existentes ao redor do veículo. Ele cria uma imagem a partir do retorno dos pulsos emitidos pelo dispositivo. Os pulsos são emitidos um por vez, variando o ângulo de envio da onda sonora, de modo que uma onda não haja interferências entre as ondas. Uma varredura capturada por um sonar do tipo mecânico pode ser visto na Figura 18. Esse tipo de sonar tem sido bastante aplicado para resolução do problema de SLAM (WILLIAMS et al., 2000)(TENA et al., 2001)(MALLIOS et al., 2010)(RIBAS et al., 2008).

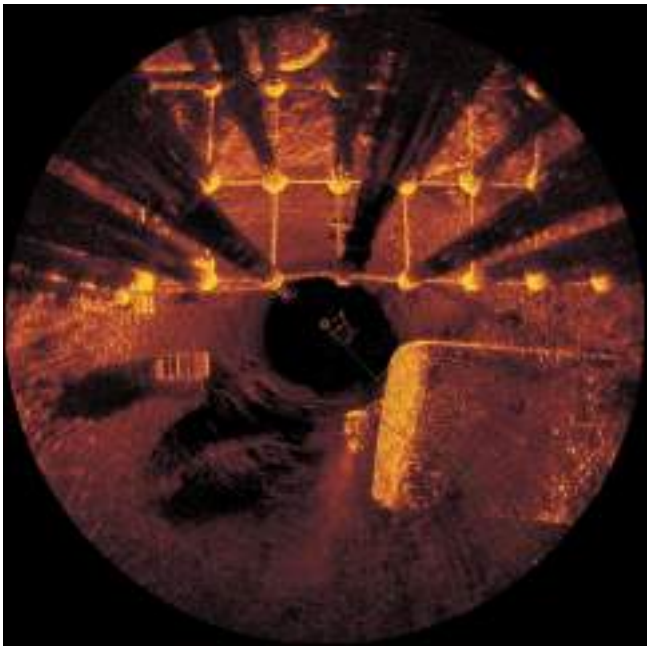

Figura 18: Imagem adquirida com um sonar de imageamento mecânico. (SONAR. . . , 2014a)

Forward Looking Imaging Sonar

Esse sonar é capaz de criar uma imagem do fundo do mar, como mostrado na Figura 19. Essa imagem pode ser processada para descoberta do tipo de solo existente no fundo do oceano, bem como para detecção de objetos. Essa tecnologia tem sido utiliza por métodos para SLAM nos últimos anos. (LEONARD; CARPENTER; FEDER, 2001)(WALTER; HOVER; LEONARD, 2008)

Figura 19: Sonar do tipo Forward Looking comercial e exemplo de imagem capturada. (TELEDYNE. . . , 2014)

3.3 Estado da Arte em SLAM Subaquático

A primeira solução para o problema de SLAM Subaquático validada com dados reais foi apresentada por pesquisadores da Universidade de Sydney, na Austrália, no ano de 2000. Williams et al. (2000) apresentaram um algoritmo que detecta cilindros colocados no ambiente a partir da leitura de um sonar mecânico de imagem. A posição dessas *features* é mapeada, juntamente com a posição atual do robô usando um Filtro de Kalman Estendido (EKF) . O robô utilizado para aquisição dos dados foi o AUV Oberon, equipado com um sensor de pressão e um giroscópio de fibra ótica com três eixos. O robô utiliza esses sensores em conjunto com o sonar para a criação de um mapa 3D do ambiente. A principal desvantagem é o custo computacional da abordagem, possuindo complexidade quadrática ao número de *features* no ambiente, limitando sua aplicação para ambientes com poucas features.

Um ano depois, Leonard, Carpenter e Feder (2001), do Massachusetts Institute of Technology, USA, apresentou um algoritmo utilizando um sonar de imagem do tipo *Forward Looking* de onde eram extraídos *features* pontuais para a criação de um mapa baseado no Filtro de Kalman Extendido. O algoritmo é testado com dados reais de uma expedição, em que os sensores comuns aos robôs subaquáticos foram montados em uma estrutura na lateral de um navio. O algoritmo utilizou dados de DVL e IMU para estimativa de navegação e Sistema de Posicionamento Global Diferencial (DGPS) para ground truth. Os resultados foram comparados com *dead reckoning*, obtendo uma melhor estimativa de posicionamento do robô. O método possui as mesmas restrições de escalabilidade encontradas por Williams et al. (2000).

No mesmo ano, Tena et al. (2001) desenvolveu um método para extração de *features* em imagens de sonar mecânico e a posterior associação desses dados utilizando o algoritmo *Multiple Hypothesis Tracking Filter - MHTF*, para criação do mapa utilizando um EKF. O algoritmo MHTF mostrou-se inviável devido ao custo exponencial ao número de hipóteses utilizadas como possíveis associações.

Em 2003, Paul Newman e seus colegas do Massachussets Institute of Technology (MIT) publicaram dois trabalhos em SLAM subaquático sobre dados capturados com o AUV Caribou, em uma expedição em águas rasas na costa da Itália. No primeiro trabalho, Newman e Leonard (2003) apresentaram um método para mapeamento e localização utilizando apenas dados de distância dos *transponders* de um sistema LBL. O método não possuía informação a priori sobre as posições dos *transponders* nem da posição do robô, resolvendo o problema tanto para a localização do robô quanto para a localização dos *transponders*, apresentando uma boa abordagem para auto calibração de um sistema LBL. No entanto, como apresentado pelos autores, o método apresentava uma grande sensibilidade à correta inicialização das posições dos *transponders*.

No segundo trabalho, Newman, Leonard e Rikoski (2003) apresentaram uma proposta para SLAM em tempo constante com a utilização de EKF com limite de *features* e criação de submapas locais. O algoritmo extrai *features* pontuais de um Sonar de Abertura Sintética, no entanto a associação de dados é feita de forma manual. Os resultados foram comparados com um EKF completo, obtendo resultados similares com a vantagem de execução em tempo constante.

Em 2004, Ruiz, Petillot e Lane (2004) apresentaram um método para SLAM utilizando Sidescan Sonar com EKF. A extração de *features* é realizada de forma manual, bem como a associação de dados. A utilização desse tipo de sonar só é viável quando o robô passa mais de uma vez em uma determinada trajetória, possibilitando a ocorrência de fechamento de *loop*.

No mesmo ano, Williams e Mahon (2004) desenvolveram um método capaz de fazer fusão sensorial de dados de sonar com imagens de uma câmera ótica. As *features* são detectadas no sonar e projetadas na imagem da câmera, sendo rastreadas frame a frame com o algoritmo de fluxo ótico Lucas-Kanade. O método não trata fechamento de *loop* de grande distâncias.

Ainda em 2004, Eustice, Pizarro e Singh (2004) apresentam um algoritmo para SLAM visual capaz de estimar a posição do robô nos seis graus de liberdade, onde obtiveram excelentes resultados aplicando um EKF com todo o histórico de posições percorridas pelo robô. Esse algoritmo foi chamado de Visual Augmented Navigation (VAN).

Roman e Singh (2005) apresentam um método também baseado no Filtro de Kalman Estendido com a estimativa de todo o percurso navegado pelo robô. A plataforma utilizada foi o ROV JASON, do MIT, equipado com um sonar do tipo Multibeam para aquisição de dados batimétricos do ambiente e sensores de navegação, como DVL, sensores de atitude e pressão. O algoritmo implementa a criação de submapas batimétricos, possuindo uma maior escalabilidade quando comparado a apenas um mapa global. A detecção de *loops* é feita a partir da comparação dos dados recebidos do sonar. Os dados foram validados comparando-se os resultados com um sistema LBL e *dead reckoning*.

Em 2006, em uma cooperação entre Espanha e Canadá, Sàez et al. (2006) publicaram um trabalho explorando odometria visual no robô AQUA, equipado com câmera stereo. O método cria um mapa denso com as imagens, aplicando algoritmo de minimização de entropia para melhorar a qualidade do mapa. Apesar de obter resultados favoráveis, o algoritmo possui um alto custo computacional, estando limitado a execuções offline devido ao processo de otimização do mapa.

No mesmo período, Olson, Leonard e Teller (2006) estenderam o método desenvolvido em Newman e Leonard (2003), tratando as limitações de inicialização de posicionamento dos *transponders* LBL. Para isso, realizam uma filtragem de ruído na

leitura de distância dos transponders, removendo outliers e fazendo uma votação para a possível posição do *transponder*. Essa posição é utilizada para inicialização de posicionamento do *transponder* no EKF, superando a limitação existente no método anterior e resolvendo o problema de localização tanto do robô quanto dos *transponders*.

Em 2007, com o objetivo de mapear uma formação rochosa imersa em água no México, pesquisadores da Carnegie Mellon University, EUA, desenvolveram um método baseado em grades de ocupação 3D utilizando Octrees para possibilitar a execução em tempo real(FAIRFIELD; KANTOR; WETTERGREEN, 2007). O método estende o algoritmo FastSLAM baseado em grade de ocupação para ambientes 3D. A aquisição dos dados é feita com um sonar personalizado com 54 beams direcionais distribuídos ao longo de um cilindro, de modo a varrer toda a lateral do robô. No entanto, devido ao seu custo computacional, sua aplicação não é viável para navegações longas.

Eustice, Pizarro e Singh (2008) publicaram uma extensão do algoritmo desenvolvido em (EUSTICE; PIZARRO; SINGH, 2004), detalhando melhor o método e aplicando-o em outros ambientes. Apresentam a utilização do algoritmo *Bag of Words* para detecção de *loops* sobre as imagens e odometria visual para cálculo do deslocamento entre duas posições percorridas pelo robô. O problema de complexidade quadrática do Filtro de Kalman é tratado no trabalho de Mahon et al. (2008), onde o EKF é substituído por um Filtro de Informação Estendido, reduzindo a complexidade computacional do algoritmo.

No mesmo ano, na Espanha, Ribas et al. (2008) apresentaram um método para SLAM em ambientes semi-estruturados feitos pelo homem, utilizando um sonar de imagem mecânico, explorando características geométricas desses ambientes. Como plataforma de testes foi utilizado o robô ICTNEU, equipado com DVL, Giroscópios e DGPS, esse último utilizado como *ground truth*. A imagem do sonar era processada com algoritmo *Hough* para detecção de linhas, as quais são inseridas no mapa. O algoritmo utilizado baseava-se no Filtro de Kalman Estendido, com criação de submapas para redução do custo computacional. Os resultados demonstraram a vantagem do método frente ao *dead-reckoning*.

Ainda em 2008, Walter, Hover e Leonard (2008) apresentaram um método para inspeção de cascos de embarcações utilizando o algoritmo *Exactly Sparse Extended Information Filter*. *Landmarks* foram adicionados ao ambiente, sendo capturados com o uso de um sonar de imagem do tipo *Forward Looking*, com associação de dados feita de forma manual.

Em 2010, Mallios et al. (2010) desenvolveram um método baseado em *scan matching* probabilístico sobre a imagem de um sonar mecânico. O algoritmo foi testado no mesmo *dataset* utilizado por Ribas et al. (2008), obtendo uma precisão superior nesse ambiente.

No próximo ano, Burguera, Gonzàlez e Oliver (2011) exploraram a criação de um

EKF com o mapa centrado na posição atual do robô, objetivando uma diminuição do erro de fechamento de *loops*. Os resultados apresentaram uma maior precisão quando comparado a criação de mapas referenciados em um ponto do ambiente.

Em 2012, Ferreira et al. (2012) exploram a criação, em tempo real, de um mosaico do fundo do mar. O método utiliza o algoritmo Speed-Up Robust Features (SURF) (BAY; TUYTELAARS; GOOL, 2006) para detecção de *features* nas imagens, eliminando falsos positivos na associação dos dados. Além disso, para alcançar execução em tempo real, os autores criam pequenos mosaicos locais, que são unidos para a criação de um mosaico global.

Nesse mesmo ano, Barkby et al. (2012) desenvolveram um SLAM batimétrico utilizando Filtro de Partículas *Rao-Blackwellized* e regressão de processos Gaussianos. O vetor de estados de cada partícula armazena toda a trajetória do robô, sendo que os resultados obtidos foram semelhantes à aplicação de um filtro de partículas com gride de ocupação, porém a um menor custo computacional e uma menor utilização de memória.

Em 2013, Kim e Eustice (2013) apresentaram um método interessante para SLAM Visual. Para superar problemas com baixa visibilidade e *features* limitadas, cenário comum no ambiente subaquático, as imagens capturadas são processadas com uma equalização de contraste e posteriormente é calculado o grau de saliência existente na imagem, com relação ao frame anterior armazenado. Essa métrica utiliza-se do algoritmo *Bag of Words* (SIVIC; ZISSERMAN, 2003) e é responsável pela identificação de *keyframes*, para registro no algoritmo de SLAM. Após essa etapa é calculada a transformação entre as imagens selecionadas, o que é injetado no algoritmo iSAM, um método de SLAM gráfico de tempo real.

A grande maioria dos trabalhos desenvolvidos na área utiliza uma abordagem probabilística para a resolução do problema, utilizando Filtros, entre eles o EKF, Filtro de Partículas e o Filtro de Informação Estendido (EIF). Para contornar a limitação de escalabilidade, a maioria dos autores busca a redução no número de *landmarks* armazenados no mapa ou pela criação de vários submapas. Existem ainda algumas soluções promissoras utilizando SLAM gráfico baseados na otimização eficiente da trajetória percorrida pelo robô. Já existem soluções em tempo real, como utilizado por Kim e Eustice (2013).

Entre os sensores existe uma prevalência na utilização de sonares, principalmente pela capacidade de capturar informações navegando longe do fundo do oceano. No entanto, para tarefas de inspeção e intervenção, muitos pesquisadores tem optado pela utilização de sensores óticos.

O próximo capítulo apresentará os métodos tomados como base para o presente trabalho.

4 Métodos

O presente capítulo abordará os dois métodos que serviram de base para o desenvolvimento do presente trabalho. Primeiramente será detalhado o funcionamento do algoritmo RatSLAM, desenvolvido por Milford e Wyeth (2008). Logo em seguida será explicado o método FAB-MAP, desenvolvido por Cummins e Newman (2008).

4.1 RatSLAM

Milford e Wyeth (2008) buscaram inspiração na natureza para propor uma nova forma de resolver o problema de SLAM na robótica. Encontraram na neurosciência estudos de mecanismos cerebrais dos mamíferos, os quais estão relacionados a tarefas de navegação. Com base nos estudos dos neuroscientistas, criaram um sistema capaz de mapear ambientes desconhecidos e localizar o robô nesses ambientes.

4.1.1 Mecanismos cerebrais encontrados nos mamíferos

Questões relacionadas a forma como os seres vivos percebem o ambiente ao seu redor para localizar-se são objeto de estudo há centenas de anos (MCNAUGHTON et al., 2006). Entre os animais estudados, estão os ratos, morcegos, macacos, entre outros. O resultado desses estudos foi a descoberta de neurônios especiais relacionados à representação do espaço no cérebro desses mamíferos.

4.1.1.1 Neurônios espaciais

As primeiras descobertas remontam à década de 1970, em que O'Keefe e Dostrovsky (1971) reportaram campos receptivos espaciais em neurônios no Hippocampo de ratos, onde há uma predominância de células do tipo piramidal. Esses neurônios possuíam a pecualiaridade de aumentar sua taxa de disparo de acordo com a posição do rato no ambiente. Em virtude dessa relação com a localização do animal no ambiente, essas células foram denominadas *Place Cells*.

A região do ambiente onde ocorre um aumento da taxa de disparo é chamada de campo de disparo da célula. A *Place Cell* possui apenas um campo de disparo por ambiente. No entanto, ao trafegar de um ambiente para outro, as *Place Cells* são remapeadas, criando novos campos de disparo. Essas células são ancoradas ao ambiente por meio de pistas sensoriais (landmarks). Quando há uma movimentação nas características visuais do ambiente, como por exemplo a rotação das paredes, os campos de disparo das *Place Cells* sofrem a mesma rotação (MOSER; KROPFF; MOSER, 2008).

Uma outra característica interessante diz respeito à manutenção dos campos de disparo na ausência de informações sensorias externas. Quando o rato está navegando em um ambiente e a luz é apagada, os campos de disparo das *Place Cells* são mantidos por alguns minutos, implicando que o mamífero utiliza informações de movimentação própria como entrada nos neurônios. Em 1978, O'Keefe e Nadel (1979) propuseram que as *Place Cells* são os elementos básicos de um mapa topológico não centrado e distribuído.

Em 1990, gravações das taxas de disparo de neurônios da região do *Postsubiculum* de roedores levaram a descoberta de células relacionadas à orientação da cabeça do mamífero (TAUBE; MULLER; JR, 1990). Por esse motivo, essas células foram chamadas de *Head Direction Cells*.

As *Head Direction Cells*, de forma similar às *Place Cells*, são ancoradas ao ambiente por meio de *landmarks* visuais. Ainda, cada *Head Direction Cell* mantém seu campo de disparo relativo às outras *Head Direction Cells* (MCNAUGHTON et al., 2006). Essas células não possuem nenhum comportamento relacionado com a posição do animal no ambiente, apenas à orientação do mesmo.

Algun anos mais tarde, novos experimentos realizados em ambiente maiores ($\geq 1m^2$) levaram à descoberta de neurônios com múltiplos campos espaciais de disparo (HAFTING et al., 2005). Essas células foram encontradas principalmente no *Entorhinal Cortex* dos mamíferos. Seu padrão de disparo possui um formato de grade triangular, a qual cobre todo o ambiente explorado pelo animal, motivo pelo qual essas células foram chamadas de *Grid Cells*.

As *Grid Cells* aumentam seu campo de disparo sempre que o mamífero cruza os vértices de uma grade triangular, diminuindo seu disparo quando o mesmo afasta-se dos vértices. Cada célula possui três características que as diferenciam: Orientação da grade, espaçamento da grade e fase (MOSER; KROPFF; MOSER, 2008). O espaçamento da grade tende a aumentar de acordo com a posição da célula no cérebro dos ratos, formando grades menores na região ventral e maiores na região dorsal (MCNAUGHTON et al., 2006). A formação em grade sugere a participação desses neurônios em um sistema métrico de representação do ambiente (HAFTING et al., 2005), similar às *Place Cells* encontradas no Hippocampo.

A Figura 20 apresenta um esquema representativo do campo de disparo das células de acordo com a posição e orientação do rato no ambinete. A primeira célula (a), representa o comportamento ideal de uma *Place Cell*, com apenas um campo de disparo no ambiente e nenhuma relação com a orientação da cabeça do animal. A segunda célula (b) é uma *Grid Cell*, apresentando uma padrão de disparos em formato de uma grade triangular, sem qualquer relação com a orientação do rato no ambiente. Por último (c), é apresentada uma *Head Direction Cell*, cujo padrão de disparo só está relacionado a orientação da cabeça do rato.

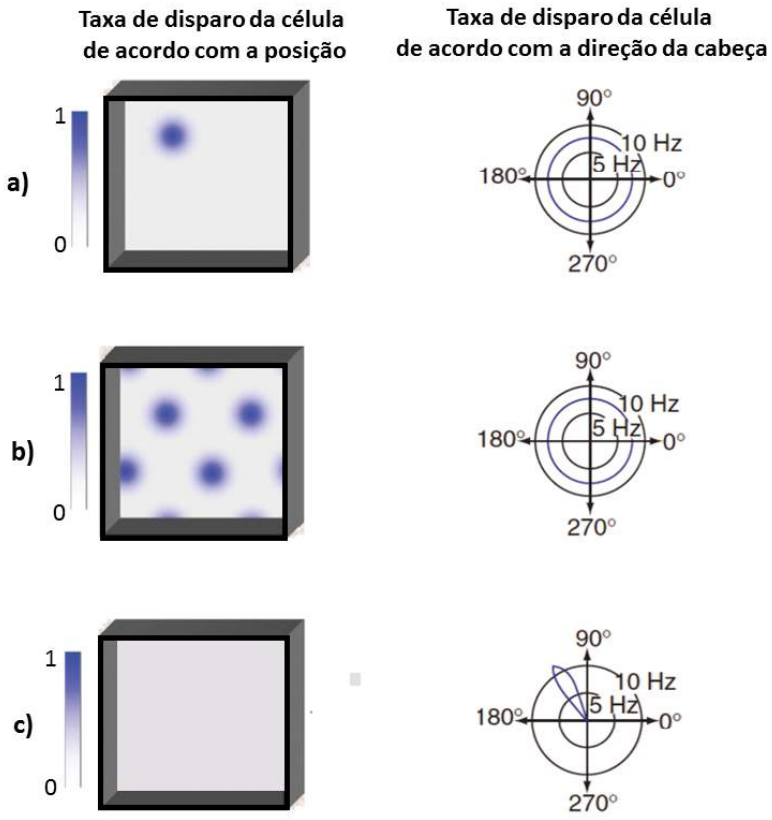

Figura 20: Células idealizadas encontradas no cérebro dos mamíferos. **a)** *Place Cell* **b)** *Grid Cell* **c)** *Head Direction Cell*. Imagem modificada de (WYETH; MILFORD, 2009)

4.1.1.2 Integração de Caminho

Os modelos atuais dos campos de disparo das *Grid Cells* sugerem que essas células integram sinais de velocidade e direção oriundos de outras partes do cérebro, embora ainda não esteja clara a origem desses sinais proprioceptivos (MOSER; KROPFF; MOSER, 2008). Nesse contexto, as informações visuais capturadas no ambiente são utilizadas para calibração inicial e correção do erro intrínseco acumulado, associado à integração de velocidade. (MOSER; KROPFF; MOSER, 2008). Sugestões de como os ratos integram a velocidade angular surgiram com o descobrimento das *Head Direction Cells*.

Após o descobrimento de mecanismos de integração de caminho existentes no cérebro dos ratos, tornou-se claro que os mesmos não apenas conseguem calcular um vetor direcional para um determinado ponto de uma arena, mas sim manter um tipo de mapa representativo do ambiente, utilizando pistas visuais e informações de movimentação própria (MCNAUGHTON et al., 2006).

Mesmo com essas descobrertas, ainda não há uma certeza acerca dos neurônios responsáveis pela integração de caminho, mas evidências sugerem que as *Grid Cells* são

responsáveis por essa tarefa. Isso decorre da característica de manutenção dos campos de disparo quando ocorre a transição para um novo ambinete, o que não é verdade para as *Place Cells* (MOSER; KROPFF; MOSER, 2008). Além disso, a origem dos sinais de movimentação própria e os mecanismos para integrar esses sinais com os landmarks do ambinete não foram determinados até o momento (MOSER; KROPFF; MOSER, 2008).

4.1.2 Arquitetura do Sistema

Inspirado pelas descobertas da *Place Cells*, *Head Direction Cells* e *Grid Cells*, Milford e Wyeth (2008) desenvolveram um método para solucionar o problema de SLAM a partir de uma nova abordagem. Como objetivo estava a criação de um algoritmo robusto e aplicável a missões de longa duração, nas quais o robô necessita manter uma representação consistente do ambiente durante toda a navegação. Partindo de uma abordagem bioinspirada, desenvolveram um método capaz de mapear ambientes ambíguos, criando uma representação semi-métrica do ambiente.

Devido sua inspiração no cérebro dos ratos, o método foi chamado de RatSLAM. A arquitetura do sistema é dividida em três módulos: *Local View Cells*, *Pose Cells* e *Experience Map*, mostrados na Figura 21.

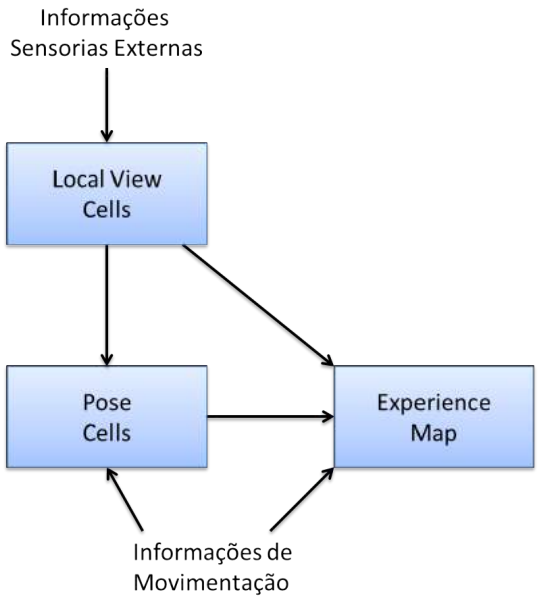

Figura 21: Arquitetura do Sistema RatSLAM

O módulo *Local View Cells* é responsável pela captura de informações visuais do ambiente e subsequente processamento dos dados sensoriais, sendo formado por um conjunto de células do tipo *Local View*, em que cada uma está relacionada a uma informação visual do ambiente. Cada nova informação é comparada com as anteriores em busca de uma correspondência. Em caso de alta similaridade, a Local View Cell correspondente é

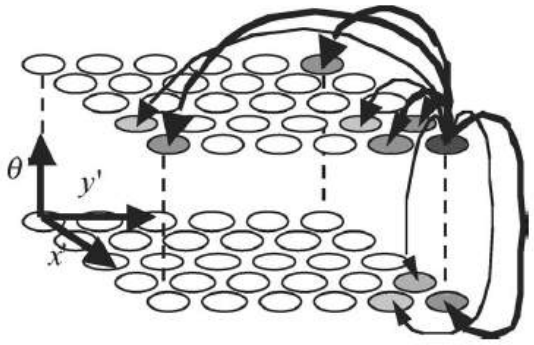

Figura 22: Rede neural de atração contínua para representação das Pose Cells

ativada. Quando não há nenhuma informação sensorial similar no mapa, um nova Local View Cell é criada.

Em cada instante de tempo , inúmeras Local View Cells podem estar ativas, sendo o grau de ativação proporcional à similaridade com a entrada. A ativação das *Local View Cells* é utilizada como entrada no módulo *Pose Cells*, na forma de conexões sinápticas. Os pesos sinápticos são calculados por aprendizado não supervisionado associativo (MILFORD; WYETH, 2008).

O módulo *Pose Cell* é o centro do sistema, formado por uma rede de neurônios para estimativa de posição e orientação do robô. Seguindo os principais trabalhos em modelagem dos neurônios espaciais (STRINGER et al., 2002b), (STRINGER et al., 2002a), Milford propôs a utilização de uma Rede Neural de Atração Contínua (CANN), como mostrado na Figura 22.

A atualização da rede é feita em etapas sequenciais, sendo descritas em detalhes a seguir.

4.1.2.1 Dinâmica da Pose Cell

Cada neurônio na rede recebe conexões sinápticas dos outros neurônios, moduladas por um peso sináptico proporcional a distância entre os neurônios na rede. Milford utilizou uma função gaussiana para cálculo dos pesos (4.1), levando em consideração que a rede está organizada de forma que os neurônios de uma borda da rede estão ligados aos neurônios da borda oposta (ver Figura 22.

$$\varepsilon = e^{-\frac{d^2_{x'}+d^2_{y'}}{2k_p}} e^{-\frac{d^2_{\theta'}}{2k_d}} \tag{4.1}$$

Considerando $r_{x'y'\theta'}$ a taxa de ativação de um neurônio genérico, cada neurônio recebe excitação lateral dos neurônios mais próximos a ele na rede, de acordo com

Equação 4.2.

$$r_{x'y'\theta'} = \sum_{a=1}^{n_{x'}} \sum_{b=1}^{n_{y'}} \sum_{c=1}^{n_{\theta'}} \varepsilon . r_{abc} \tag{4.2}$$

Com a distância entre os neurônios calculada por:

$$\begin{aligned} d_{x'} &= \min(|x' - a|, n_{x'} - |x' - a|) \\ d_{y'} &= \min(|y' - b|, n_{y'} - |y' - b|) \\ d_{\theta'} &= \min(|\theta' - c|, n_{\theta'} - |\theta' - c|) \end{aligned} \tag{4.3}$$

Após a exitação, cada neurônio da rede recebe uma inibição lateral e outra global, representada em 4.4. Os pesos de inibição local são iguais aos de excitação, porém com sinal negativo. Essa sequencia é realizada para simular uma excitação na forma de um chapéu mexicano, amplamente utilizado em redes neurais recorrentes. (KOHONEN, 1990).

$$r_{x'y'\theta'} = \sum_{a=1}^{n_{x'}} \sum_{b=1}^{n_{y'}} \sum_{c=1}^{n_{\theta'}} \omega . r_{abc} - \phi \tag{4.4}$$

Logo em seguida é realizada a entrada de movimentação na rede, para levar em consideração o deslocamento realizado pelo robo entre as duas atualizaçõess. A integração de caminho não possui um mecanismo para aumento da incerteza, como utilizado em algoritmos probabilisticos. No RatSLAM, a atividade de cada neurônio é deslocada na rede no sentido de movimentação do robõ, isto é, faz-se uma cópia da atividade atual da rede e desloca-se toda a atividade na direcão do movimento do veículo.

Por fim, ativa-se as sinapses oriundas das *Local View Cells* ativas ((4.6). A ligação entre as Local View Cells e as Pose Cells é aprendida por associação. Quando uma Local View Cell está ativa, reforça-se a ligação sináptica entre ela e todas as Pose Cells ativas, proporcional a taxa de ativação de cada uma, como apresentado em 4.5.

$$\beta_{ix'y'\theta'} = max(\beta_{ix'y'\theta'}, \lambda l_i r_{x'y'\theta'}), \tag{4.5}$$

$$\Delta r_{x',y',\theta'} = \beta_{ix'y'\theta'} . r_{x',y',\theta'} \tag{4.6}$$

Por fim, a atividade da rede neural é normalizada, de modo que a atividade total da rede seja unitária. Esse processo permite tratar a rede como uma distribuição de probabilidades sobre todo o espaço mapeado, onde cada neurônio possui a probabilidade de o robô estar na posição espacial mapeada pela célula.

Após a atualização, a Pose Cell possui uma estimativa de posição do robô, a qual é passada ao *Experience Map*. Este recebe entrada da *Pose Cell* e das *Local View Cells* ativas, de modo a mapear todas as "experiências"vivencidadas pelo robô. Cada experiência é representada por uma tupla e_i (ver equação (4.7)), formada da união das *Pose Cells* com as *Local View Cells* ativas, localizada em uma posição do espaço p^i.

$$e_i = \{R^i, L^i, p^i\} \tag{4.7}$$

O mapa é criado na forma de um grafo, unindo as experiências na ordem em que são criadas. Cada nova experiência é comparada com todas as anteriores, de acordo com uma métrica de similaridade. A métrica compara as *Pose Cells* ativas das duas experiências e as Local View Cells ativas, de acordo com:

$$S = \alpha_p |R^i - R| + \alpha_l |L^i - L| \tag{4.8}$$

As variáveis α_p e α_l definem a contribuição de cada código na comparação. Quando a métrica S for menor que um limiar S_{max}, uma nova experiência é adicionada no mapa, juntamente com a transição relativa ao deslocamento do robô, calculado por odometria:

$$t_{ij} = \{\Delta p^{ij}\} \tag{4.9}$$

A nova experiência é então criada na posição indicada pela odometria:

$$e_j = \{R^j, L^j, p^i + \Delta p^{ij}\} \tag{4.10}$$

O evento de fechamento de loop ocorre quando a nova experiência é similar a uma existente no mapa, de acordo com a métrica (4.8). O fechamento de loop é importante para a redução do erro de odometria acumulado no sistema. Quando duas experiências são ditas iguais, o erro de posição acumulado é distribuído no sistema, de acordo com a seguinte equação:

$$\Delta p^i = \alpha \left[\sum_{j=1}^{N_f} (p^j - p^i - \Delta p^{ij}) + \sum_{k=1}^{N_t} (p^k - p^i - \Delta p^{ki}) \right] \tag{4.11}$$

onde α é uma constante de correção, N_f o número de ligações de saída da experiência e_i e N_t o número de ligações de entrada na experiência e_i.

4.1.3 Aplicações

O método RatSLAM foi desenvolvido inicialmente para pequenos ambientes (MILFORD; WYETH; PRASSER, 2004), sendo posteriormente extendido para ambientes

externos (PRASSER; MILFORD; WYETH, 2006). Alguns anos depois foi utilizado para o mapeamento de um bairro inteiro, compreendendo 66Km de extensão (MILFORD; WYETH, 2008). O único sensor utilizado foi uma câmera, montada no topo de um carro, capturando imagens a 10 Hz.

Em 2009, demonstrou-se a possibilidade do algoritmo manter um mapa atualizado por um longo período de tempo (WYETH; MILFORD, 2009). Foi utilizado um robô Pioneer, equipado com uma camera omnidirecional em um escritório. O robô era responsável pela entrega de equipamentos entre as salas do escritório por um período de 2 semanas.

4.2 FAB-MAP

Proposto por Cummins e Newman (2008), o FAB-MAP (Fast Appearance-based Mapping) é um algoritmo para reconhecimento de locais baseado em aparência. Baseado em princípios probabilísticos, estima se duas imagens foram capturadas do mesmo local no ambiente. Ainda, caso não encontre correspondências entre as imagens previamente armazenadas, o algoritmo indica se a imagem pertence a um local ainda não explorado.

A abordagem básica do FAB-MAP foi inspirada nos sistemas para reconhecimento de imagens usando bag-of-words (SIVIC; ZISSERMAN, 2003; NISTER; STEWENIUS, 2006). Essa abordagem foi extendida com o aprendizado de um modelo gerador para os dados do vocabulário, levando em conta que algumas palavras(words) de aparência tendem a ocorrer simultaneamente no ambiente, pois pertencem ao mesmo objeto. (CUMMINS; NEWMAN, 2008).

4.2.1 Representação em Bag-of-Words

A representação de uma imagem como um conjunto de palavras é inspirada em algoritmos de recuperação de textos em grandes bancos de dados. Sivic e Zisserman (2003) propos a utilização do algoritmo para recuperação de frames específicos em uma sequência de vídeos, obtendo uma alta taxa de acertos com baixo custo computacional.

A primeira etapa do algoritmo é responsável pela construção de um vocabulário de palavras visuais (features). Para isso, utiliza-se um conjunto de imagens de treinamento, passando por uma etapa de extração automática de descritores com um algoritmo de detecção de features (O algoritmo SIFT (LOWE, 1999) é um exemplo de algoritmo a ser empregado nessa etapa). Todo o conjunto de features passa por um processo de clusterização, em que ocorre a formação de grupos de palavras similares. O número de grupos pode ser escolhido previamente (ARTHUR; VASSILVITSKII, 2007) ou encontrado de forma incremental (TEYNOR; BURKHARDT, 2007).

Após o treinamento do vocabulário torna-se possível a representação da imagem

por um descritor de bag-of-words. Essa representação inicia pela extração das features da nova imagem, utilizando o mesmo algoritmo empregado na etapa de treinamento. Cada feature encontrada é relacionada com uma palavra do dicionário. Dessa forma, cria-se um vetor com as ocorrências de cada palavra na imagem. Esse vetor é então normalizado e representa a frequência de aparição das palavras na imagem.

O algoritmo FAB-MAP utiliza esse algoritmo com uma pequena modificação na representação da imagem. Cada elemento do vetor é uma variável binária, indicando a existência ou não da palavra.

Dessa forma, em um instante de tempo k, cada entrada sensorial é processada com base no vocabulário de tamanho $|v|$. A imagem será representada pelo vetor binário $Z_k = \{z_1, z_2, ..., z_{|v|}\}$, com z_i indicando a presença ou ausência da i-ésima palavra do vocabulário.

4.2.2 Algoritmo FAB-MAP

O algoritmo FAB-MAP considera o mundo como um conjunto de locais discretos, onde cada local é descrito por uma distribuição no espaço de aparências. Como citado anteriormente, os dados sensoriais são convertidos para um representação do bag-of-words. Para cada novo local, pode-se calcular o grau de similaridade entre a nova percepção e a distribuição de cada local existente no mapa.

A cada instante k, o mapa do ambiente é uma coleção de locais n_k discretos e independentes: $\mathcal{L}^k = \{L_1, L_2, ..., L_{n_k}\}$

O algoritmo utiliza Chow Liu Trees para aproximar as distribuições discretas que representam os locais. Chow Liu Trees é uma estrutura que aproxima a distribuição de probabilidades sobre as palavras do dicionário, de forma que cada variável depende de no máximo uma outra variável. Essa estrutura foi escolhida por ser capaz de aproximar a distribuição de probabilidades com um custo computacional tratável, obtendo resultados melhores que a aproximação de Bayes. De forma simples, essa árvore captura a informação mútua existente no vocabulário, isto é, quais palavras tendem a aparecer(e desaparecer) ao mesmo tempo.

O aprendizado da distribuição é realizado com um conjunto de treinamento, realizado offline e uma única vez. Com a árvore construída, o FAB-MAP estima se duas aparências foram capturadas no mesmo local ou pertencem a locais distintos. Desde sua criação já foram feitas diversas extensões do algoritmo, visando principalmente sua aplicação para criação de mapas em grande escala (CUMMINS; NEWMAN, 2010) (CUMMINS; NEWMAN, 2011).

4.2.3 Aplicações

A primeira versão do FAB-MAP foi validada em dois datasets urbanos, utilizando uma camera com campo de visão perpendicular a trajetória. Ambos datasets compreendiam aproximadamente 2Km cada, onde o sistema foi capaz de detectar fechamentos de loop corretos, sem falsos positivos. (CUMMINS; NEWMAN, 2008)

Posteriormente o sistema foi extendido para aumentar sua performance, onde ocorreu sua validação em um ambiente com 1000Km de extensão. O sistema foi executado em tempo real, tornando o algoritmo FAB-MAP estado da arte em SLAM baseado em aparência. (CUMMINS; NEWMAN, 2011)

(MADDERN et al., 2009) analisaram a integração do algoritmo FAB-MAP com o RatSLAM, onde o primeiro seria responsável pela detecção de loops para o último. Os testes foram realizados sobre o dataset do subúrbio de St. Lucia, na Austrália, mesmo ambiente utilizado para validação do RatSLAM. Nesse ambiente, devido a alta quantidade de loop entrelaçados, a integração dos dois sistemas obteve resultados inferiores com relação aos testes apenas com o algoritmo RatSLAM. No entanto, os autores afirmam que a integração deve obter resultados melhores em datasets com alta variação de iluminação, ou seja, navegações por longos períodos de tempo, pois o algoritmo de scanline utilizado pelo RatSLAM é sensível a variações de iluminação.

Glover et al. (2010) realizou experimentos com a integração dos dois algoritmos, em cenários de longa duração, capturados em diferentes períodos do dia. Os resultados demonstraram uma melhor performance do sistema, aumentando a precisão e o recall quando comparado aos dois sistemas independentes.

No próximo capítulo será apresentada uma proposta de integração do algoritmo FAB-MAP com uma extesão do algoritmo RatSLAM, modificado para o mapeamento de ambientes subaquáticos.

5 DolphinSLAM: Sistema Bioinspirado para Localização e Mapeamento de Robôs Subaquáticos

Frente aos desafios encontrados no ambiente subaquático, um sistema para localização e mapeamento é imprescindível para a atuação de um robô móvel. Aqui será apresentada uma proposta de sistema para resolver o problema de SLAM nesses ambientes, baseada em uma abordagem diferenciada, inspirada no cérebro dos mamíferos.

Partindo do desenvolvimento do método RatSLAM, o qual resolve o problema de SLAM em ambientes terrestres de forma robusta e eficiente, o presente trabalho estende o algoritmo para operação em ambientes subaquáticos. Para contornar as os efeitos da turbidez, monotonicidade e não estruturação do ambiente subaquático, utiliza-se o algoritmo FAB-MAP para detecção de locais, estimando probabilisticamente se duas leituras sensoriais foram capturadas no mesmo local.

Levando em consideração sua inspiração no cérebro dos mamíferos, aliado ao foco em ambientes subaquáticos, o método aqui proposto será chamado DolphinSLAM. No entanto, até o presente momento não foi possível provar a existência dos neurônios espaciais (ver subseção 4.1.1) nos golfinhos.

As principais contribuições do DolphinSLAM são listadas a seguir:

- Desenvolvimento de um algoritmo bioinspirado para mapeamento de ambientes subaquáticos com custo computacional linear ao tamanho dos *landmarks* armazenados, possível de ser utilizado com diferentes sensores baseados em imagens, como câmeras e sonares.
- Integração do algoritmo FAB-MAP à abordagem inspirada. Com isso, o sistema apresentará robustez a mudanças na iluminação, devido ao efeito da turbidez na água, similar ao que ocorre em missões terrestres de longa duração. Além disso, devido à inexistência de um conhecimento *a priori* do que será encontrado no fundo do oceano, a aplicação de um método probabilístico de reconhecimento de locais baseado em um modelo generativo do vocabulário de palavras tende a aumentar a confiabilidade do sistema de percepção.
- Desenvolvimento de um algoritmo de código aberto para posterior utilização em pesquisas futuras.

Como citado anteriormente, o algoritmo estende o RatSLAM, modificando grande parte de sua estrutura, a fim de criar um método aplicável ao ambiente subaquático. O módulo de percepção (*Local View Cells*) foi totalmente reescrito, integrando o FAB-MAP para o reconhecimento de locais previamente visitados, o que modifica o nível de excitação dos neurônios do tipo *Local View*. O algoritmo é capaz de tratar informações de câmeras e sonares, o que não era possível anteriormente.

A rede neural, centro do algoritmo, teve sua dinâmica modificada. A excitação passou a ser com uma função do tipo Chapéu Mexicano, não sendo mais necessária uma inibição lateral após a excitação. Essa modificação torna o algoritmo mais eficiente.

Os neurônios da rede (Pose Cells) passaram a mapear o espaço tridimensional. Dessa forma, os neurônios apresentam um comportamento similar as *Place Cells* 3D encontradas nos morcegos (YARTSEV; ULANOVSKY, 2013). O mecanismo de integração de caminho forma alterados de acordo com essa modificação para o espaço tridimensional. Ainda, a normalização da rede foi modificada de modo a modelar uma ativação em cada neurônio no intervalo de zero a um, como proposto em (STRINGER et al., 2002a).

O mapa de experiências foi totalmente remodelado de acordo com as peculiaridades inerentes a um ambiente tridimensional, bem como o mecanismo de correção do erro acumulado.

O sistema DolphinSLAM está estruturado de forma modular, como apresentado na Figura 23. Cada módulo será explicado em detalhes nas próximas seções.

5.1 Sistema de coordenadas

Com o objetivo de padronização, antes de entrar na explicação do método é necessário definir-se o sistema de coordenadas adotado. O mesmo pode ser visualizado na Figura 24. O sistema de coordenadas global segue a seguinte convenção:

- O eixo x acompanha o sentido do norte magnético da terra, quando esta informação está disponível. Caso contrário, é fixado na primeira posição do robô, em concordância com seu sistema de coordenadas.
- O eixo y é definido com seu eixo crescente na direção leste. Da mesma forma, na ausência de leitura magnética da terra, o mesmo é fixado em cima do eixo y do robô, em sua primeira inicial.
- O eixo z, em concordância com a regra da mão direita, possui sua direção crescente apontado para baixo, em direção ao fundo do mar.

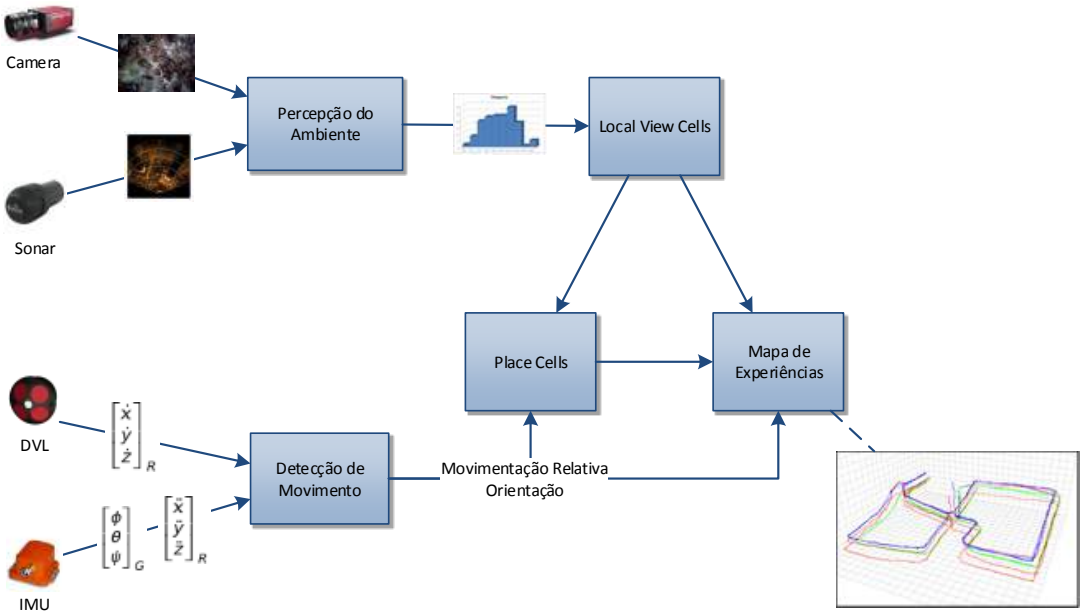

Figura 23: Arquitetura do DolphinSLAM. O módulo de percepção do ambiente é responsável pelo processamento dos dados oriundos dos sensores exteroceptivos. O módulo detecção de movimento é responsável pela integração do sinal de velocidade e ou aceleração, relacionando a orientação do veículo a fim de estimar o deslocamento em um sistema de referencia global. O módulo Local View Cells apresenta uma estimativa de local por onde o robô navega, baseado nas imagens capturadas do ambiente. O módulo Place Cell é responsável pela estimativa de posição do robô, baseada no deslocamento do mesmo e na informação oriunda do módulo Local View Cells. O módulo Mapa de Experiências é responsável pela criação e manutenção de uma mapa representativo do ambiente e estimativa da posição do robô em um sistema de coordenadas global.

As rotações em torno de cada eixo coordenado, representadas pelos ângulos de Euler, são definidas por:

- ϕ Rotação em torno do eixo x, chamada de *roll*
- θ Rotação em torno do eixo y, chamada de *pitch*
- ψ Rotação em torno do eixo z, chamada de *yaw*

O sistema de coordenadas do robô é, convenientemente, definido:

- O eixo x será definido no sentido longitudinal do robô, apontado para a frente do mesmo.

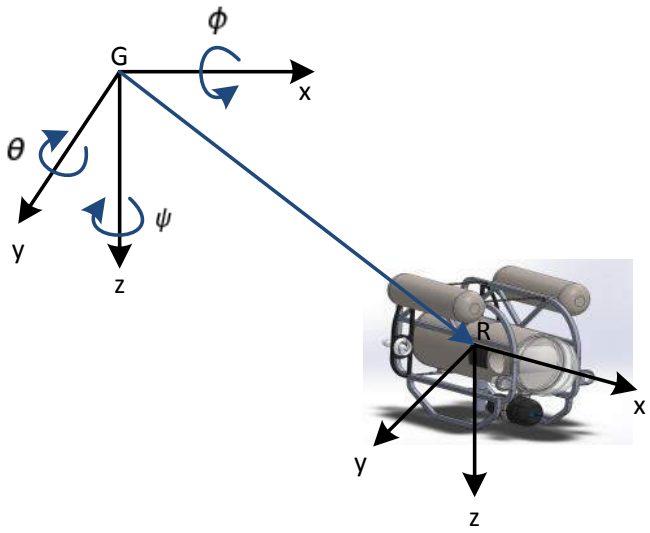

Figura 24: Sistema de Coordenadas.

- O eixo y será definido perpendicularmente ao eixo x, apontado para *stardboard*, ou seja, para a direita do corpo do robô.
- O eixo z, seguindo a regra da mão direita, será apontado para baixo do robô, perpendicular aos eixos x e y.

5.2 Percepção do Ambiente

Os dados sensoriais, oriundos dos sensores óptico e acústico, serão processados no módulo de percepção do ambiente. A proposta atual utiliza como sensores *exteroceptivos* uma câmera monocular e um sonar de imagem do tipo *forward looking*. O sensor escolhido para a missão depende das características do meio e do tipo de exploração. A câmera é indicada para missões em águas com baixo nível de turbidez e navegação próxima aos objetos, enquanto o sonar é indicado para missões com campo de visão amplo e/ou águas com alta turbidez.

5.2.1 Registro das Imagens ópticas

A captura de imagens óticas no ambiente subaquático é sensível à turbidez da água. No entanto, a aplicação desses sensores é necessária em tarefas de inspeção, o que justifica sua utilização no método proposto. Como exemplo, pode-se citar um campo de

extração de petróleo *offshore*, composto por dutos de transmissão de alguns quilômetros de extensão, os quais necessitam inspeção visual periódica.

O primeiro passo do processamento é a extração de pontos característicos das imagens. Baseado em estudos de robustez dos algoritmos de detecção de *features* pontuais em imagens subaquáticas (GARCIA; GRACIAS, 2011), foi utilizado o detector Hessian para detecção de features. Para descrição, utilizou-se o algoritmo SURF (BAY; TUYTELAARS; GOOL, 2006), com um descritor de 128 posições.

5.2.2 Registro das Imagens acústicas

Diferente das imagens ópticas, as quais são ricas em textura, as imagens acústicas são simples, sendo inviável a aplicação de detectores de *features* pontuais. Além disso, os algoritmos comuns utilizados para descrever *landmarks* pontuais são instáveis nas imagens de sonar, não podendo ser utilizados.

Por outro lado, as imagens de sonar são ricas em formas geométricas. Por esse motivo, optou-se pela utilização de *features* baseadas no momento de formas geométricas (HU, 1962).

Com o objetivo de melhorar a qualidade das imagens, reduzindo o ruído inerente às imagens acústicas, aplica-se um filtro Gaussiano, seguido de um filtro mediano sobre as imagens brutas. Logo após, as imagens são binarizadas, para facilitar a detecção de contornos. Por fim, os contornos são obtidos nas regiões de alta mudança de intensidade.

A descrição das formas geométricas foi realizada pelo cálculo dos sete momentos de Hu (HU, 1962), os quais são invariantes a translação, rotação e escala.

5.2.3 Representação em Bag of Words

Independente do sensor utilizado e do algoritmo empregado na detecção de características da imagem, o método DolphinSLAM utiliza o algoritmo Bag of Words para representação das imagens. Essa decisão advém da utilização do método FAB-MAP, o qual utiliza uma representação em Bag of Words como entrada do algoritmo.

A representação utilizada cria um histograma contendo as frequências de palavras existentes na imagem. Para isso, é necessário a criação do vocabulário de palavras. O processo de criação do vocabulário é apresentado na Figura 25. Utiliza-se um conjunto de imagens de treinamento, as quais são descritas de acordo com o sensor utilizado. Os descritores passam por um processo de clusterização, onde os elementos representativos de cada grupo serão as palavras do dicionário. Utilizou-se o algoritmo KMeans para clusterização (ARTHUR; VASSILVITSKII, 2007).

Cada imagem é então descrita como um histograma com as frequências das palavras

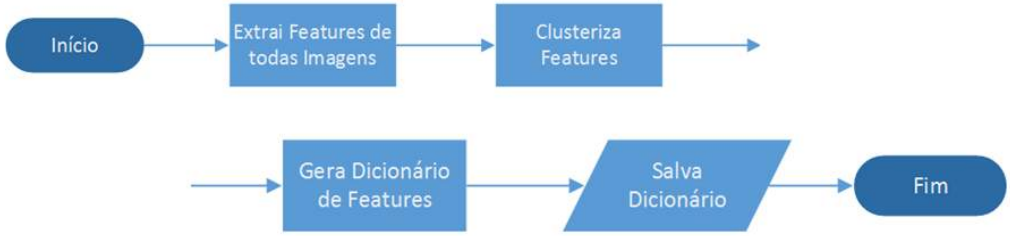

Figura 25: Etapas de treinamento do algoritmo Bag of Words

no dicionário. A Figura 26 apresenta exemplos de imagens representadas com o algoritmo Bag of Words.

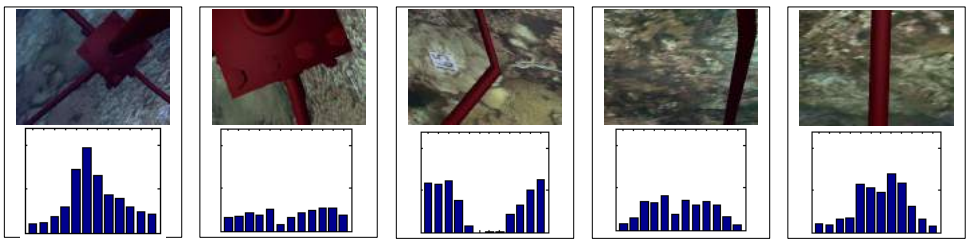

Figura 26: Exemplos de histogramas do Bag of Features

5.3 Detecção de Movimento

O tratamento dos dados oriundos de sensores *proprioceptivos* é igualmente importante para qualquer sistema de navegação robótico. O módulo de detecção de movimento é responsável pelo processamento sensorial relacionado à movimentação do robô no ambiente.

O ambiente subaquático requer a utilização de sensores diferenciados para a estimativa de movimentação, diferentes dos sensores empregados na robótica móvel. Por exemplo, a contagem de rotação dos motores não pode ser utilizada para estimar a velocidade do robô, pois o mesmo está sujeito a um alto arraste inerente ao meio físico no qual opera.

Como apresentado no Capítulo 3, o principal sensor utilizado para medição de velocidade é a DVL, a qual estima a movimentação do robô pelo envio de ondas sonoras na direção do fundo do mar. No entanto, para integração do sinal de velocidade, faz-se necessário uma leitura da orientação atual do robô, pois a DVL retorna as informações de velocidade no sistema de coordenadas fixo ao robô (ver seção 5.1.

Para a leitura da orientação do robô pode-se empregar magnetômetros, giroscópios ou uma unidade de medição inercial (IMU). Independente do sensor utilizado, a transformação entre os sistemas de coordenadas utilizada a orientação em ângulos de Euler (ϕ, θ

e ψ):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}^G = R(\phi, \theta, \psi) \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}^R , \tag{5.1}$$

onde R é a matriz de rotação para transformar a velocidade do frame do robô para o frame global.

O deslocamento do robô deve ser acumulado de acordo com o tempo de atualização do sistema, que normalmente é maior que a taxa de amostragem do sensor de movimentação.

A captura de imagens é utilizada como gatilho para atualização do sistema DolphinSLAM. Podem ser utilizados todos os frames capturados ou apenas alguns *keyframes* . Com isso, o deslocamento deve ser acumulado entre o tempo de dois *keyframes* capturados.

Sendo Δ_t o tempo entre duas leituras do sensor DVL, o deslocamento instantâneo é calculado por:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{t+1}^G = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_t^G + \Delta_t \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_t^G \tag{5.2}$$

O deslocamento acumulado é reinicializado após o envio da informação de movimentação para os módulos *Place Cells* e Mapa de Experiências. O *dead reckoning*, para fins de validação do método, será calculado pela soma de todos os deslocamentos relativos, desde o início da navegação.

5.4 Local View Cells

Local View Cells são neurônios que representam a impressão visual de determinado local. Ou seja, são células especiais utilizadas para codificar a informação sensorial capturada no ambiente. À medida que o robô desloca-se no ambiente, adquirindo novas impressões visuais, ocorre a criação dinâmica de novas células do tipo *Local View*.

O método DolphinSLAM representa as informações sensoriais por um histograma de palavras do Bag of Words. Esses histogramas serão avaliados pelo algoritmo FAB-MAP a fim de estimar a probabilidade de terem sido capturados no mesmo local. Há uma etapa de treinamento do FAB-MAP, responsável pelo aprendizado do modelo de geração das palavras existentes no vocabulário.

A Figura 27 apresenta um conjunto de *Local View Cells*, responsáveis pela codificação do histograma da imagem capturada. O conjunto de todas as *Local View Cells* é

representado por L e armazena o nível de ativação do neurônio l_i:

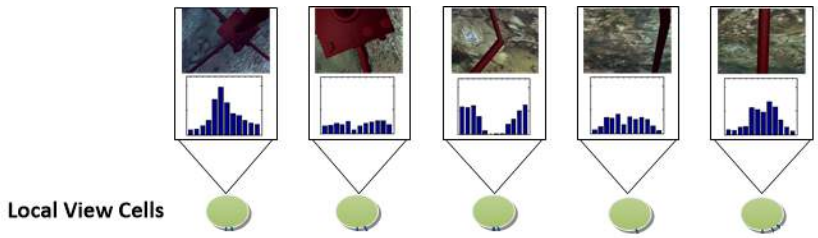

Figura 27: Diagrama para visualização das Local View Cells. Cada Local View Cell está relacionada com um histograma do Bag of Features.

$$L = \{l_1, l_2, l_3, ..., l_i, ..., l_m\}, \tag{5.3}$$

Onde m é o número de *Local View Cells* existentes no sistema.

5.4.1 Treinamento da árvore de probabilidades Chow Liu

O algoritmo FAB-MAP possui uma etapa de treinamento para aprendizado da árvore de probabilidades Chow Liu, como explicado na seção 4.2. Utiliza-se o mesmo conjunto de imagens de treinamento empregado na geração do vocabulário do Bag of Words. Cada imagem é descrita como um histograma do BoW. O conjunto de todos os histogramas é utilizado para treinamento, onde são encontrados os padrões de co-ocorrência de palavras, codificado na árvore de probabilidades.

5.4.2 Criação de Local View Cells

O algoritmo FAB-MAP estima a probabilidade de um imagem ter sido adquirida em um local previamente armazenado no mapa(p_i), ou pertencer a um novo local (p_{new}). Quando a probabilidade de pertencer a um novo local superar as probabilidades, uma nova *Local View Cell* é criada, como apresentado na Figura 28.

A taxa de ativação l_{new} da nova local view cell será setada como 1.

$$l_{new} = 1.0$$

5.4.3 Ativação de Local View Cells

Em casos de reconhecimento de local, em que as probabilidades calculadas para determinados locais pelo FAB-MAP são maiores do que para um novo local, ocorre a ativação de neurônios existentes no sistema. A cada instante de tempo, há a possibilidade

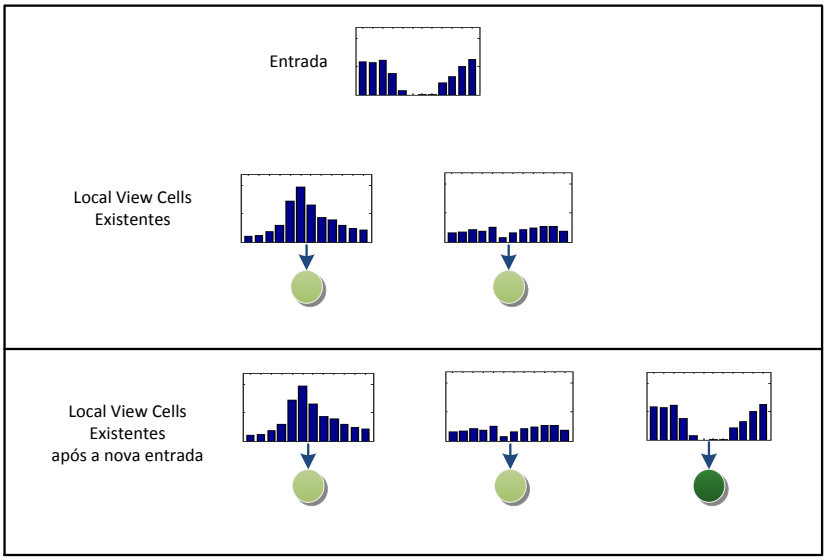

Figura 28: Criação de novas Local View Cells a partir do histograma de entrada

de mais de um local possuir alta probabilidade. Uma Local View Cell é dita ativa quando sua probabilidade, calculada pelo FAB-MAP, é maior que um limiar P_s.

A Figura 29 exemplifica a situação onde um local existente no mapa é reconhecido, ativando-se a *Local View Cell* correspondente.

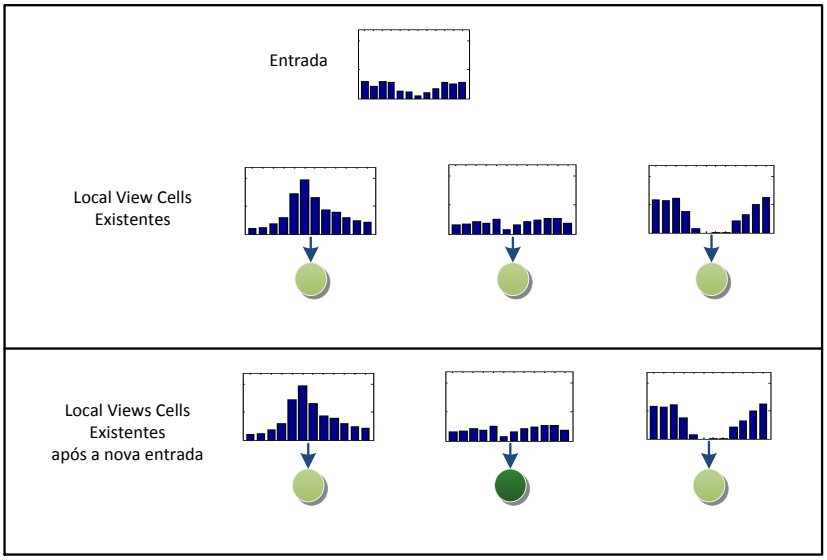

Figura 29: Ativação das Local View Cells a partir do histograma de entrada

A taxa de ativação das *Local View Cells* é dada por:

$$l_i = \begin{cases} p_i & \text{se} \quad p_i \geq P_s \\ 0 & \text{caso contrario} \end{cases} \tag{5.4}$$

O conjunto de *Local View Cells* ativas em determinado instante será responsável pela entrada externa nas *Place Cells*, na forma de excitações sinápticas. As conexões sinápticas serão calculadas na subseção 5.5.3

5.5 Place Cells

O módulo principal do sistema DolphinSLAM é composto por um conjunto de células neuronais do tipo *Place Cells*. De acordo com estudos da neurosciência (ver subseção 4.1.1), *Place Cells* são células que aumentam sua taxa de disparo quando o mamífero aproxima-se de determinada região em um ambiente. Yartsev e Ulanovsky (2013) comprovou a existência de *Place Cells* tridimensionais no cérebro de morcegos. Embora não seja possível, por questões éticas, realizar experimentos com golfinhos, em virtude sua semelhança de classe biológica, aliado a similaridade do ambiente em que navegam (tridimensional), presumiu-se sua existência nesses animais. Dessa forma, optou-se por utilizar um modelo computacional de neurônios do tipo *Place Cell 3D*.

Para modelagem dos neurônios, utilizou-se o modelo mais aceito entre os neurocientistas, uma Rede Neural de Atração Contínua (CANN) (GIOCOMO; MOSER; MOSER, 2011) (STRINGER et al., 2002a). Redes CANN são redes neurais artificiais utilizadas para a modelagem de mecanismos neuronais com capacidade de memorização. Cada neurônio recebe ligações sinápticas externas à rede e ligações recorrentes, oriundas dos outros neurônios da rede. Sua dinâmica interna caracteriza-se pela criação de atratores, isto é, estados internos em que a atividade neuronal converge para um grupo de neurônios específicos (atratores). A região da rede com alta atividade, após a convergência para um atrator, é chamada de pacote de ativação.

A representação gráfica da rede é apresentada na Figura 30. Cada círculo na imagem representa um neurônio do tipo *Place Cell* e a intensidade do círculo representa a taxa de ativação do neurônio. Os neurônios estão organizados em forma de uma caixa, em que cada eixo da caixa representa um eixo cartesiano de mapeamento, demonstrado pelo sistema de coordenadas $(x'y'z')$ fixado na imagem.

Os neurônios em cada extremidade da rede estão ligados à extremidade oposta, de forma semelhante a quaisquer dois neurônios vizinhos na rede. Dessa forma, a capacidade espacial de mapeamento não está limitada ao número de neurônios existentes na rede. Com isso, cada neurônio será responsável por mapear regiões periódicas do ambiente. Essa

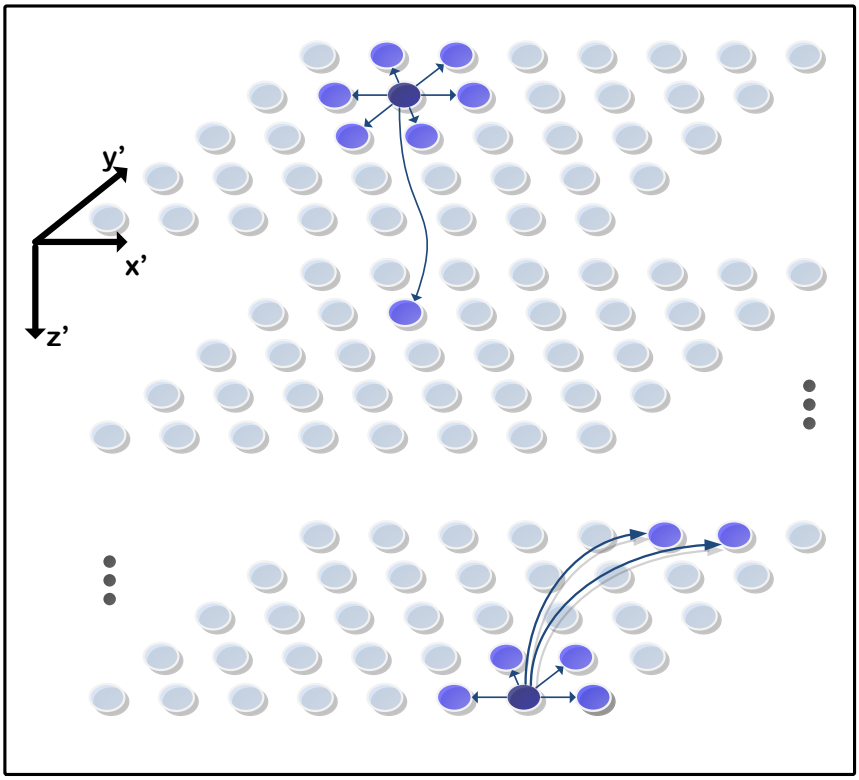

Figura 30: Rede Neural para modelagem das Place Cells 3D. A rede está organizada em uma topologia tridimensional, onde cada neurônio está ligado a seus vizinhos por conexões sinápticas. Os neurônios em uma extremidade estão ligados à extremidade oposta, a fim de resolver problemas de contorno e aumentar a capacidade de mapeamento da rede.

característica baseia-se em outro tipo de neurônio existente no cérebro dos mamíferos, as *Grid Cells*.

A dinâmica de atualização da rede é realizada por um conjunto de etapas sequenciais: excitação lateral, integração de caminhos, excitação externa e normalização.

5.5.1 Excitação Lateral

Neurônios próximos na rede cooperam para a criação de um pacote de ativação, enquanto neurônios afastados competem pela manutenção de seu pacote de ativação. Esse mecanismo de competição/cooperação é possível pela excitação lateral recorrente da rede. Todos os neurônios da rede possuem ligações sinápticas recorrentes, cuja sinapse é moderada pela distância entre as células na rede.

Dois neurônios próximos devem ter sinapses recorrentes positivas entre si, para que possam cooperar pela geração do pacote de ativação. No entanto, neurônios distantes devem ter sinapses negativas ou nulas, habilitando a competição entre eles na rede. O

modelo proposto utiliza uma função do tipo "chapéu mexicano" para a geração dos pesos sinápticos da rede. Dessa forma, neurônios próximos terão ligações sinápticas positivas, neurônios à média distância terão ligações sinápticas negativas e neurônios a longa distância terão ligações sinápticas nulas.

O peso sináptico ε entre dois neurônios r é assim calculado:

$$\varepsilon = \left(1 - \frac{d^2}{\sigma^2}\right) e^{-\frac{d^2}{2\sigma^2}}, \tag{5.5}$$

Onde σ é o desvio padrão da função "chapéu mexicano", representada em número de células, e d é a distância entre os dois neurônios na rede:

$$d = \sqrt{d_{x'}^2 + d_{y'}^2 + d_{z'}^2}, \tag{5.6}$$

$$\begin{aligned} d_{x'} &= \min(|x' - a|, n_{x'} - |x' - a|) \\ d_{y'} &= \min(|y' - b|, n_{y'} - |y' - b|) \\ d_{z'} &= \min(|z' - c|, n_{z'} - |z' - c|) \end{aligned} \tag{5.7}$$

Cada neurônio recebe excitação lateral de todos os neurônios da rede, de acordo com a proximidade entre eles. A quantidade de excitação que um neurônio receberá é definida por:

$$r_{x'y'z'} = \sum_{a=1}^{n_{x'}} \sum_{b=1}^{n_{y'}} \sum_{c=1}^{n_{z'}} \varepsilon . r_{abc} \tag{5.8}$$

5.5.2 Integração de Caminhos

A rede neural é responsável pela estimativa de posição do robô no ambiente. Para tal, ela deve levar em consideração a movimentação realizada pelo robô desde a última atualização da rede. Esse deslocamento é calculado pelo módulo de Detecção de Movimento, como explicado anteriormente.

De forma similar à integração de caminhos realizada pelo algoritmo RatSLAM, o método proposto não utiliza modelos biológicos nessa etapa da rede devido ao alto custo computacional dos modelos bioinspirados desenvolvidos até o momento. Com isso, foca-se em uma solução eficiente para a resolução do problema.

A integração de caminhos é realizada pelo deslocamento lateral da atividade neuronal de cada célula, de acordo com o deslocamento realizado pelo robô. A Figura 31 apresenta uma representação gráfica da solução utilizada. O robô desloca-se no plano xy uma distância equivalente ao espaço de mapeamento de cada neurônio. Logo, a atividade

neuronal deve ser deslocada uma unidade para a direita (sentido do deslocamento). O funcionamento é idêntico para a dimensão z da rede.

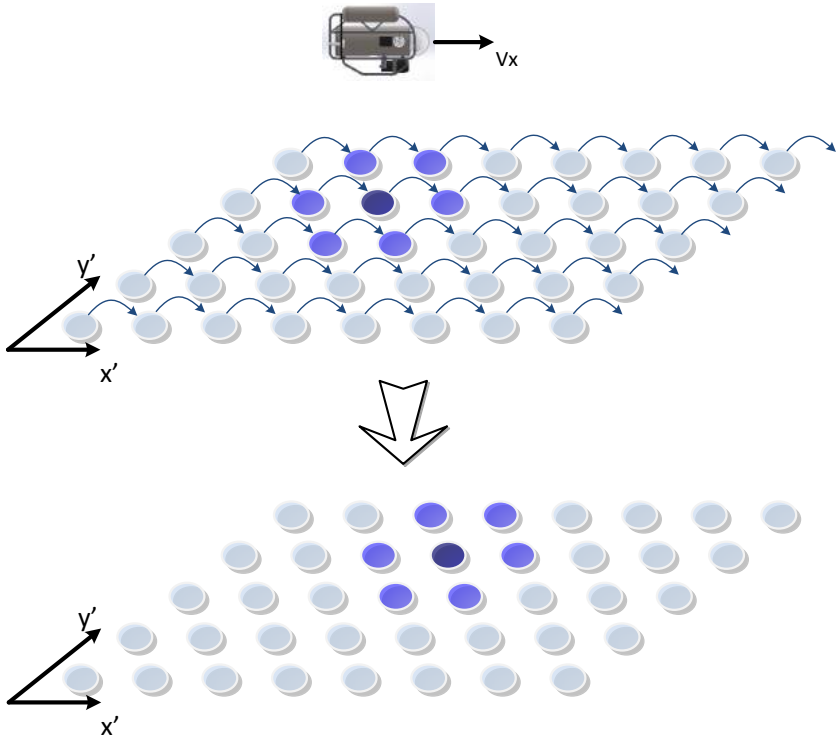

Figura 31: Modelo de Integração de caminho. O pacote de ativação é transladado no sentido do deslocamento. Não há crescimento na incerteza resultante da movimentação.

Vale ressaltar que o método empregado não prevê um aumento na incerteza da estimativa devido ao deslocamento do robô.

5.5.3 Excitação Externa

Uma característica importante das *Place Cells* biológicas é sua relação com os *landmarks* visuais encontrados no ambiente. Eles servem como mecanismo de correção do erro de localização, executando uma espécie de reinicialização sobre a estimativa de posição da *Place Cell*.

Esse importante mecanismo de correção do erro inerente ao processo de integração do caminho é colocado no modelo na forma de conexões sinápticas de entrada. Uma determinada *Place Cell* está relacionada a um conjunto de *Local View Cells*, isto é, cada *Place Cell* está ancorada a no mínimo uma *Local View Cell*. De forma análoga, uma *Local View Cell* pode estar relacionada a mais de uma *Place Cell*. Um exemplo de ligações sinápticas do tipo *Local View Cell – Place Cell* é apresentado na Figura 32.

Os pesos sinápticos são definidos por aprendizado não supervisionado associativo. Quando duas células (Uma Local View e uma Place Cell) estão ativas no mesmo instante

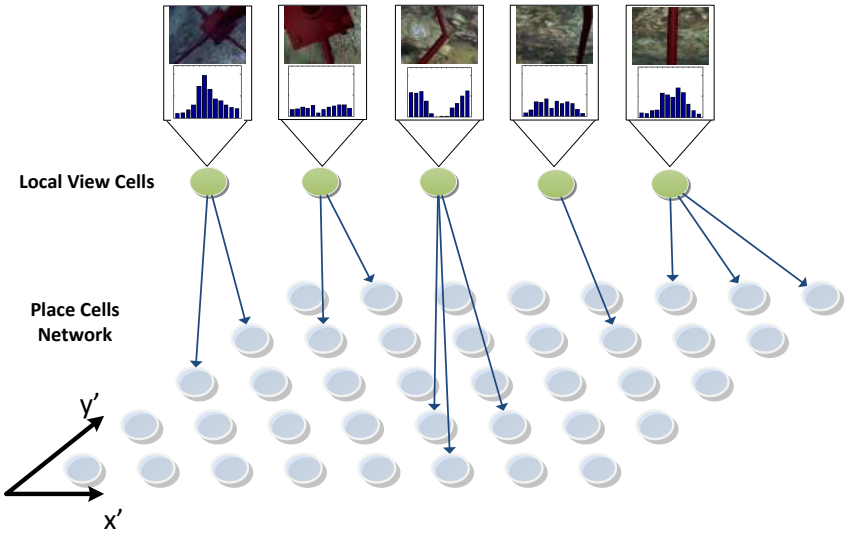

Figura 32: Diagrama para visualização das Local View Cells e suas conexões com as Place Cells.

de tempo, a ligação sináptica entre elas é reforçada. Após uma ligação ser reforçada, esta não será mais esquecida. Esse comportamento de não esquecimento deve-se à ancoragem biológica das *Place Cells* aos *landmarks* visuais.

O peso sináptico β é calculado por:

$$\beta_{ix'y'z'} = max(\beta_{ix'y'z'}, \lambda.l_i.r_{x'y'z'}), \tag{5.9}$$

sendo l_i a taxa de ativação da *Local View Cell* i, $r_{x'y'z'}$ a taxa de ativação do neurônio $x'y'z'$ e λ uma constante de aprendizado.

A quantidade de excitação externa, recebida por um neurônio *Place*, é calculada pela equação:

$$\Delta r_{x'y'z'} = \sum_{i=1}^{m} \beta_{ix'y'z'}.l_i \tag{5.10}$$

5.5.4 Normalização

A última etapa de atualização da rede é a normalização da atividade neuronal. Seu objetivo é limitar a atividade sináptica da rede, evitando um crescimento ilimitado do pacote de ativação.

Inspirada nos trabalhos de (STRINGER et al., 2002b),(STRINGER et al., 2002a), a excitação máxima de um neurônio será limitada em uma unidade. A maior atividade da rede passará a ser unitária. Os neurônios com atividade menor serão reduzidos na mesma

proporção, como proposto:

$$\eta = \max(r_{x'y'z'}) \tag{5.11}$$

$$r_{x'y'z'} = \frac{r_{x'y'z'}}{\eta} \tag{5.12}$$

5.6 Mapa de Experiências

O Mapa de Experiências é o módulo responsável pela criação de um mapa com todas as experiências "vivenciadas" pelo robô durante sua navegação. Trata-se de um mapa topológico, semi-metrico, baseado na trajetória percorrida pelo robô durante a navegação. Cada posição armazenada no mapa conterá informações sobre a atividade neuronal da *Place Cell* R e ao conjunto de *Local View Cell* L ativas.

Essas informações serão codificadas em uma experiência e_i, na forma de uma tupla:

$$e_i = \{R^i, L^i, p_i\}, \tag{5.13}$$

onde p_i é a posição espacial da experiência no mapa. A primeira experiência é colocada na posição inicial $(0, 0, 0)$. As demais são calculadas pelo deslocamento relativo do robô.

5.6.1 Criação de experiências

Novas experiências são criadas dinamicamente, de acordo com a navegação do robô no ambiente. Cada experiência armazena uma informação visual, representada pela*Local View Cell* e uma estimativa de posição, representada pela $PlaceCell$. Uma nova experiência é criada quando uma das duas informações é modificada, seja visual ou posição.

As novas informações são fornecidas como entrada ao módulo Mapa de Experiências, onde são comparadas com todas as experiências armazenadas. A métrica de similaridade S da entrada com a experiência e_i é comparada com um limiar S_{min}.

$$S^i = \alpha_R|R^i - R| + \alpha_L|L^i - L|, \tag{5.14}$$

Onde R é a entrada representativa da *Place Cell* e L é a entrada representativa da *Local View Cell*. α_R e α_L são constantes de ponderação de cada entrada na métrica de similaridade. Quando a métrica S não excede o limiar S_{min} em nenhuma experiência, cria-se uma nova experiência no mapa, com as informações de entrada. A posição da

experiência no mapa é definida a partir da experiência anterior e do deslocamento do robô Δp:

$$e_j = \{R^j, L^j, p^i + \Delta p^{ij}\} \tag{5.15}$$

A nova experiência é ligada a anterior de forma topológica, armazenando a informação de odometria.

$$t_{ij} = \{\Delta p^{ij}\ \} \tag{5.16}$$

5.6.2 Fechamento de loop

Após determinado período de navegação, espera-se que o robô revisite locais antigos. Nesse momento, ocorrerá o reconhecimento dos locais, baseado nas informações das $LocalViewCells$, as quais serão responsáveis pela mudança do pacote de ativação na rede de $PlaceCells$. Quando as informações de entrada no Mapa de Experiências forem similares, a métrica S será superior ao limiar em determinada experiência.

Esse evento, conhecido como fechamento de loop no contexto de SLAM, é responsável pela redução do erro inerente a integração de caminhos. Sendo e_k a experiência reconhecida, o mapa será atualizado da seguinte maneira:

1. Cria-se uma nova experiência e_j no mapa, com as representações das Local View Cells e Place Cells atuais, posicionada a partir da experiência anterior e do deslocamento realizado pelo robô:
$$e_j = \{R^j, L^j, p^i + \Delta p^{ij}\} \tag{5.17}$$

2. Cria-se uma ligação entre a experiência anterior e a atual:
$$t_{ij} = \{\Delta p^{ij}\} \tag{5.18}$$

3. Cria-se uma ligação da experiência atual e a experiência reconhecida, com o deslocamento baseado na informação visual, extraída pela transformação das duas imagens:
$$t_{jk} = \{\Delta p^{im}\} \tag{5.19}$$

A criação de uma nova experiência no momento do reconhecimento deve-se ao fato do robô não estar exatamente na mesma posição no momento das duas aquisições sensoriais. O ambiente subaquático não é estruturado, inexistindo vias de trânsito. Logo, as duas posições serão próximas, não iguais. Dessa forma, necessita-se o desenvolvimento de algoritmos para estimativa de posição do robô com base na imagem. Devido ao amplo

escopo do trabalho, o desenvolvimento de algoritmos para a resolução desse problema será colocado como trabalhos futuros. O erro acumulado até o fechamento de loop deve ser propagado sobre o caminho percorrido pelo robô. Para isso, calcula-se o erro de posição ϵ_p entre a experiência atual e a experiência reconhecida, descontando-se o deslocamento calculado com base na imagem Δp^{im}.

$$\epsilon = p_j + \Delta p^{im} - p_k \tag{5.20}$$

O erro é propagado por todas as experiências. Dado n o número de experiências existentes entre e_k e e_j, as ligações t são atualizadas por:

$$\Delta t = -\frac{\epsilon}{n+1} \tag{5.21}$$

Todas as experiências tem suas posições recalculadas, com base no peso das arestas (t) que as conectam.

5.7 Implementação Open Source

O método DolphinSLAM foi implementado com a utilização do Sistema Operacional Robótico (ROS) (QUIGLEY et al., 2009), amplamente utilizado pelos pesquisadores da área de robótica. Para o processamento dos dados, utilizou-se a biblioteca de visão computacional OpenCV (BRADSKI, 2000). Utilizou-se ainda, para algoritmos gerais, a biblioteca BOOST (SCHÄLING, 2011).

O código está disponível, sob licença GNU GPL v2.0 (GNU. . . , 2015), no endereço eletrônico <https://github.com/dolphin-slam/dolphin_slam>.

O próximo capítulo apresentará os experimentos realizados para validação do método em diferentes cenários e sensores.

6 Experimentos

O presente capítulo apresentará os experimentos utilizados para validação do método proposto e análise de desempenho em diferentes cenários. Foram realizados três estudos de caso. O primeiro, em ambiente semi-simulado, obetiva a correta parametrização do algoritmo proposto. O segundo ambiente, uma piscina, objetiva validar o sistema em um ambiente real. Como terceiro estudo, o método foi alimentado com imagens de sonar de uma marina.

6.1 Métricas de avaliação

A primeira métrica de avaliação será o número de Local View Cells criadas e reconhecidas. Como explicado no capítulo anterior, as Local View Cells são ativadas pelo retorno do algoritmo FAB-MAP, o qual estima a probabilidade de uma local ser conhecido ou ser novo.

Como segunda métrica, será apresentado o mapa criado, onde pode obter-se uma análise qualitativa do mapa.

Por fim, na existência de Ground Truth, será calculado o erro de localização ao longo do tempo. Esse erro será calculado pela distância euclidiana entre a posição real e a posição estimada pelo Mapa de Experiências. Da mesma forma, para comparação, será calculado o erro de localização do Dead Reckoning.

6.2 Estudo de Caso 1: Ambiente simulado de extração de petróleo

O primeiro experimento foi realizado no Simulador Subaquático UWSim (PRATS et al., 2012), em um cenário composto por dutos e *manifolds*, interconectados de forma similar a um ambiente de extração de petróleo. Abaixo dos dutos, como textura do fundo do oceano, foi utilizado um mosaico criado a partir de imagens de um ambiente real (RIBAS et al., 2012).

O robô utilizado foi o Girona 500 simulado, equipado com uma câmera direcionada para baixo, com o objetivo de capturar os dutos e a textura do fundo. Para simular um nível de incerteza no retorno dos sensores, foi utilizado um ruído gaussiano de $0.025m/s$ para a DVL e 0.5^0 para a IMU. A frequência de amostragem da DVL foi configurada para $5Hz$, metade da frequência da IMU ($10Hz$).

O ambiente simulado pode ser visto na Figura 33. Durante a navegação, o robô percorreu uma distância de aproximadamente 64 metros.

Figura 33: Ambiente utilzado nos testes do primeiro cenário

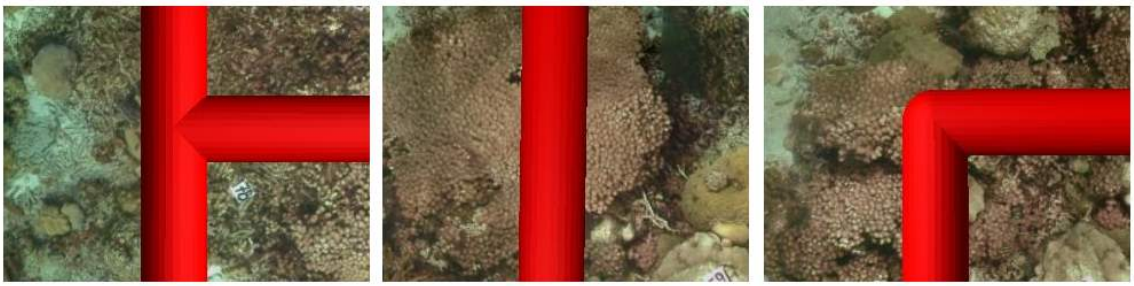

Figura 34: Imagens capturadas no ambiente simulado

Algumas imagens capturadas pelo robô podem ser vistas na Figura 34.

6.2.1 Parâmetros

A Tabela 1 apresenta os parâmetros utilizados no primeiro estudo de caso. Os parâmetros de percepção do ambiente dizem respeito ao processamento das imagens. O parâmetro Hessian Threshold é um parâmetro de configuração do detector do SURF (BAY; TUYTELAARS; GOOL, 2006). O segundo parâmetro é o número de grupos do Bag of Words. Para treinamento do Bag of Words e do FAB-MAP, foram utilizadas 360 imagens nesse estudo de caso.

O algoritmo FAB-MAP possui 4 parâmetros. Dois deles são relativos ao detector do sistema. O primeiro representa a probabilidade de falsos positivos (FP), isto é, um objeto não estar na cena e ser detectado. O segundo diz respeito a probabilidade de falsos negativos, isto é, quando um objeto está na cena e não foi detectado. O terceiro parâmetro diz respeito a probabilidade de um nó no mapa levar para um novo local. Por fim, o parâmetro de suavização é responsável por evitar uma reconhecimento errado devido ao

baixo número de imagens de treinamento com determinadas objetos.

A Place Cell possui alguns parâmetros de configuração. O número de neurônios deve ser estipulado a priori, bem como o espaço mapeado por cada neurônio, em cada dimensão. A desvio padrão de excitação lateral, configurada em número de neurônios, é representada por σ. A constante de aprendizado das sinapses entre as Local View Cells e as Place Cells é representada por λ.

O mapa de experiências possui três parâmetros, ligados ao fechamento de loop. Os parâmetros α_r e α_l indicam a importância das Local View Cells e das Place Cells, respectivamente, no fechamento de loop. A constante S_{min} representa o limiar de reconhecimento de experiências.

Tabela 1: Parâmetros de configuração do sistema no primeiro estudo de caso

Percepção do Ambiente	
Hessian Threshold	100
N grupos BoW	500
N imagens de treinamento	360
Local View Cells	
FAB-MAP detector prob. FP	0.39
FAB-MAP detector prob. FN	0.0
FAB-MAP prob. novo local	0.9
FAB-MAP suavização	0.99
Place Cells	
$n_x \times n_y \times n_z$	$20 \times 20 \times 20$
Distância entre neurônios	$0.2m \times 0.2m \times 0.2m$
σ	3 células
λ	0.1
Mapa de Experiências	
α_r	0.5
α_l	0.5
S_{min}	0.75

Fonte: Produzido pelo autor

6.2.2 Resultados

A padrão de criação e reconhecimento das Local View Cells é apresentado na Figura 35. Percebe-se, a partir do instante $t = 90s$, que não houve mais Local View Cells criadas. O algoritmo FAB-MAP foi capaz de reconhecer os locais corretamente nesse cenário.

A Figura 36 apresenta histogramas representativos de bag of words de seis imagens capturadas durante a navegação.

O mapa de experiências criado pode ser visto na Figura 37. A trajetória real percorrida pelo robô é apresentada em verde. O mapa de experiências é apresentado em

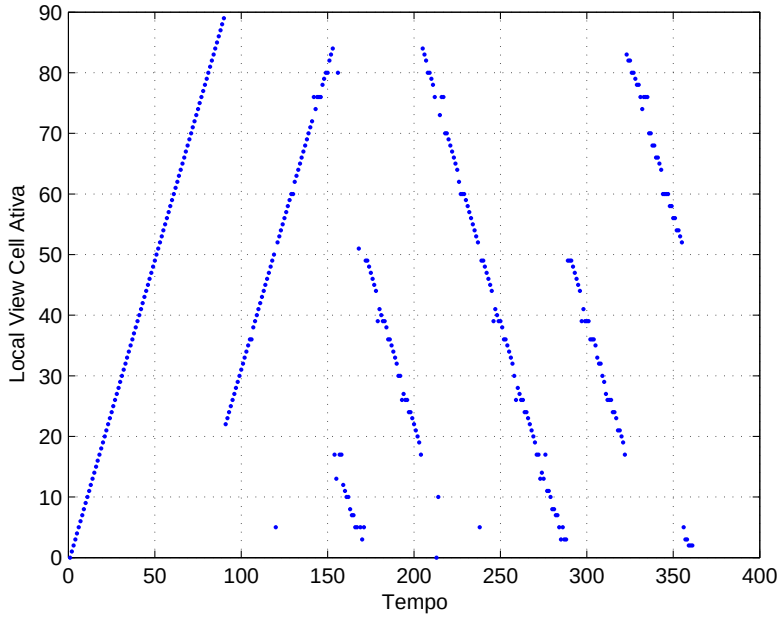

Figura 35: Criação e reconhecimendo de Local View Cells ao longo do tempo.

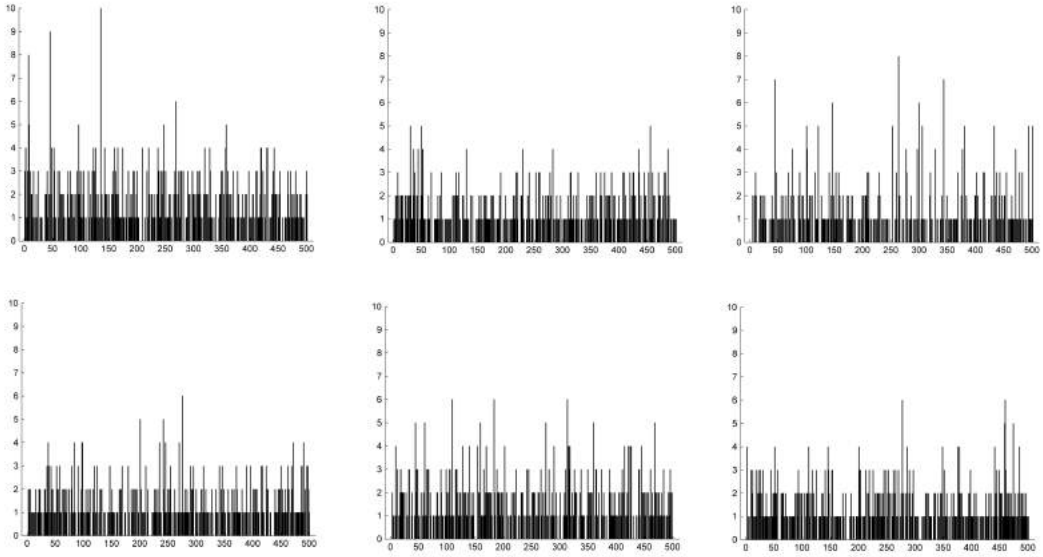

Figura 36: Histogramas do bag of words de imagens do cenário simulado.

azul, enquanto o dead reckoning é apresentado em vermelho. A trejetória criada pelo mapa de experiêncais aproxima-se à trajetória real após os eventos de fechamento de loop.

O erro de localização (Figura 38) do mapa de experiências (azul) é inferior ao dead reckoning (vermelho) a partir do instante $t = 95s$, após o primeiro fechamento de loop. Ocorre um intervalo de tempo entre o reconhecimeno de locais pelo FAB-MAP ($t = 90s$) e o fechamento de loop no mapa de experiências. Esse tempo é moderado pela Place Cell e é responsável pelo tratamento de falsos positivos no reconhecimento de locais, o que pode ocorrer devido a monotonicidade do ambiente.

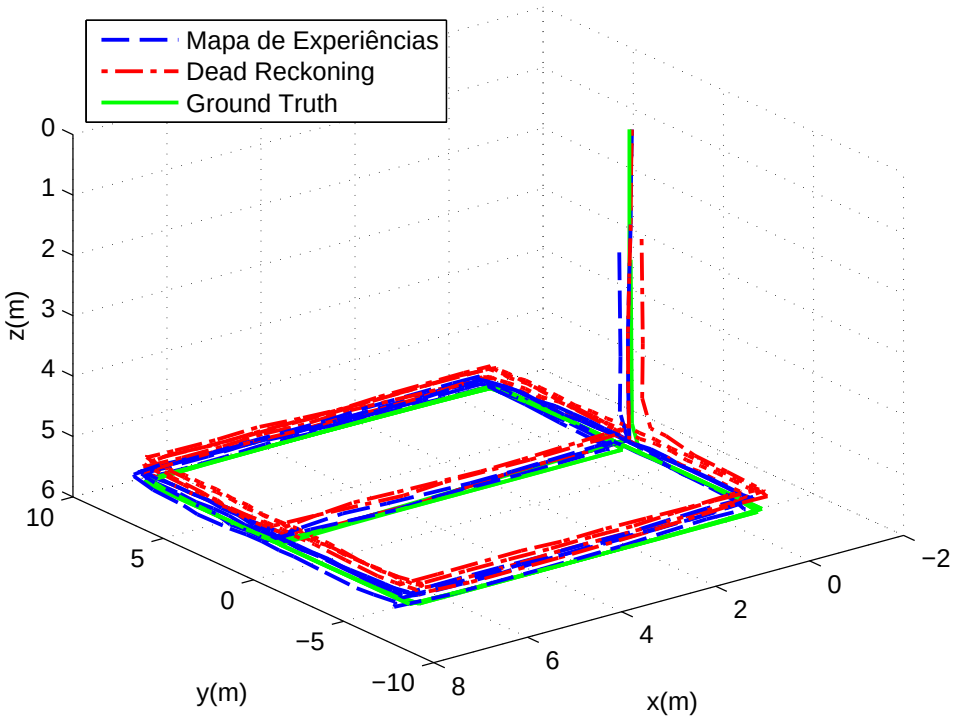

Figura 37: Resultado do ambiente simulado. Em verde está representada a trajetória real do robô(Ground Truth). Em vermelho a trajetória gerada por dead reckoning. Em azul, o mapa de experiências, gerado pelo DolphinSLAM.

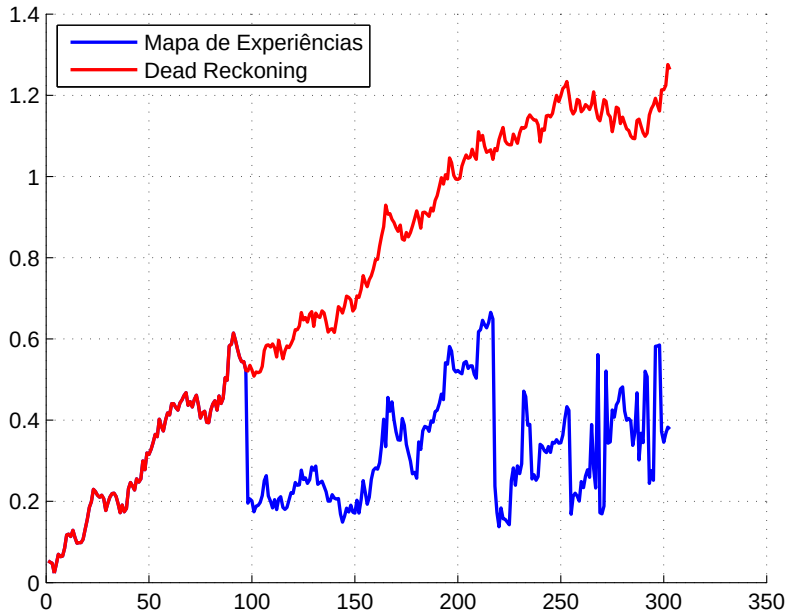

Figura 38: Erro de localização. Em vermelho o erro do dead reckoning. Em azul, o erro de localização estimada pelo mapa de experiências.

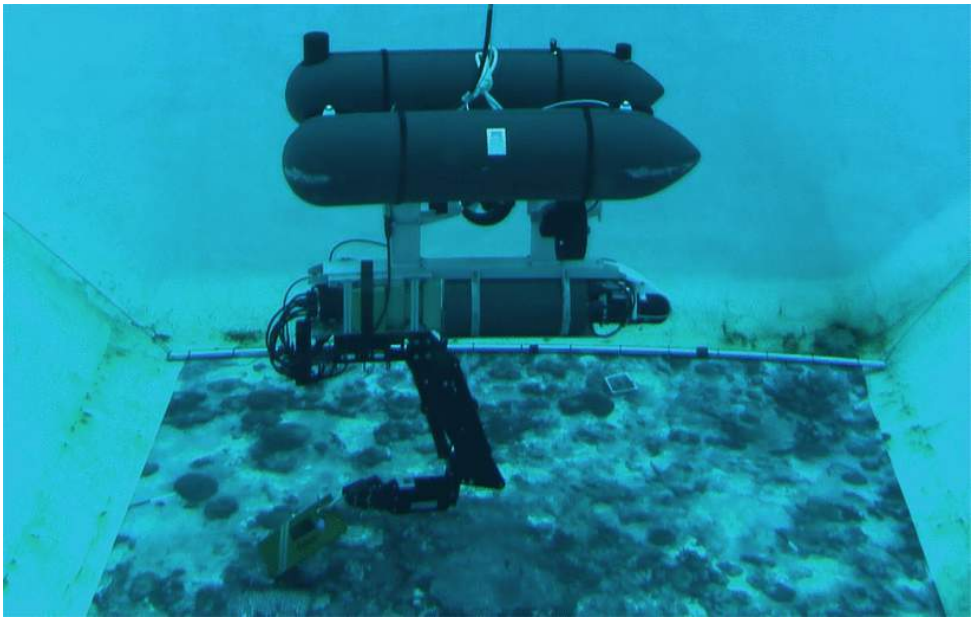

Figura 39: Cenário utilizado no segundo estudo de caso, em que pode-se visualizar a piscina com a textura no fundo e o robô Girona 500. Imagem extraída de (RIBAS et al., 2012)

6.2.3 Discussão

O algoritmo DolphinSLAM teve um comportamento excelente no ambiente simulado, sendo capaz de reconhecer lugares previamente visitados, devido ao FAB-MAP, e reduzir o erro de localização

6.3 Estudo de Caso 2: Piscina

O segundo estudo de caso foi realizado em um *dataset* da piscina existente no Centro de Investigação em Robótica Submarina (*Centre d'Investigació en Robòtica Submarina* - CIRS) da Universidade de Girona, na Espanha (RIBAS et al., 2012). O robô Girona 500 foi utilizado para geração dos dados, equipado com os mesmo sensores do experimento anterior: DVL e IMU. O objetivo é a validação do sistema com dados sensoriais reais.

O ambiente de testes, juntamente com o robô utilizado, podem ser vistos na Figura 39

6.3.1 Parâmetros

A Tabela 2 apresenta os parâmetros utilizados no segundo estudo de caso.

Tabela 2: Parâmetros de configuração do sistema no segundo estudo de caso

Percepção do Ambiente	
Hessian Threshold	1000
N grupos BoW	500
N imagens de treinamento	41
Local View Cells	
FAB-MAP detector prob. FP	0.39
FAB-MAP detector prob. FN	0.0
FAB-MAP prob. novo local	0.9
FAB-MAP suavização	0.99
Place Cells	
$n_x \times n_y \times n_z$	$20 \times 20 \times 20$
Distância entre neurônios	$0.2m \times 0.2m \times 0.2m$
σ	3 células
λ	0.4
Mapa de Experiências	
α_r	0.5
α_l	0.5
S_{min}	0.8

Fonte: Produzido pelo autor

6.3.2 Resultados

A Figura 40 apresenta as *Local View Cells* ativas ao longo do tempo. Pode-se notar que o algoritmo FAB-MAP não foi capaz de detectar a presença de loops. Dessa forma, o mapa de experiências (Figura 42 criado pelo sistema DolphinSLAM é igual ao dead reckoning, não havendo redução do erro de odometria.

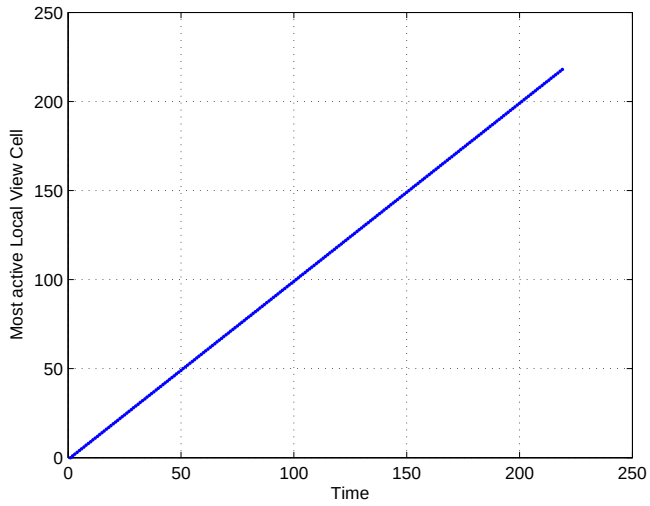

Figura 40: Local View Cells ativas ao longo do tempo.

A Figura 41 apresenta histogramas representativos de bag of words de seis imagens capturadas durante a navegação.

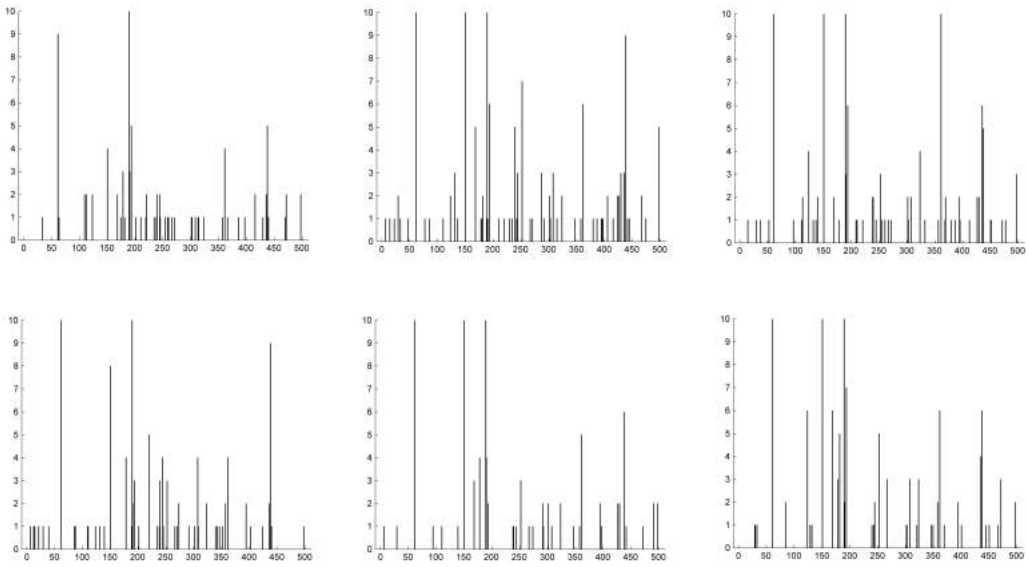

Figura 41: Histogramas do bag of words de imagens do segundo estudo de caso.

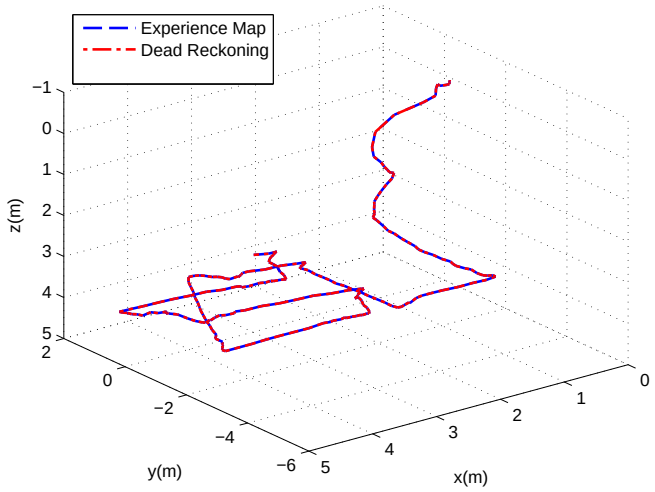

Figura 42: Resultado do segundo cenário. Em azul, mapa de experiências. Em vermelho pode ser visto o dead reckoning.

6.3.3 Discussão

O algoritmo FAB-MAP não foi capaz de realizar o correto fechamento de loop no ambiente. Sua falha deve-se ao baixo conjunto de treinamento disponível (41 imagens), não sendo possível uma boa aproximação da distribuição de probabilidades do modelo de geração de palavras do vocabulário. Além disso, diferente do ambiente apresentado em (CUMMINS; NEWMAN, 2011), as features detectadas no ambiente subaquático, no nosso caso corais, são muito similares entre si. Além disso, não há a existência de formas geométricas no ambiente, tornando a tarefa de reconhecimento de locais mais complexa.

6.4 Estudo de Caso 3: Marina

O terceiro experimento objetiva fazer um estudo da viabilidade de aplicação do sistema com dados de sonar. Para isso, foi criado o *dataset* ARACATI com o robô Seabotix LBV300-5, equipado com um sonar Teledyne BlueView P900-130. O robô foi fixado na parte inferior de uma prancha, de modo a manter-se na superfície para recebimento de dados do DGPS, utilizado como *Ground Truth*. Devido a ausência de sensores propioceptivos, as informações motoras foram geradas a partir dos dados de localização do DGPS. Foi gerado um sinal de velocidade a partir da derivação dos sinais de posição, sujeito a um ruído gaussiano de $0.1m/s$. O sistema foi executado a uma taxa de amostragem de 1 frame por segundo.

Uma vista aérea do ambiente de testes é apresentada na Figura 43, juntamente com o robô utilizado.

Figura 43: Cenário utilizado no terceiro estudo de caso: a) Visualização do Google Earth da área onde o experimento foi realizado b)Uma vista aproximada do ambiente, composto por embarcações, pedras e postes. c) Caminho percorrido pelo robô, tendo início no ponto verde e terminando no ponto azul.

Algumas imagens capturadas pelo robô, após a etapa de processamento podem ser visualizadas na Figura 46. Cada região em branco na imagem será tratada como um objeto, sendo descrito pelos 7 momentos de Hu (HU, 1962).

6.4.1 Parâmetros

A Tabela 3 apresenta os parâmetros utilizados no terceiro estudo de caso.

6.4.2 Resultados

As Local View Cells ativas ao longo do tempo são apresentadas na Figura 47.

Figura 44: Imagem de sonar capturada no ambiente.

Figura 45: O ROV utilizado no experimento

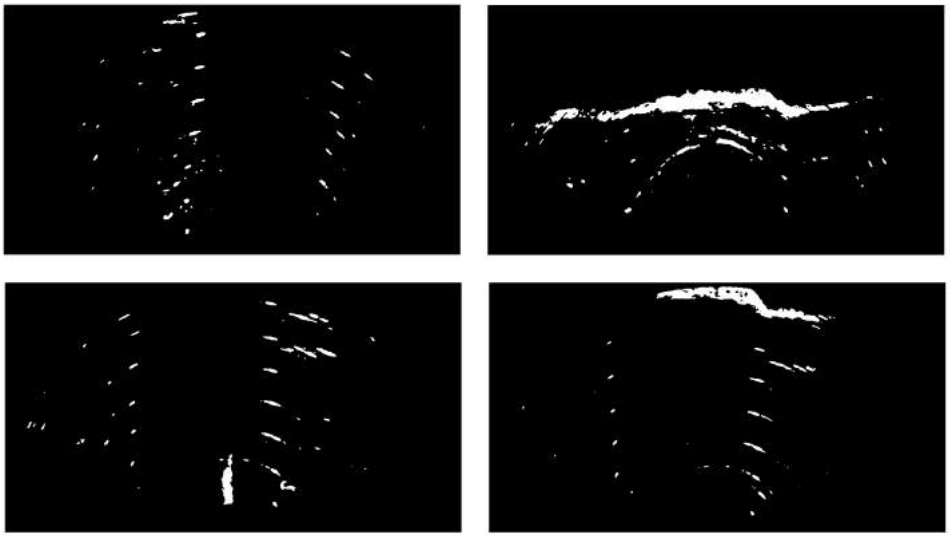

Figura 46: Imagens de sonar após a etapa de processamento.

A Figura 48 apresenta histogramas representativos de bag of words de seis imagens capturadas durante a navegação.

O mapa gerado pelo sistema é apresentado na Figura 49.

Os erros de localização do sistema e do dead reckoning são mostrado na Figura 50.

Tabela 3: Parâmetros de configuração do sistema no terceiro estudo de caso

Percepção do Ambiente	
N grupos BoW	200
N imagens de treinamento	188
Local View Cells	
FAB-MAP detector prob. FP	0.39
FAB-MAP detector prob. FN	0.0
FAB-MAP prob. novo local	0.9
FAB-MAP suavização	0.99
Place Cells	
$n_x \times n_y \times n_z$	$15 \times 15 \times 15$
Distância entre neurônios	$0.25m \times 0.25m \times 0.25m$
σ	2 células
λ	0.25
Mapa de Experiências	
α_r	0.5
α_l	0.5
S_{min}	0.75

Fonte: Produzido pelo autor

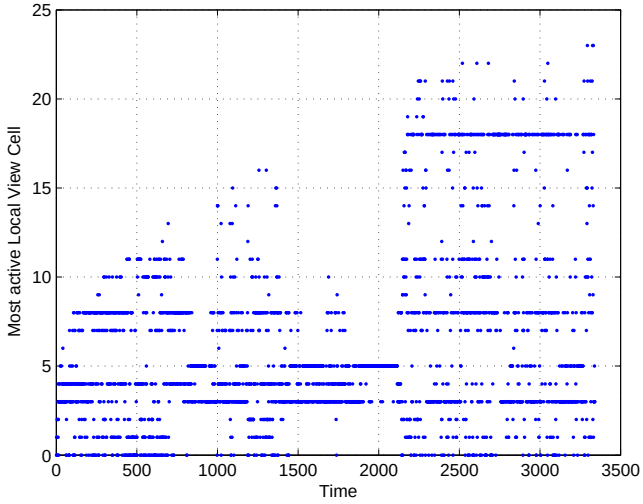

Figura 47: Local View Cells ativas ao longo do tempo

Nota-se, devido a incapacidade do FAB-MAP reconhecer lugares nesse ambiente, que o erro de localização é igual para as duas trajetórias.

6.4.3 Discussão

O algorimto FAB-MAP também não apresentou bons resultados nesse cenário. O principal problema nesse ambiente deve-se à alta similaridade entre as cenas capturas no abmiente (estacas e uma fileira de pedras). Além disso, o ruído inerente à leitura sensorial faz com que os descritores sejam instáveis, sendo representados por diferentes palavras em

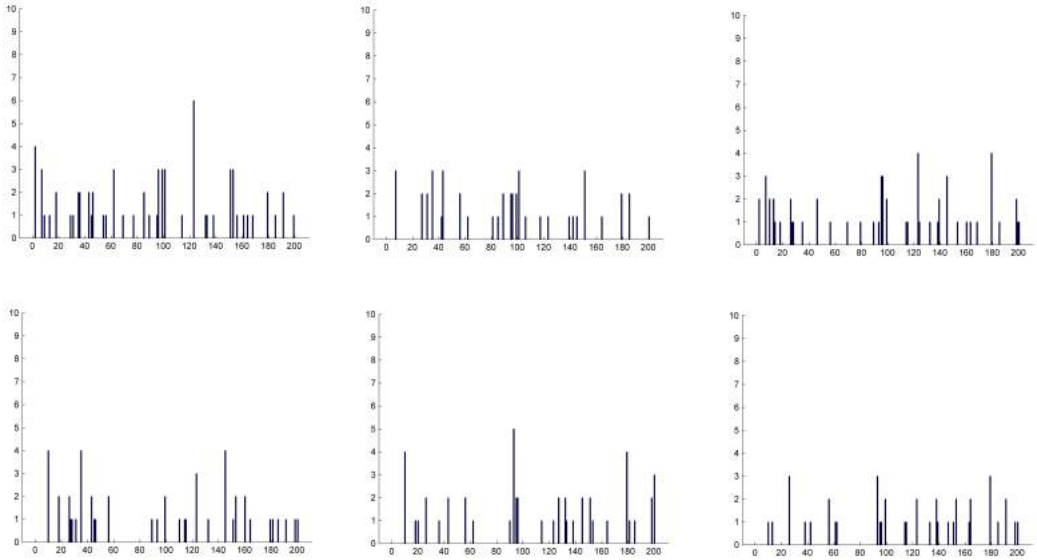

Figura 48: Histogramas do bag of words das imagens de sonar.

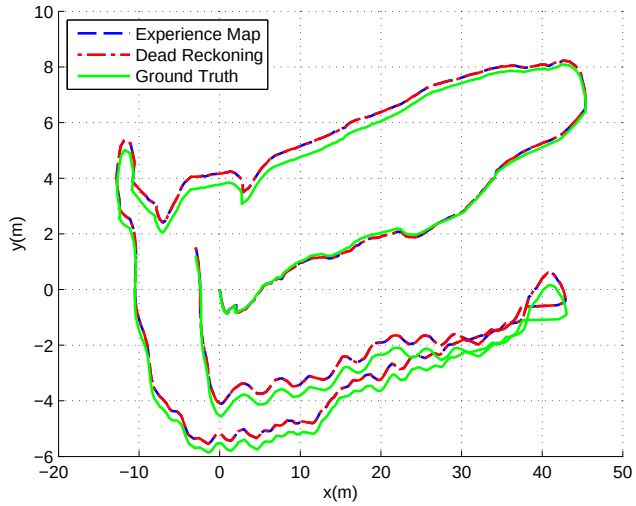

Figura 49: Em azul pode ser visto o mapa gerado pelo DolphinSLAM. Em vermelho, pode-se visualizar o dead reckoning. O ground truth é apresentado em verde

imagens consecutivas.

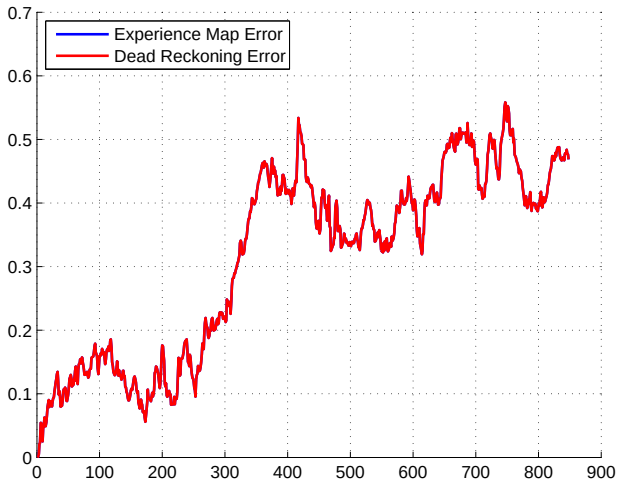

Figura 50: Erro de localização. Em azul, erro do mapa de experiências. Em vermelho, erro de localização do dead reckoning.

7 Conclusão

O presente trabalho apresentou um novo método para resolução do problema de SLAM em ambientes subaquáticos, partindo da modelagem de neurônios especiais encontrados no cérebro dos mamíferos.

Foi realizado um estudo dos principais sensores utilizados pela robótica subaquática e dos desafios relacionados ao desenvolvimento de algoritmos para esses ambientes: turbidez da água, não estruturação dos ambientes, monotonicidade dos objetos encontrados, inexistência de sistemas de posicionamento global e complexidade no controle robótico devido ao arraste.

O sistema desenvolvido baseou-se em uma metodologia bioinspirada no cérebro dos mamíferos, desenvolvendo um modelo computacional para as *Place Cells* tridimensionais, responsáveis pela estimativa de posição do robô no ambiente. Utilizou-se o algoritmo RatSLAM como base para integração do modelo a um sistema completo para SLAM, tornando o DolphinSLAM apto ao mapeamento de ambientes vastos.

A utilização do algoritmo FAB-MAP possibilita uma maior robustez de reconhecimento de locais em ambientes monótonos e com variação de iluminação, devido a turbidez do água.

O primeiro estudo de caso demonstrou a viabilidade da proposta, onde o DolphinSLAM foi capaz de mapear um ambiente de distribuição de petróleo, composto por dutos de distribuição e uma textura real de corais. O erro de localização do robô, após o primeiro fechamento de loop, manteve-se inferior a metade do erro de localização por dead reckoning. A complexidade do sistema possui um crescimento linear ao número de landmarks criados no mapa, sendo aplicável ao mapeamento de ambientes vastos.

No segundo e terceiro ambientes o módulo de percepção não foi capaz de reconhecer locais previamente visitados. Isso deve-se ao fato dos datasets serem pequenos, impossibilitando ao FAB-MAP um bom conjunto de testes para o treinamento do modelo de geração das palavras do vocabulário.

Como trabalhos futuros, vislumbra-se a aplicação do método em cenários reais similares aos simulados, onde estruturas para exploração de petróleo estejam presentes. Também busca-se aprofundar os estudos relacionados a pistas sensoriais multi-modais, integrando informações acústicas e ópticas.

Referências

AMBIENTE Subaquatico. 2014. Disponível em: <http://www.macleans.ca/economy/business/maybe-its-better-down-where-its-wetter>. Acesso em: 20 de agosto de 2014. Citado na página 34.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. [S.l.], 2007. p. 1027–1035. Citado 2 vezes nas páginas 58 e 65.

BARKBY, S. et al. Bathymetric particle filter slam using trajectory maps. *The International Journal of Robotics Research*, SAGE Publications, p. 1409–1430, 2012. Citado na página 49.

BARRERA, A.; WEITZENFELD, A. Biologically-inspired robot spatial cognition based on rat neurophysiological studies. *Autonomous Robots*, Springer, v. 25, n. 1-2, p. 147–169, 2008. Citado na página 33.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: *Computer Vision–ECCV 2006*. [S.l.]: Springer, 2006. p. 404–417. Citado 3 vezes nas páginas 49, 65 e 79.

BRADSKI, G. Open computer vision. *Dr. Dobb's Journal of Software Tools*, 2000. Citado na página 77.

BRASILIS, R. Exploração do pré-sal alavanca inovação tecnológica. *Revista Brasilis*, 2012. Citado na página 17.

BURGUERA, A.; GONZàLEZ, Y.; OLIVER, G. Underwater slam with robocentric trajectory using a mechanically scanned imaging sonar. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2011. p. 3577–3582. Citado na página 49.

CUMMINS, M.; NEWMAN, P. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, SAGE Publications, v. 27, n. 6, p. 647–665, 2008. Citado 4 vezes nas páginas 20, 51, 58 e 59.

CUMMINS, M.; NEWMAN, P. Accelerating fab-map with concentration inequalities. *IEEE Transactions on Robotics*, v. 26, n. 6, p. 1042–1050, 2010. Citado na página 59.

CUMMINS, M.; NEWMAN, P. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, SAGE Publications, v. 30, n. 9, p. 1100–1123, 2011. Citado 3 vezes nas páginas 59, 60 e 85.

DISSANAYAKE, G. et al. Map management for efficient simultaneous localization and mapping (slam). *Auton. Robots*, Kluwer Academic Publishers, v. 12, n. 3, p. 267–286, 2002. Citado na página 32.

DOUCET, A. et al. Rao-blackwellised particle filtering for dynamic bayesian networks. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. [S.l.]: Morgan Kaufmann Publishers Inc., 2000. (UAI '00), p. 176–183. Citado na página 31.

DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: Part i. *IEEE Robotics & Automation Magazine*, IEEE, v. 13, n. 2, p. 99–110, 2006. Citado 3 vezes nas páginas 22, 23 e 29.

DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping (slam): part ii. *Robotics Automation Magazine, IEEE*, v. 13, n. 3, p. 108–117, 2006. Citado na página 27.

DVL Explorer. 2014. Disponível em: <http://www.rdinstruments.com/explorer.aspx>. Acesso em: 20 de agosto de 2014. Citado na página 41.

ESTRADA, C.; NEIRA, J.; TARDÓS, J. D. Hierarchical slam: real-time accurate mapping of large environments. *Robotics, IEEE Transactions on*, IEEE, v. 21, n. 4, p. 588–596, 2005. Citado na página 32.

EUSTICE, R.; PIZARRO, O.; SINGH, H. Visually augmented navigation in an unstructured environment using a delayed state history. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2004. v. 1, p. 25–32. Citado 2 vezes nas páginas 48 e 49.

EUSTICE, R. M.; PIZARRO, O.; SINGH, H. Visually augmented navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, IEEE, v. 33, n. 2, p. 103–122, 2008. Citado na página 49.

FAIRFIELD, N.; KANTOR, G.; WETTERGREEN, D. Real-time slam with octree evidence grids for exploration in underwater tunnels. *Journal of Field Robotics*, Wiley Online Library, v. 24, n. 1-2, p. 03–21, 2007. Citado na página 48.

FERREIRA, F. et al. Real-time optical slam-based mosaicking for unmanned underwater vehicles. *Intelligent Service Robotics*, Springer-Verlag, v. 5, n. 1, p. 55–71, 2012. Citado na página 49.

GARCIA, R.; GRACIAS, N. Detection of interest points in turbid underwater images. In: IEEE. *OCEANS, 2011 IEEE-Spain*. [S.l.], 2011. p. 1–9. Citado na página 65.

GIOCOMO, L. M.; MOSER, M.-B.; MOSER, E. I. Computational models of grid cells. *Neuron*, Elsevier, v. 71, n. 4, p. 589–603, 2011. Citado na página 70.

GLOVER, A. J. et al. Fab-map+ ratslam: appearance-based slam for multiple times of day. In: IEEE. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. [S.l.], 2010. p. 3507–3512. Citado na página 60.

GNU GLP V2. 2015. Disponível em: <http://choosealicense.com/licenses/gpl-2.0/>. Acesso em: 15 de fevereiro de 2015. Citado na página 77.

HAFTING, T. et al. Microstructure of a spatial map in the entorhinal cortex. *Nature*, Nature Publishing Group, v. 436, n. 7052, p. 801–806, 2005. Citado 2 vezes nas páginas 19 e 52.

HU, M.-K. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, IEEE, v. 8, n. 2, p. 179–187, 1962. Citado 2 vezes nas páginas 65 e 86.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, v. 82, n. Series D, p. 35–45, 1960. Citado na página 28.

KIM, A.; EUSTICE, R. M. Real-time visual slam for autonomous underwater hull inspection using visual saliency. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2013. v. 1, p. 2. Citado na página 50.

KINSEY, J. C.; EUSTICE, R. M.; WHITCOMB, L. L. A survey of underwater vehicle navigation: Recent advances and new challenges. In: *IFAC Conference of Manoeuvering and Control of Marine Craft*. [S.l.: s.n.], 2006. Citado na página 39.

KOHONEN, T. The self-organizing map. *Proceedings of the IEEE*, IEEE, v. 78, n. 9, p. 1464–1480, 1990. Citado na página 56.

LASER Subaquático. 2014. Disponível em: <http://www.2grobotics.com/solutions/products/underwater-laser-scanner-uls-200/>. Acesso em: 20 de agosto de 2014. Citado na página 43.

LEONARD, J. L.; CARPENTER, R. N.; FEDER, H. J. S. Stochastic mapping using forward look sonar. *Robotica*, Cambridge University Press, v. 19, n. 5, p. 467–480, 2001. Citado 2 vezes nas páginas 46 e 47.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157. Citado na página 58.

MADDERN, W. et al. Augmenting ratslam using fab-map-based visual data association. In: AUSTRALIAN ROBOTICS AND AUTOMATION ASSOCIATION INC. *Proceedings of Australasian Conference on Robotics and Automation 2009*. [S.l.], 2009. Citado na página 60.

MAHON, I. et al. Efficient view-based slam using visual loop closures. *IEEE Transactions on Robotics*, v. 24, n. 5, p. 1002–1014, 2008. Citado na página 49.

MALLIOS, A. et al. Ekf-slam for auv navigation under probabilistic sonar scan-matching. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2010. p. 4404–4411. Citado 2 vezes nas páginas 45 e 49.

MASUMOTO, Y. *Global Positioning System (GPS)*. [S.l.]: Google Patents, 1993. US Patent 5,210,540. Citado na página 34.

MCNAUGHTON, B. L. et al. Path integration and the neural basis of the 'cognitive map'. *Nature Reviews Neuroscience*, v. 7, n. 8, p. 663–678, 2006. ISSN 1471-003X. Disponível em: <http://dx.doi.org/10.1038/nrn1932>. Citado 4 vezes nas páginas 19, 51, 52 e 53.

METROPOLIS, N.; ULAM, S. The monte carlo method. *Journal of the American statistical association*, Taylor & Francis Group, v. 44, n. 247, p. 335–341, 1949. Citado na página 30.

MILFORD, M. J.; WYETH, G. F. Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System. *IEEE Transactions on Robotics*, IEEE, v. 24, n. 5, p. 1038–1053, 2008. Citado 6 vezes nas páginas 20, 33, 51, 54, 55 e 57.

MILFORD, M. J.; WYETH, G. F.; PRASSER, D. Ratslam: a hippocampal model for simultaneous localization and mapping. In: IEEE. *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*. [S.l.], 2004. v. 1, p. 403–408. Citado na página 57.

MONTEMERLO, M.; THRUN, S. Simultaneous localization and mapping with unknown data association using fastslam. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2003. p. 1985–1991. Citado na página 32.

MONTEMERLO, M. et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In: *AAAI/IAAI*. [S.l.: s.n.], 2002. p. 593–598. Citado na página 31.

MOSER, E.; KROPFF, E.; MOSER, M. Place cells, grid cells, and the brain's spatial representation system. *Annu. Rev. Neurosci.*, Annual Reviews, v. 31, p. 69–89, 2008. Citado 4 vezes nas páginas 51, 52, 53 e 54.

NEWMAN, P.; LEONARD, J. Pure range-only sub-sea slam. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2003. p. 1921–1926. Citado 2 vezes nas páginas 47 e 48.

NEWMAN, P. M.; LEONARD, J.; RIKOSKI, R. J. Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar. In: *Eleventh International Symposium on Robotics Research*. [S.l.]: Springer Verlag, 2003. p. 409–420. Citado na página 47.

NISTER, D.; STEWENIUS, H. Scalable recognition with a vocabulary tree. In: IEEE. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* . [S.l.], 2006. v. 2, p. 2161–2168. Citado na página 58.

O'KEEFE, J.; DOSTROVSKY, J. The hippocampus as a spatial map: Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, Elsevier Science, 1971. Citado 2 vezes nas páginas 19 e 51.

O'KEEFE, J.; NADEL, L. The hippocampus as a cognitive map. *Behavioral and Brain Sciences*, Cambridge Univ Press, v. 2, n. 04, p. 487–494, 1979. Citado na página 52.

OLSON, E.; LEONARD, J.; TELLER, S. Robust range-only beacon localization. *IEEE Journal of Oceanic Engineering*, v. 31, n. 4, p. 949–958, 2006. Citado na página 48.

PAULL, L. et al. Auv navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, v. 39, n. 1, p. 131–149, Jan 2014. Citado na página 35.

PRASSER, D.; MILFORD, M.; WYETH, G. Outdoor simultaneous localisation and mapping using ratslam. In: SPRINGER. *Field and Service Robotics*. [S.l.], 2006. p. 143–154. Citado na página 57.

PRATS, M. et al. An open source tool for simulation and supervision of underwater intervention missions. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2012. p. 2577–2582. Citado na página 78.

QUIGLEY, M. et al. ROS: an open-source robot operating system. In: *IEEE ICRA - Workshop on Open Source Software*. [S.l.: s.n.], 2009. Citado na página 77.

RIBAS, D. et al. Girona 500 auv: From survey to intervention. *IEEE/ASME Transactions on Mechatronics*, v. 17, n. 1, p. 46–53, 2012. Citado 2 vezes nas páginas 78 e 83.

RIBAS, D. et al. Underwater slam in man-made structured environments. *Journal of Field Robotics*, v. 25, n. 11-12, p. 898–921, October 2008. Citado 3 vezes nas páginas 32, 45 e 49.

ROMAN, C. N.; SINGH, H. Improved vehicle based multibeam bathymetry using sub-maps and slam. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.: s.n.], 2005. p. 3662–3669. Citado na página 48.

RUIZ, I. T.; PETILLOT, Y.; LANE, D. Concurrent mapping and localization using sidescan sonar. *IEEE Journal of Oceanic Engineering*, v. 29, n. 2, p. 442–456, 2004. Citado na página 47.

SCHÄLING, B. *The boost C++ libraries*. [S.l.]: Boris Schäling, 2011. Citado na página 77.

SàEZ, J. M. et al. Underwater 3d slam through entropy minimization. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2006. p. 3562–3567. Citado na página 48.

SIVIC, J.; ZISSERMAN, A. Video google: A text retrieval approach to object matching in videos. In: IEEE. *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. [S.l.], 2003. p. 1470–1477. Citado 3 vezes nas páginas 20, 50 e 58.

SMITH, R. C.; CHEESEMAN, P. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, Sage Publications, v. 5, n. 4, p. 56–68, 1986. Citado na página 28.

SONAR de Imageamento Mecânico. 2014. Disponível em: <http://www.tritech.co.uk/news-article/tritech-launches-seaking-hammerhead-dst-sonar-at-oceanology-international-2010>. Acesso em: 20 de agosto de 2014. Citado na página 46.

SONAR do tipo Sidescan. 2014. Disponível em: <http://www.seascape.nl/support/sidescan-sonar>. Acesso em: 20 de agosto de 2014. Citado na página 45.

SONARES Multi Beam e Single Beam. 2014. Disponível em: <http://oceanexplorer.noaa.gov/okeanos/explorations/ex1105/media/ex1105_4.html>. Acesso em: 20 de agosto de 2014. Citado 2 vezes nas páginas 44 e 45.

STAUFFER, N. W. AUVs: From idea to implementation. *Energy Futures*, MIT Energy Initiative, 2011. Citado na página 17.

STAUFFER, N. W. Deepsea oil and gas recovery: Designing robots that can help. *Energy Futures*, MIT Energy Initiative, 2011. Citado na página 17.

STRINGER, S. et al. Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells. *Network: Computation in Neural Systems*, Informa UK Ltd UK, v. 13, n. 4, p. 429–446, 2002. Citado 4 vezes nas páginas 55, 62, 70 e 74.

STRINGER, S. et al. Self-organizing continuous attractor networks and path integration: one-dimensional models of head direction cells. *Network: Computation in Neural Systems*, Informa UK Ltd UK, v. 13, n. 2, p. 217–242, 2002. Citado 2 vezes nas páginas 55 e 74.

TAUBE, J.; MULLER, R.; JR, J. R. Head-direction cells recorded from the postsubiculum in freely moving rats. i. description and quantitative analysis. *The Journal of Neuroscience*, Soc Neuroscience, v. 10, n. 2, p. 420–435, 1990. Citado 2 vezes nas páginas 19 e 52.

TELEDYNE Blueview P900. 2014. Disponível em: <http://www.seatronics-group.com/equipment-rental/geophysical-15/multibeam-sonars-7/teledyne-blueview-p900-130-148>. Acesso em: 20 de agosto de 2014. Citado na página 46.

TENA, I. et al. Feature extraction and data association for auv concurrent mapping and localisation. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2001. p. 2785–2790. Citado 2 vezes nas páginas 45 e 47.

TEYNOR, A.; BURKHARDT, H. Fast codebook generation by sequential data analysis for object classification. In: *Advances in Visual Computing*. [S.l.]: Springer, 2007. p. 610–620. Citado na página 58.

THRUN, S. Simultaneous localization and mapping. *Robotics and cognitive approaches to spatial mapping*, Springer, p. 13–41, 2008. Citado 8 vezes nas páginas 22, 25, 26, 27, 28, 29, 30 e 32.

USBL - SBL - LBL (Acoustic Positioning). 2014. Disponível em: <http://www.amloceanographic.com/Technical-Demo/USBL-SBL-LBL_2>. Acesso em: 17 de agosto de 2014. Citado 3 vezes nas páginas 36, 37 e 38.

WALTER, M.; HOVER, F.; LEONARD, J. Slam for ship hull inspection using exactly sparse extended information filters. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2008. p. 1463–1470. Citado 2 vezes nas páginas 46 e 49.

WILLIAMS, S.; MAHON, I. Simultaneous localisation and mapping on the great barrier reef. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2004. v. 2, p. 1771–1776. Citado na página 48.

WILLIAMS, S. et al. Autonomous underwater simultaneous localisation and map building. In: *IEEE International Conference on Robotics and Automation (ICRA)*. [S.l.: s.n.], 2000. v. 2, p. 1793–1798. Citado 3 vezes nas páginas 45, 46 e 47.

WYETH, G.; MILFORD, M. Spatial cognition for robots. *Robotics & Automation Magazine, IEEE*, IEEE, v. 16, n. 3, p. 24–32, 2009. Citado 2 vezes nas páginas 53 e 58.

YARTSEV, M. M.; ULANOVSKY, N. Representation of three-dimensional space in the hippocampus of flying bats. *Science*, American Association for the Advancement of Science, v. 340, n. 6130, p. 367–372, 2013. Citado 2 vezes nas páginas 62 e 70.