

DISSERTAÇÃO DE MESTRADO Nº 844

**DESENVOLVIMENTO DE UM SISTEMA DE NAVEGAÇÃO EM AMBIENTES
INTERNOS PARA UM ROBÔ PESSOAL**

Diana Sabina Albán Peñafiel

DATA DA DEFESA: 04/12/2014

Universidade Federal de Minas Gerais
Programa de Pós-Graduação em Engenharia Elétrica
Laboratório de Sistemas de Computação e Robótica
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil
Fone: +55 31 3409-3470



Desenvolvimento de um Sistema de Navegação em Ambientes Internos para um Robô Pessoal

Diana Sabina Albán Peñafiel

Orientador: Prof. Guilherme Augusto Silva Pereira, Doutor

Belo Horizonte, 04 de Dezembro de 2014

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**DESENVOLVIMENTO DE UM SISTEMA DE NAVEGAÇÃO EM
AMBIENTES INTERNOS PARA UM ROBÔ PESSOAL**

Diana Sabina Albán Peñafiel

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Guilherme Augusto Silva Pereira

Belo Horizonte - MG

Dezembro de 2014


"Desenvolvimento de um Sistema de Navegação em Ambientes Internos para um Robô Pessoal"

Diana Sabina Albán Peñafiel


Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 04 de dezembro de 2014.


Por:



Prof. Dr. Guilherme Augusto Silva Pereira
DEE (UFMG) - Orientador



Prof. Dr. Luciano Cunha de Araújo Pimenta
DELT (UFMG)



Prof. Dr. Erickson Rangel do Nascimento
DCC (UFMG)

Dedicatória

Em memoria a meu querido pai, Fausto Albán, o Negrito

Para meu maior tesouro, minha mãe, Rosa Peñafiel

A meus amados irmãos, Matilde, Bolivar, Patricio

A minha bela cachorra Pekitas

Agradecimentos

É chegado o momento de agradecer pelas lembranças, o apoio e o carinho que recebi ao longo deste período de mestrado.

Ao Pai Celestial, a mim Deus, por ter me dado o valor, força para lutar e conseguir este sonho, por me permitir acordar dia a dia e ver o céu, o sol, a chuva que ele fez para mim. Como não ser grata com ele por me permitir ter na minha vida pessoas tão especiais como são meus familiares, meu namorado e meus amigos. Sei que esta é a ultima parte a ser escrita, mas considero que é a mais importante, porque lembra-se de todos os momentos bons e ruins pelos que atravessei e das pessoas e fatos que fizeram me continuar, lutar e ser feliz.

A minha família por ter me incentivado, ajudado, apoiado e proferido as palavras certas nas horas mais difíceis. A meu Pai, Fausto Albán, por ter-me compreendido quando queria sair do país para estudar, embora ele estava doente. Obrigada Pai, eu sei que você não vai ler esta mensagem, mas quero te agradecer porque sei que sempre cuidou de mim desde o céu.

A minha mãe, Rosa Peñafiel, porque sem seu constante trabalho e sacrifício, eu não tivesse chegado a este ponto crucial da minha vida. Obrigada por sempre esperar por minha chamada, por sentir saudades, lembrar sempre de mim e continuar junto a mim.

A minha irmã, minha segunda mãe, minha amiga e confidente, Matilde Albán, obrigado por ser cúmplice de todas as loucuras que vêm à mente, por me apoiar e me dar sempre encorajo, força para nunca me deixar cair. Obrigado por ser minha *manager*, por lutar muito e provar-me que não consegue-se nada sem sacrifício.

Para meus irmãos e Patrício e Bolívar Albán, obrigada por ficaram sempre juntos apesar dos problemas, por me deixar ter a satisfação de tê-los como uma família unida.

A meu amosho Fredy, obrigada por ser meu guia, meu cozinheiro, meu amigo, meu cúmplice, meu colega, etc. Amosho, obrigada por ter a paciência para cuidar de mim, suportar o meu temperamento e por mostrar o seu grande amor apesar dos contratemplos.

Também gostaria de agradecer a todas as pessoas que são muito especial na minha

vida, meus amigos e amigas da UFMG, obrigado a todos por ser "minha família" durante a minha estadia no Brasil em especial em Belo Horizonte, com vocês meninos e meninas tudo foi melhor. Valciene, Ernesto, Wendy, Fabricio, Tamara, Renato, Douglas, Alcy, Lianny, Ana, Caio, Marcel, Natalia, Bryan, Diego, Maria, Camilo, Erika, Polyana, Marco, Reza, Kossar, Carlos, Hayanna, Samira, David, Estevão, Tcharles, Lucas, Alexandre, Gaby, Danilo, Gabriel, Heitor, Henrique, Carlos, Wilmer e todos os outros colegas por ser umas excelentes pessoas.

Gostaria de manifestar a minha imensa gratidão ao professor Guilherme Pereira, respectivo orientador deste projeto, pelos ensinamentos, cobranças, apoio e confiança em mim depositada. Também quero agradecer aos professores da linha de pesquisa Controle, Automação, e Robótica do PPGEE pela conhecimento prestado durante meus estudos de Mestrado.

Agradeço o apoio financeiro e o suporte do Senescyt no Equador pela bolsa de estudos de mestrado, ao CNPq, Capes e FAPEMIG pelo apoio ao projeto, com materiais, e recursos.

A todos vocês muito obrigada, sei que sozinha eu jamais teria conseguido.

“Não jogue seus sonhos pela janela só porque hoje não conseguiu conquistá-los, espere pelo amanhã para tentar de novo. Não desista tao fácil, são seus sonhos os culpados de você ter esperança e força de vontade para seguir em frente.”

Bruna Martins

Resumo

Esta dissertação apresenta o desenvolvimento de um sistema de navegação em ambientes internos para o robô pessoal chamado MARIA. O sistema de navegação permite ao robô deslocar-se desde uma pose inicial até uma pose final dentro de um ambiente interno previamente mapeado, ao receber instruções de voz. A metodologia envolve: (i) a geração de mapas do ambiente mediante a aquisição dos dados sensoriais de um sensor Kinect e um sensor de proximidade a laser, (ii) a estimativa da pose do robô no ambiente mediante o algoritmo de localização Monte Carlo (AMCL), (iii) a implementação de um planejador de caminho por frentes de ondas e (iv) a geração e a execução de rotas livres de colisão. O mapa bidimensional gerado a partir do laser é utilizado para estimar a pose do robô dentro do ambiente. O mapa tridimensional gerado com o uso do Kinect é utilizado para realizar uma projeção bidimensional que considera os obstáculos presentes no ambiente até o tamanho do robô, para ser utilizado na implementação do algoritmo de planejamento de caminhos. A metodologia utiliza a ideia de que a forma mais eficiente de se navegar manipuladores móveis é utilizar mapas bidimensionais (2D) que levam em consideração os obstáculos de grande altura presentes no caminho do robô e que os mapas tridimensionais sejam usados apenas quando o robô precisa realizar uma tarefa de detecção e manipulação de objetos. Para avaliar a metodologia, o sistema desenvolvido foi simulado e embarcado na plataforma robótica. O manipulador móvel responde a comandos de voz para a execução da tarefa, além de atuar de uma forma predeterminada à presença de objetos no modelados no ambiente.

Abstract

This master thesis presents the development of a navigation system for the personal robot named MARIA in indoor environments. The navigation system enables the robot to move from one initial pose to a final pose within a previously mapped internal environment, while receiving voice instructions. The methodology involves: (i) generation of maps of the environment through the acquisition of sensory data from a Kinect sensor and a laser based proximity sensor, (ii) the position estimation robot of the upon the Monte Carlo Localization Algorithm (AMCL) (iii) path planning using wavefront and (iv) generation and execution of a collision free route. The two-dimensional map generated by the laser is used to estimate the pose of the robot in the environment. The three-dimensional map generated by using a Kinect device is used to perform a two-dimensional projection of large obstacles found in the environment so it can be used for path planning. The methodology uses the idea that the most efficient way to navigate mobile manipulators is to use two-dimensional maps (2D) that takes into account the height of the obstacles present in the path of the robot and the three-dimensional maps are used only when the robot needs to perform a task that requires of detecting and manipulating objects. To evaluate the methodology, the designed system was simulated and embedded in the robotic platform. The mobile manipulator responds to voice commands to perform the task, as well as acting in a predetermined manner to the presence of unmapped objects in the environment.

Sumário

Resumo	vii
Abstract	viii
Lista de Figuras	xi
Lista de Tabelas	xiv
Lista de Siglas	xv
1 Introdução	1
1.1 Definição do Problema	4
1.2 Objetivos	5
1.3 Metodologia	6
1.4 Organização da Dissertação	6
2 Revisão da Literatura	7
2.1 O problema da navegação em ambientes internos	7
2.1.1 Mapeamento e Localização	10
2.1.2 Planejamento, Geração e Execução de Trajetórias	14
2.1.3 Interface de Controle	16
3 Metodologia	19
3.1 Plataforma Robótica	21
3.1.1 Laser SICK LMS-291	22

3.1.2	Sensor Kinect	24
3.1.3	Pioneer 3-AT	26
3.2	Ferramentas Computacionais	27
3.2.1	ROS (Robot Operating System)	27
3.2.2	AMCL (<i>Monte Carlo Localization</i>)	29
3.2.3	RGBDSLAM	30
3.2.4	OctoMap	31
3.3	Desenvolvimento do sistema de navegação	32
3.3.1	Mapeamento	32
3.3.2	Seleção de pontos referenciais no ambiente	36
3.3.3	Reconhecimento e Confirmação de voz	37
3.3.4	Planejamento da trajetória	38
4	Resultados Experimentais	42
4.1	Resultados do mapeamento	42
4.1.1	Cenário I	42
4.1.2	Cenário II	45
4.1.3	Cenário III	47
4.2	Resultados de Localização	50
4.3	Resultados do Reconhecimento de Voz	53
4.4	Resultados do Planejamento de Caminhos	55
5	Conclusões e Trabalhos Futuros	62
	Referências	65

Lista de Figuras

1.1 Manipuladores móveis: STEWARD e PR2	3
1.2 Descrição geral da plataforma robótica MARIA com todos os componentes utilizados neste projeto.	5
2.1 Diagrama hierárquico do processo de navegação	11
2.2 Exemplo de representação de ambientes: (a) mapa topológico; (b) mapa métrico	12
2.3 Exemplo de planejamento e execução de trajetória por um robô móvel.	14
3.1 Esquema de conexão dos módulos implementados para o funcionamento do sistema de navegação proposto <i>offline</i>	20
3.2 Esquema de conexão dos módulos implementados para o funcionamento do sistema de navegação proposto <i>online</i>	21
3.3 Descrição geral da plataforma robótica MARIA com todos os componentes utilizados neste projeto.	22
3.4 Laser SICK LMS-291.	23
3.5 Configurações do laser SICK LMS-291 para este projeto	23
3.6 Descrição dos elementos que compõem o sensor Kinect	25
3.7 Campo de visão do Kinect	25
3.8 Plataforma móvel Pioneer 3-AT.	26
3.9 Exemplo do funcionamento do AMCL	29
3.10 Exemplo de extração de linhas pelo método RANSAC.	30
3.11 Esquema hierárquico de execução do RGBDSLAM.	31
3.12 Sistemas de referência atribuídos a cada elemento da plataforma robótica MARIA.	34

3.13	Representação do robô MARIA	35
3.14	Exemplo da criação do espaço de configurações	36
3.15	Conversão do sistema de referência da imagem binária para a referência do ambiente real	37
3.16	Exemplo de funcionamento do algoritmo por frentes de ondas	39
3.17	Exemplo da geração de trajetória pelo planejador até a posição desejada.	40
3.18	Execução da trajetória pelo robô	41
4.1	Cenário I, observado de quatro pontos de vista diferentes.	43
4.2	Planta baixa e representação da trajetória realizada no cenário I.	43
4.3	Mapas bidimensionais do Cenário I	44
4.4	Mapas tridimensionais do Cenário I	44
4.5	Cenário II, observado de quatro pontos de vista diferentes.	46
4.6	Planta baixa e representação da trajetória realizada no cenário II.	46
4.7	Mapas 2D encontrados para o Cenário II	47
4.8	Mapas tridimensionais do Cenário II	47
4.9	Cenário III, pontos de vista diferentes.	48
4.10	Planta baixa e representação da trajetória realizada no cenário III.	48
4.11	Mapas bidimensionais do cenário III	49
4.12	Mapa tridimensional do Cenário III, vista superior	50
4.13	Posicionamento das câmeras no cenário II.	51
4.14	Funcionamento do AMCL	51
4.15	Experimento e comparação do Posicionamento do robô dentro do cenário II.	52
4.16	Comparação entre as estimativas de posição	53
4.17	Execução do planejador de caminhos em simulação, utilizando um mapa desenhado similar a um ambiente doméstico	56
4.18	Execução do planejador de caminhos em simulação, utilizando o mapa 2D do cenário I gerado com o laser	57

4.19	Execução do planejador de caminhos em simulação, utilizando o mapa 2D projetado do cenário I	58
4.20	Execução do planejador de caminhos em simulação utilizando o mapa 2D projetado do cenário III	59
4.21	Execução do planejador de caminhos no ambiente real do cenário I	60
4.22	Experimento completo	61

Lista de Tabelas

3.1	Características técnicas do laser SICK LMS-291.	24
3.2	Características técnicas do sensor Kinect	26
4.1	Taxa de sucesso do modulo de reconhecimento de voz de acordo com o numero de tentativas.	54
4.2	Resultados Experimentais no Reconhecimento de Voz.	54

Lista de Siglas

AMCL	<i>Monte Carlo Localization;</i>
BCI	<i>Brain-Computer Interface;</i>
BSD	<i>Berkeley Software Distribution;</i>
CORO	<i>Laboratório de Computação e Robótica;</i>
CCD	<i>Charge-Coupled Device;</i>
CMOS	<i>Complementary Metal-Oxide-Semiconductor;</i>
HOGMAN	<i>Direction Cosine Matrix;</i>
ICP	<i>Iterative Closest Point;</i>
IFR	<i>International Federation of Robotics;</i>
LMS	<i>Laser Measurement Sensor;</i>
MARIA	<i>Manipulator Robot for Interaction and Assistance;</i>
LED	<i>Light Emitting Diode;</i>
PR2	<i>Personal Robot 2;</i>
PSD	<i>Position Sensitive Detector;</i>
PSR	<i>Public Service Robots;</i>
RANSAC	<i>RAndom Sampling Consensus;</i>
RFID	<i>Radio Frequency Identification;</i>
RGBDSLAM	<i>Red Green Blue - Depth Simultaneous Localization and Mapping;</i>
ROS	<i>Robot Operating System;</i>
SLAM	<i>Simultaneous Localization and Mapping;</i>
STAGE	<i>Scenario Toolkit And Generation Environment;</i>
SURF	<i>Speeded-Up Robust Features;</i>
UFMG	<i>Universidade Federal de Minas Gerais;</i>
VSTR	<i>Variable Single-Tracked Robot;</i>
WAV	<i>Windows Wave (audio format/file extension).</i>

1 *Introdução*

“Quando as pernas estão cansadas, ande com o coração.”

Paulo Coelho

A sociedade sempre almejou eficiência e precisão na realização de tarefas. Sendo assim, percebe-se que os seres humanos sempre idealizaram algum dispositivo ou máquina que os auxiliassem nas tarefas perigosas ou impossíveis de se realizar, se for realizado pelo ser humano, como por exemplo no fundo do mar ou em lugares que tenham riscos de radiação. Desse modo, o termo “robótica” começou a ser utilizado para descrever uma área tecnológica que se dedica a planejar e construir esses dispositivos que automatizam alguns processos, principalmente na área industrial [Thrun et al., 2005]. A robótica é uma área que requer o conhecimento de uma variedade de disciplinas como engenharia mecânica, engenharia elétrica, ciência da computação, dentre outras [Hata, 2010], para o desenvolvimento e a integração de técnicas e algoritmos para a criação desses dispositivos, conhecidos como robôs, deixando de lado a ideia de que os robôs fazem parte somente da ficção científica.

Com o decorrer do tempo, a tecnologia para os robôs foi evoluindo e, com isso, as suas aplicações em outras áreas nas quais o homem atua, tentando sempre tornar a vida do ser humano mais confortável e com maior qualidade [Ortiz, 2013]. Atualmente, os robôs e os sistemas robotizados atuam até mesmo nas tarefas mais simples do dia a dia. De maneira geral, os robôs de hoje executam cada vez mais as tarefas que o ser humano não quer ou não pode realizar. Dessa forma, verifica-se que os robôs são importantes para o cenário mundial, seja para fins militares, de produção industrial, domésticos, dentre outras. [Murphy, 2000] aponta que os robôs tem se desenvolvido em várias áreas, dentre elas destacam-se a agricultura, a medicina, o transporte e a educação. No entanto, existem alguns âmbitos na sociedade em que a aplicação dos robôs merecem atenção especial como: eventos nucleares, localização de naufrágios, exploração em vulcões, vigilância, limpeza, etc [Souza, 2012].

Nas áreas citadas, é possível enfatizar que existem diversas classes de robôs que se diferenciam de acordo com as suas aplicações e formas de trabalhar. Dentre essas classes, merece destaque os robôs manipuladores e os robôs móveis.

Os robôs da primeira classe são fixos e constituídos basicamente de braços manipuladores, enquanto que os robôs móveis possuem a habilidade de se deslocarem pelo ambiente, seja por terra, pelo ar ou pela água [Ortiz, 2013], o que flexibiliza a sua atuação. Os robôs móveis são multifuncionais pois não precisam estar fixados a uma célula de trabalho, podendo ser utilizados em tarefas onde não existam limites geográficos. Dentro do grupo dos robôs móveis existem os robôs voltados para trabalhar em ambientes internos, onde se destacam os robôs de serviço.

Segundo a Federação Internacional de Robótica (IFR)¹, um robô de serviço consiste em um robô que executa tarefas úteis para os seres humanos, excluindo aplicações de automação industrial. Existem robôs de serviço para uso pessoal nas áreas de entretenimento e educação, assim como também para uso doméstico, principalmente na limpeza [Pereira, 2003]. Além disso, esses robôs podem ser utilizados para uso profissional, ou seja, nas tarefas comerciais como: robôs de vigilância para detecção de incêndios ou pessoas desconhecidas; robôs para ajudar pessoas com capacidade reduzida e/ou idosas; robôs de entrega em escritórios ou hospitais; robôs de reabilitação e robôs para cirurgia em hospitais. Todas essas tarefas mencionadas anteriormente são realizadas com ajuda de uma pessoa devidamente treinada ou executada pelos robôs de forma autônoma.

No âmbito dos robôs de serviços, destacam-se aquelas pesquisas voltadas para a execução de tarefas cotidianas em ambientes internos e de entretenimento [Hornung et al., 2012, Mohamed and Capi, 2012, Kwang-Hyun et al., 2008, Gockley R., 2006, Santana et al., 2008]. Alguns exemplos dessas tarefas cotidianas podem ser observados na Figura 1.1, tendo em (a) o robô pessoal STEWARD na tarefa de ajudar pessoas em hospitais e em (b) o robô PR2 realizando uma tarefa doméstica em ambientes internos.

Para que as tarefas mencionadas anteriormente sejam realizadas pelos robôs de forma autônoma, ou seja, sem a intervenção humana, essas precisam, dentre outras coisas, de uma capacidade de navegação. Conceitualmente, o termo navegar, na robótica, consiste em guiar um robô em seu espaço de trabalho por um caminho, de forma a levá-lo de uma posição e orientação iniciais até uma posição e orientação finais [Barrera, 2010]. De maneira geral, a navegação é uma das tarefas mais complexas na problemática da locomoção dos robôs, pois deve integrar subtarefas fundamentais como: o mapeamento,

¹Disponível em: [//www.ifr.org/service-robots/](http://www.ifr.org/service-robots/)

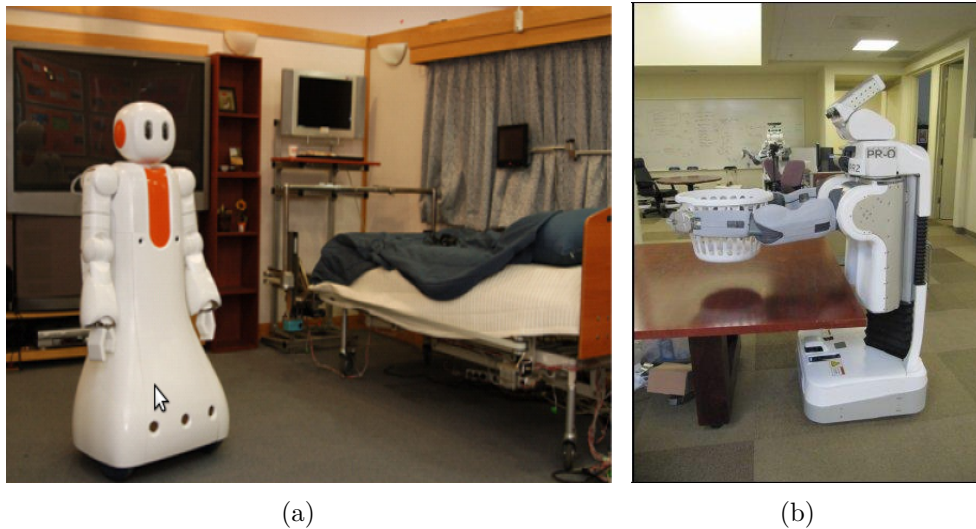


Figura 1.1: Manipuladores móveis orientados a auxiliar em tarefas: (a) Hospitalar, STEWARD [Kwang-Hyun et al., 2008] e (b) Domésticas, PR2 [Hornung et al., 2012].

que é obtido através da avaliação do cenário ou ambiente real; a localização, dada a partir da posição atual do robô em relação ao ambiente; o planejamento, que permite determinar as rotas a serem percorridas pelo robô utilizando os dados do ambiente e, a geração de uma trajetória e sua execução.

Para que a navegação consiga ser realizada com êxito e os robôs móveis alcancem sua autonomia, alguns desafios da navegação devem ser considerados:

- **Percepção**, isto é, a obtenção e a interpretação de informações sensoriais, onde os sensores são um ponto chave, pois recebem informações de posição de um objeto ou obstáculo. Além disso, entregam imprecisões nas leituras, assim como limitações na aquisição da informação devido a áreas de oclusão, o que aumenta a incerteza sobre os dados.
- **Mapeamento**, que envolve a construção de um mapa espacial a medida que se obtêm as informações sensoriais.
- **Localização**, que deve permitir de forma autônoma que o robô descubra sua posição e orientação apoiando-se na odometria², nos dados obtidos através dos sensores e no mapa do ambiente.
- **Planejamento da rota**, que deve encontrar um caminho viável sem colisões em

²A odometria é um método simples que permite estimar a posição baseando-se na informação sobre a rotação das rodas para estimar mudanças na posição ao longo do tempo.

ambientes conhecidos e/ou desconhecidos. Nos dois casos a solução para esse problema consistirá em otimizar o desempenho (tempo e custo de controle) e satisfazer as restrições (obstáculos e velocidade).

- **Execução da tarefa**, onde as ações motoras são determinadas e adaptadas às mudanças do ambiente.

Com os fatos apresentados anteriormente, apresenta-se em seguida a definição do problema tratado nesta dissertação.

1.1 Definição do Problema

No laboratório de Computação e Robótica (CORO) da Universidade Federal de Minas Gerais (UFMG) existe uma plataforma robótica denominada MARIA (*Manipulator Robot for Interaction and Assistance*). Essa plataforma é composta por uma base móvel não-holonômica Pioneer 3AT, um sensor laser SICK LMS 291, um sensor Kinect, um tablet e um braço manipulador Jaco. Com essa plataforma robótica, surgiu a necessidade de que a mesma possa locomover-se em ambientes internos, incluindo ambientes residenciais e comerciais.

Deste modo, a solução considerada nesta dissertação foi o desenvolvimento de um sistema de navegação em ambientes internos para o robô MARIA, que deve envolver a geração de mapas do ambiente, a localização, o planejamento do caminho, a geração e a execução de rotas livres de colisão.

Na Figura 1.2 é apresentada a estrutura robótica do manipulador móvel e todos os componentes anteriormente descritos utilizados neste projeto. É importante salientar que a manipulação dos objetos propriamente dita, está fora do escopo deste trabalho. No entanto, nada impede que um módulo de manipulação possa ser acoplado futuramente.

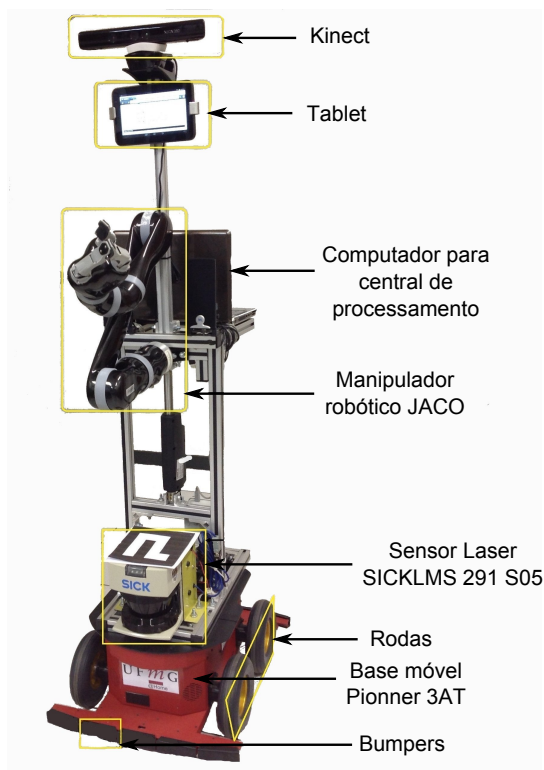


Figura 1.2: Descrição geral da plataforma robótica MARIA com todos os componentes utilizados neste projeto.

1.2 Objetivos

O objetivo principal deste trabalho é desenvolver um sistema de navegação em ambientes internos para um robô pessoal de grande altura, fazendo uso dos sensores laser e Kinect. O robô é considerado de grande altura porque tem as dimensões: 165 cm de altura, 50 cm de comprimento e 64 cm de profundidade. Como o sistema de navegação desenvolvido integra alguns subsistemas fundamentais, os objetivos específicos desta dissertação foram:

1. Integrar algoritmos que permitam a geração de mapas tanto em 3D como em 2D.
2. Implementar e integrar algoritmos para a navegação do robô móvel.
3. Integrar uma interface de voz para a comunicação do robô móvel com o usuário.
4. Realizar a simulação computacional da navegação do robô, assim como os experimentos com a plataforma robótica.

1.3 Metodologia

Neste trabalho foi desenvolvido um sistema de navegação em ambientes internos para um robô pessoal chamado MARIA. O sistema de navegação permitiu ao robô deslocar-se de uma pose inicial até uma pose final dentro de um ambiente interno previamente mapeado ao receber instruções de voz.

Uma solução viável encontrada foi integrar etapas fundamentais para resolver o problema da navegação, como a geração de mapas do ambiente mediante a aquisição dos dados sensoriais de um sensor Kinect e um laser. O mapa bidimensional gerado pelo laser é utilizado para estimar a pose do robô dentro do ambiente. O mapa tridimensional gerado fazendo uso do Kinect é utilizado para realizar uma projeção bidimensional dos obstáculos com dimensões até o tamanho do robô que encontram-se no ambiente. Essa projeção é utilizada na implementação de um algoritmo de planejamento de caminhos. O manipulador móvel responde a comandos de voz para a execução da tarefa, além de atuar de uma forma predeterminada à presença de objetos no mundo que encontram-se fora do mapa modelado. Por fim, o sistema desenvolvido foi simulado e embarcado na plataforma robótica para sua execução.

1.4 Organização da Dissertação

Esta dissertação está organizada da seguinte maneira: O Capítulo 1 tem um caráter introdutório e destina-se a orientar o leitor sobre o problema abordado na dissertação, bem como um breve detalhamento da solução abordada, os objetivos a serem alcançados e uma descrição da organização da dissertação. O Capítulo 2 é dedicado à revisão bibliográfica no campo da navegação de robôs móveis em ambientes internos e, algumas pesquisas que abordam o problema do mapeamento de ambientes. No Capítulo 3, é apresentada a metodologia utilizada para o desenvolvimento do sistema de navegação em ambientes internos, assim como também os sensores e as ferramentas computacionais utilizados para execução deste trabalho. O Capítulo 4 apresenta os resultados experimentais obtidos em cada etapa integrante do sistema de navegação, tanto no ambiente de simulação, como no ambiente real. Por fim, no Capítulo 5, são apresentadas as conclusões e as propostas para trabalhos futuros.

2 Revisão da Literatura

Neste capítulo descreve-se o problema da navegação em ambientes internos e algumas soluções encontradas na literatura. Além disso, descreve-se as etapas que compõem um sistema de navegação.

2.1 O problema da navegação em ambientes internos

Não se pode negar a grande influência que os robôs móveis têm na sociedade. Por esse motivo, grande parte desses robôs, assim como está descrito no Capítulo 1, foram desenvolvidos com o objetivo de ajudar aos seres humanos. Nesse caso, esses robôs devem estar aptos a realizar tarefas humanas com rapidez, agilidade e precisão. Para que tarefas como atividades domésticas, etc, sejam executadas, é preciso um sistema de navegação que permita a esses robôs se locomoverem pelo ambiente tanto externo como interno, como é o caso do robô RHINO que foi usado como guia de museu [Burgard et al., 1998] e do robô MINERVA que tinha o objetivo de entreter pessoas em lugares públicos [Thrun et al., 1999]. O sistema de navegação geralmente inclui a construção de um mapa do ambiente, a localização dentro de seu entorno, um planejamento de trajetória e um algoritmo de desvio de obstáculos [Souza, 2012]. Portanto, há a necessidade de se investigar e propôr tecnologias de sensoriamento do mundo em torno dos robôs, além de meios que permitam interagir de forma efetiva para adaptá-los ao ambiente em que operam, convertendo-os em máquinas capazes de tomar suas próprias decisões para alcançar um objetivo e explorar ambientes desconhecidos.

Dessa maneira, o enfoque deste trabalho é voltado para robôs móveis em ambientes internos, especificamente robôs de serviço, os quais devem mover-se em um ambiente humano sem causar nenhum tipo de acidente. Existem na literatura diversas soluções para resolver o problema da navegação em ambientes internos. Dentre as soluções existentes, merece destaque àquela proposta por Chung et al., 2004. Nesse trabalho os autores apresentam uma arquitetura integrada de navegação com o objetivo de criar um mapa estático

do ambiente utilizando dois sensores a laser colocados nas partes frontal e traseira dos robôs. Esse trabalho foi testado em três robôs de serviço, dois deles do tipo PSR's (*Public Service Robot*) que são holonômicos e um robô diferencial. As tarefas que esses realizaram foram: entrega, patrulha, guia e limpeza do chão em locais cheios de pessoas. Com a realização dessas tarefas, Chung et al., 2004 perceberam que utilizando a arquitetura dos autores os robôs lidaram com as incertezas do ambiente e as principais dificuldades que envolvem a problemática da localização, bem como as questões de segurança.

Uma proposta que trata de solucionar o problema do erro de deslocamento de um robô é apresentada em [Hayashi, 2007], onde foi desenvolvido um sistema de navegação com um controle de *auto-drive* que detectava se o robô estava se conduzindo com precisão e segurança em um quarto e um corredor. Para o cálculo do erro, utilizaram uma câmera mediante um sistema de realimentação de visão e a diferença de acelerações entre as rodas quando o robô estava avançando. A desvantagem dessa proposta é a dependência do ambiente, seja de um quarto ou um corredor para funcionar. Em [Fu et al., 2009] foi proposto um algoritmo de navegação onde o robô podia navegar de forma autônoma sem a necessidade de um mapa e, não estava limitado a funcionar em espaços predefinidos. Isso foi obtido através da aquisição da informação de uma rede de sensores sem fio colocados no ambiente interno e, a utilização de um método de triangulação para calcular a pose do robô.

Por outro lado, Jeong et al., 2009 desenvolveram um sistema de navegação autônomo para trabalhar em ambientes internos não estruturados. Esse sistema foi testado em um robô VSTR (*Variable Single-Tracked Robot*), o qual podia realizar desvio de obstáculos já que analisava um ambiente desorganizado com dois sensores: um PSD (*Position Sensitive Device*) e um LMS (*Laser Measurement Systems*), medindo a distância entre o robô e os obstáculos circundantes. Da mesma maneira, Okumura et al., 2010 criaram um sistema de navegação com um sensor laser, com o qual foi realizada a aquisição de dados do ambiente para navegar grandes distâncias em ambientes internos, tendo como pressuposto que a maior parte do interior é composto por caminhos retos com alguns cruzamentos e obstáculos ao longo das paredes laterais. Também propuseram métodos simples para o controle da direção de um robô uniclo resolvendo dessa maneira o problema da rotação do robô para uma navegação de longa distância. Outra proposta que faz uso do laser, foi a de Marder-Eppstein et al., 2010, onde o sistema é baseado na criação de um modelo de detecção em três dimensões por meio de um laser acoplado a um motor que permitia a varredura de cima para baixo permitindo adquirir maior informação do ambiente e lidar com os obstáculos.

Outra técnica para navegar foi mostrada em Zaman et al., 2011, que apresentaram um sistema de controle baseado no sistema operacional robótico ROS (do inglês, *Robot Operation System*), o qual fornece serviços para operar com robôs, nesse caso para que o robô Pioneer 3-DX faça o mapeamento, localização e navegação autônoma em ambientes internos. Foi realizado o mapeamento de ambientes dinâmicos, um simulado e dois reais com o intuito de se movimentarem de uma sala a outra, segurar e pegar um objeto de um lugar e levá-lo a outro, sempre ciente da sua própria posição.

Uma maneira bem conhecida de navegar é mostrada em Hirose and Chugo, 2011, onde esses autores utilizaram o reconhecimento de *pictographs* para ajudar na navegação de um robô de serviço em ambientes internos. O sistema de posicionamento móvel não requeria de marcadores especiais, pois estimava a posição do robô usando *pictographs* que são amplamente usados no ambiente geral dos humanos e facilmente detectados, nesse caso com uma câmera e um algoritmo para a extração de características. Uma ideia similar foi apresentada no trabalho de Paredes, 2014, que utilizou um sistema de sinalização *signage* baseado na tecnologia de identificação radio frequência (RFID), que foi desenvolvido para orientar e dar informações importantes a um robô social quando encontrava-se em um lugar desconhecido. O sistema foi testado em um ambiente real e foi provado com êxito no robô autônomo e social chamado Maggie. Assim também em [Isozaki and Chugo, 2011], o sistema de navegação era baseado no reconhecimento de marcadores LED e filtros infravermelhos, com os quais se obtinha uma alta precisão na localização do robô dada pelas câmeras colocadas no teto do ambiente interior.

Dentro do contexto da navegação de robôs, Correa and Sciotti, 2012 realizaram a navegação de um robô móvel em ambientes internos utilizando o sensor Kinect e desenvolveram um sistema de percepção que permitiu uma navegação autônoma reativa, onde o robô se deslocava evitando obstáculos usando o sensor. O sistema de navegação utilizava redes neurais artificiais para reconhecer diferentes configurações do ambiente e conseguir caminhar para frente, à esquerda, à direita e nas interseções. As redes neurais foram treinadas usando dados capturados pelo sensor Kinect. Similarmente, Abdelkrim and Issam, 2014 implementaram três controladores de lógica fuzzy, um deles para que o robô consiga ir até o objetivo, outro para evitar obstáculos e o terceiro para construir o mapa do ambiente, tudo isto para a navegação de um robô móvel Pioneer 3-AT em ambientes desconhecidos usando sensor Kinect. Além disso, a proposta mostra que o robô pode trabalhar em um ambiente escuro o que é uma vantagem em sistemas de vigilância.

Em [Hornung et al., 2012], foi realizada uma navegação livre de colisões em ambientes

não estruturados para um manipulador móvel. Essa abordagem resolveu problemas de planejamento de trajetórias em 3D usando uma representação eficiente do ambiente em octrees. Nessa abordagem utilizou-se uma combinação de representações de múltiplas camadas 2D e 3D permitindo o planejamento de forma eficiente e a execução de trajetórias em tempo quase real. Além disso, os autores apresentaram resultados experimentais através da manipulação móvel dos dois braços no robô PR2, no qual transportava grandes objetos em um ambiente desorganizado, locomovendo-se entre passagens estreitas desse mesmo ambiente. Igualmente, em [Gross et al., 2012] foi desenvolvido um sistema de navegação autônomo para um robô companheiro de assistência social para idosos. Esse sistema incluía um mapa detalhado do apartamento onde morava a pessoa a ser cuidada, uma robustez a erros de localização para auto localizar-se dentro do apartamento, uma forma confiável de evitar os obstáculos conhecidos e desconhecidos, uma forma eficiente para passar por portas estreitas e lacunas entre móveis, dentre outras.

Em [Seo and Kim, 2013] desenvolveram um sistema de navegação autônoma para aplicações de um robô de serviço em um ambiente interno, tendo nesse trabalho uma proposta completa, pois se dedica não só à criação de um mapa do ambiente, mas também à localização do robô e ao planejamento da trajetória. O sistema mostrou bom desempenho para aplicação como robô de serviço, embora tenha sido testado somente em um ambiente de simulação.

Nos trabalhos descritos anteriormente, os autores se empenharam em resolver o problema da navegação em ambientes internos utilizando vários processos conhecidos na literatura. Esses processos são: o mapeamento, a localização, o planejamento dos caminhos e a geração e execução de trajetórias (veja a Figura 2.1). Portanto, é importante conhecer os tipos de técnicas que utilizam cada um dos processos sumarizados na Figura 2.1, como será descrito a seguir.

2.1.1 Mapeamento e Localização

O processo de mapeamento envolve a captura das características do ambiente por meio de sensores, permitindo a localização do robô e a aplicação de técnicas baseadas em algoritmos de inteligência computacional [Ortiz, 2013]. Por outro lado, a localização consiste em determinar a pose do robô (x, y, θ) em relação ao ambiente, a partir de dados obtidos pelos sensores [Thrun et al., 2005]. A localização e o mapeamento são processos dependentes, sendo que para localizar um robô necessita-se que um mapa exista, enquanto que para construir um mapa é necessária a posição a ser estimada em relação a um ponto

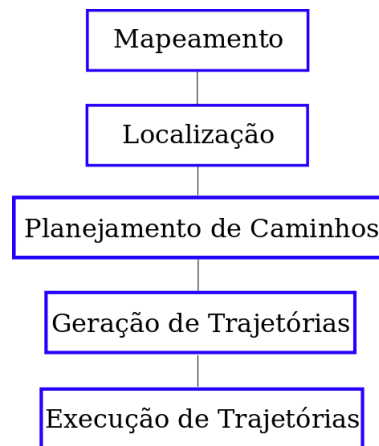


Figura 2.1: Diagrama hierárquico geral dos processos realizados em um sistema navegação.

de referência.

As primeiras investigações na linha de mapeamento de ambientes surgiram no final da década de 80. As propostas apresentadas pelos pesquisadores mostraram diferentes formas de representação dos ambientes. Dentre elas, destacam-se a representação poligonal [Vincent, 1987], por meio de grafos [Merat and Wu, 1987], por meio de esferas 3D [Goldstein et al., 1987], por meio de modelos geométricos hierárquicos [Kriegman et al., 1987], baseada na topologia do ambiente [Kuipers B., 1988], na grade de ocupação [Moravec and Elfes, 1985], dentre outras. A partir dessas representações, em [Thrun, 2002] e em [Murphy, 2000] é possível observar a classificação das representações em topológicas e métricas (Figura 2.2) [Hata, 2010, Ortiz, 2013].

A representação topológica engloba uma representação em forma de grafos, onde as arestas conectam pontos de referência, representando a existência de um caminho entre eles. Esses pontos, ou nós, por sua vez indicam locais específicos como portas, ou cruzamentos entre corredores, que podem ser identificados pelo robô. A representação métrica armazena as propriedades geométricas do ambiente e geralmente reproduz o ambiente por meio de uma grade de ocupação, sendo essa grade um plano composto por células geometricamente iguais. Essas células representam áreas específicas do espaço e também armazenam a informação sobre seu estado, como livre ou ocupado [Prado, 2013].

Uma vantagem da representação em grade de ocupação é a facilidade do robô se localizar devido a grande informação armazenada do ambiente. Entretanto, o mapa topológico garante um planejamento de trajetória muito mais rápido levando em consideração a menor quantidade de dados armazenados.

Os sensores mais utilizados para realizar o mapeamento são os lasers, sonares e câ-

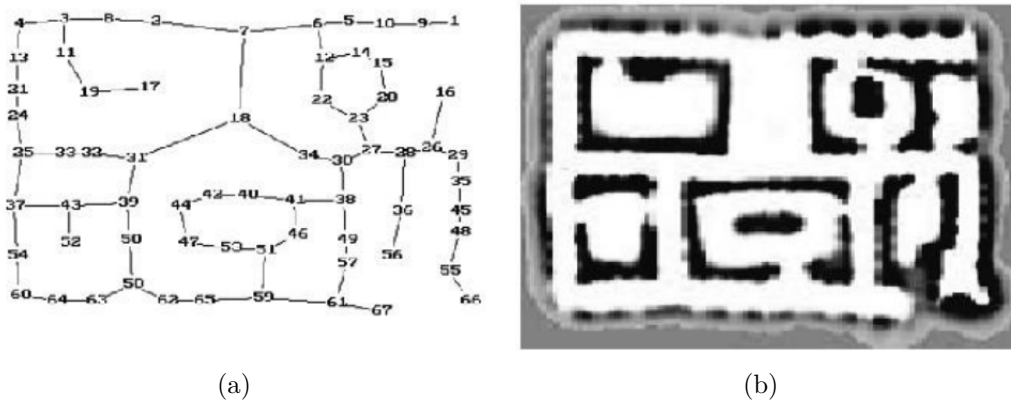


Figura 2.2: Exemplo de representação de ambientes: (a) mapa topológico formado por arestas e nós; (b) mapa métrico formado por uma grade de ocupação. Retirado de Durrant-Whyte et al., 2003.

meras. Os sonares são conhecidos e populares por seu baixo custo, mas possuem baixa precisão e exatidão. O laser, por outro lado, tem maior precisão, exatidão, e alcance, fazendo dele um dos sensores mais usados na atualidade, pois fornece também informações de fácil interpretação e que não necessitam de um processamento complexo. Temos também as câmeras de profundidade como, por exemplo, o Kinect, que são novos sistemas de sensoriamento que capturam imagens RGB junto com informações de profundidade para cada pixel.

Algumas das propostas descritas anteriormente para resolver o problema da navegação em ambientes internos fizeram uso das técnicas de representação do ambiente mencionadas. [Chung et al., 2004], por exemplo, utilizaram mapas topológicos para realizar o planejamento global, um mapa em grade de ocupação para conseguir localizar o robô com um laser, bem como dois mapas geométricos atualizáveis que serviam para o planejamento local e de informação ao usuário. Ruhnke et al., 2011 usaram o sensor laser para realizar o mapeamento de ambientes mas com o uso de algoritmos de otimização para melhorar a acurácia da pose do robô. Correa and Sciotti, 2012, também utilizaram a mesma estrutura topológica para representar a configuração do ambiente, onde os corredores (caminho à frente) são as arestas e os nós (caminho à esquerda e intersecção, por exemplo) são os vértices.

Marks et al., 2009 desenvolveram um sistema para realizar um mapeamento em grade de ocupação para ambientes não estruturados, utilizando um robô com um sistema de visão estéreo. Seo and Kim, 2013 empregaram um laser e um algoritmo de construção de mapa em grade de ocupação para representar o ambiente, onde levavam em consideração

que essa representação pode ser mais facilmente manipulável, além de permitir a fácil aplicação de um algoritmo de localização Monte Carlo para estimar a pose do robô. Seguindo as ideias propostas, Souza, 2012 apresentou um novo método de mapeamento para robô que utiliza um mapa 2.5D com representação em grade de ocupação-elevação, o que possibilitou ao robô um prévio conhecimento sobre a ocupação de um determinado local a partir da captura dos dados de câmeras estéreo.

Com frequência encontra-se na literatura autores que propõem o uso em conjunto de distintos sensores para alcançar um mapeamento mais completo e eficiente, bem como navegar com maior segurança. Entre os primeiros trabalhos nesse âmbito destacam-se o sistema de mapeamento proposto por Ahn et al., 2007, no qual utilizou-se um conjunto de sonares e um sistema de visão estéreo para construir um mapa em grade de ocupação. Outras propostas descritas na literatura levam em consideração a mesma ideia dessas técnicas, mas ao invés de criar uma representação do ambiente em 2D, criam uma em 3D, como Wen et al., 2014, que construíram um sistema de mapeamento 3D em ambientes internos com fusão de dados de um laser e uma câmera RGB-D que combina a representação 2D em grade de ocupação com a representação 3D a partir de uma estimativa de filtro de partículas. Gallegos and Rives, 2010, utilizaram um robô com uma câmera omnidirecional e um laser para realizar o mapeamento 3D de ambientes internos, onde o sensor laser foi utilizado para fornecer dados mais exatos sobre a profundidade de objetos e obstáculos. Ortiz, 2013, concentrou-se em um modelo híbrido a partir da utilização de sensores laser, câmeras estéreo e sensores Kinect, que permitiram a navegação em ambientes internos realizando tarefas de monitoramento e vigilância.

Algumas propostas que fazem uso somente do sensor laser planar trazem um modelo de detecção baseado em três dimensões, pois esses sensores são usualmente acoplados a um motor que permite a realização de movimentos de cima para baixo, permitindo uma varredura de forma a obter maiores informações do ambiente [Marder-Eppstein et al., 2010]. Além disso, outros trabalhos compreendem a utilização de técnicas que lidam com obstáculos em 3D [Thrun et al., 1999, Surmann et al., 2003, Hornung et al., 2012, Ortiz, 2013], ou abordagens que fazem uso de grade de ocupação multi-volume para representação do ambiente tridimensional [Dryanovski and Morris, 2010]. Nesta dissertação também faz-se uso da grade de ocupação para uma representação bidimensional do ambiente utilizando o sensor laser e uma representação 3D baseada em octrees usando um Kinect [Albán-Peñafiel and Pereira, 2014].

2.1.2 Planejamento, Geração e Execução de Trajetórias

Várias abordagens foram apresentadas no início deste capítulo para solucionar o problema da navegação em ambientes internos. Algumas delas realizando um planejamento e execução de trajetória para ir de uma pose inicial até uma pose final, tal que, o robô não colida com nenhum obstáculo do ambiente e que os movimentos planejados sejam consistentes com as restrições físicas do robô [Jahanbin, 1988].

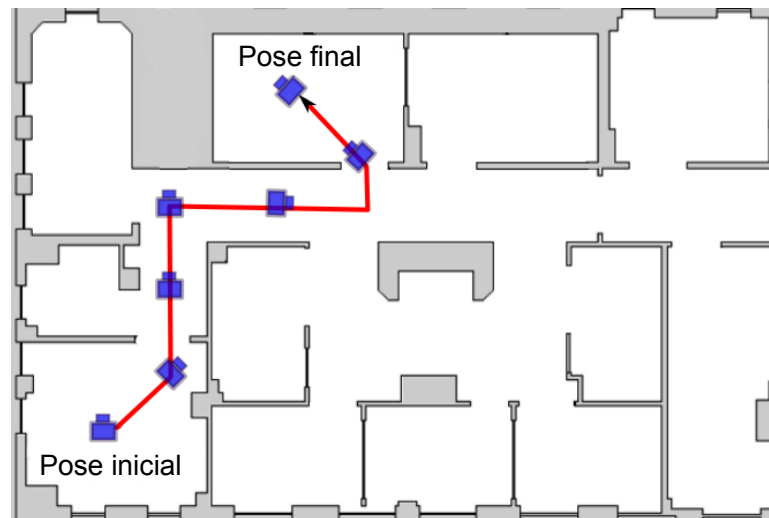


Figura 2.3: Exemplo de planejamento e execução de trajetória por um robô móvel.

A Figura 2.3 mostra um exemplo típico de planejamento de caminho, em que um robô diferencial sai da configuração inicial na parte inferior e chega na configuração final na parte superior, passando por estreitas passagens do espaço de trabalho.

Existem vários métodos apresentados na literatura para planejamento de caminhos para robôs móveis terrestres [Russell and Norvig, 2003]. Alguns dos principais algoritmos são: o mapa de rotas probabilístico [Kavraki et al., 1996], a decomposição em células [Ottoni, 2000], campos potenciais [Eder, 2003] e dentro deste, utilizado neste trabalho, o método do frentes de ondas [Choset, 2005, LaValle, 2006].

O mapa de rotas probabilístico (Roadmap), visa construir um mapa topológico que represente a conectividade do espaço livre e que reduz as informações do ambiente a um grafo com todos os possíveis caminhos [Adorno, 2008]. Uma vez que o Roadmap é construído, o planejamento de trajetórias se reduz a conectar as posições iniciais e finais do robô ao Roadmap e buscar um caminho entre esses dois pontos usando métodos de busca em grafos. Geralmente, os algoritmos que usam o método de mapa de rotas probabilístico são rápidos e de simples implementação, mas eles não fornecem uma boa representação

das informações do ambiente [de Souza, 2002]. Por outro lado, os métodos que fazem decomposição em células, decompõem o espaço livre de um ambiente representado em um mapa em uma coleção de regiões que não se sobrepõem. Utilizam um grafo de conectividade [Ottoni, 2000] para armazenar a informação de todas as células adjacentes de forma que um caminho entre duas configurações em células pode ser facilmente encontrado [Adorno, 2008].

Existem também os algoritmos que utilizam funções de potencial que consistem em calcular a direção e sentido da trajetória de acordo com a direção e o sentido da resultante das forças que se aplicam ao robô em cada instante da navegação [de Souza, 2002]. Assim, os obstáculos provocam uma força de repulsão, enquanto a configuração de destino induz uma força de atração ao robô [Adorno, 2008]. Esse método é muito utilizado para ambientes dinâmicos, visto que analisa de forma mais rápida o espaço de trabalho para evitar futuras colisões com obstáculos dinâmicos. Por outro lado, esse método tem algumas desvantagens como, por exemplo, os mínimos locais. Algumas propostas fizeram uso desses métodos, entre elas encontra-se por exemplo [Barraquand and Latombe, 1991], que utilizaram campos potenciais artificiais para robôs com vários graus de liberdade permitindo sair de mínimos locais através de passeios aleatórios. Enquanto que em [Pimenta et al., 2006], utilizou-se campos potenciais eletrostáticos artificiais resultando em apenas um mínimo local localizado no destino.

Uma abordagem relativamente simples é o planejador por frentes de ondas, também conhecido como *Wavefront Planner*. Ele representa uma lista de nós que vão se expandindo de forma ascendente no espaço livre de toda a representação em grade de ocupação do ambiente. Ele começa expandindo o tamanho de todos os obstáculos para criar o espaço de configuração e evitar colisões. Logo depois, rotula como 1 às células ocupadas pelos obstáculos e 2 a célula que será o objetivo a alcançar. A partir dessa célula, a frente de onda vai se expandindo em todas as células livres no espaço [de Souza, 2002]. Uma vez conhecida a posição do robô em uma célula da representação, se encontra a solução seguindo a sequência decrescente até a célula de rótulo 2, que corresponde à pose desejada final do robô.

Entre as características do algoritmo por frentes de ondas estão o fato que, dado uma resolução arbitrária ele retorna uma solução (se existir) em tempo finito ou uma informação de ausência de solução se não existir. Além disso, possui um único ponto de mínimo, localizado no ponto de destino. Contudo, para uma representação do ambiente de tamanho maior, esse algoritmo pode se tornar computacionalmente impraticável [Adorno,

2008].

Kwon et al., 2006 propuseram um algoritmo de planejamento de caminho local baseado no método de frentes de ondas para um robô móvel com um sistema de visão. Baseado na geometria projetiva do sistema de visão e o método de frentes de ondas conseguiu-se movimentar o robô em um ambiente determinado. Esta dissertação também faz uso do algoritmo de frentes de ondas para o planejamento de caminhos, uma vez que é um algoritmo simples e fácil de implementar. Além disso, esse algoritmo funciona para qualquer resolução arbitrária escolhida na discretização dos mapas, pois existe somente um ponto mínimo localizado no ponto objetivo.

2.1.3 Interface de Controle

Esta subseção está dedicada a mostrar os diferentes tipos de interfaces com o usuário que foram utilizados como meio para a execução do sistema de controle de robôs. No caso dos robôs autônomos, uma vez criado um sistema de navegação é necessário executar uma interface que permitirá ativar sua navegação autônoma. Portanto, apresentam-se vários tipos de trabalhos que utilizam uma interface para o controle de robôs. Entre os mais destacados está [Zhixiao Yang, 2006], onde foi proposta uma interface gráfica para a navegação de um robô de resgate remotamente controlado. O sistema é baseado em imagens e, um computador remoto foi usado para exibir as imagens recolhidas dos desastres, sendo essa a única interface do aplicativo no seu tempo de execução. Campos funcionais foram especificados para realizar as operações de resgate e, a navegação foi realizada clicando e escolhendo diversos campos funcionais. Assim também em [Sato et al., 2008], foi proposta uma interface de navegação de robôs usando uma caneta de toque e um mapa do ambiente local. O mapa do ambiente era exibido para o operador e ele selecionava o ponto objetivo no mapa para a execução da navegação do robô. A trajetória e a velocidade desejadas foram calculadas de acordo com os comandos dados pelo operador e o robô começava a mover-se automaticamente para a posição selecionada, fazendo uso das informações sensoriais.

Em [Shim et al., 2010] foi apresentado um sistema de controle inteligente para robôs com reconhecimento de voz, o qual podia reconhecer a voz de adultos e crianças de forma robusta em ambientes ruidosos. A voz era capturada usando um microfone sem fio. Basicamente, se fez a integração de um reconhecimento de voz robusto e um sistema de navegação autônomo capaz de realizar navegação em ambientes desconhecidos. Uma ideia similar é mostrada em [Lv et al., 2008], que realizaram o controle de um robô móvel

mediante comandos de voz. Esse sistema simples mostrava a capacidade de aplicar técnicas de reconhecimento de voz para aplicações de controle. O robô conseguia compreender comandos de controle falados por um operador de uma forma natural, e por tanto executava a ação correspondente, ou seja, o operador somente precisava falar uma instrução para fazer o robô realizar a tarefa. O robô podia ser facilmente controlado por qualquer operador humano que tivesse um conhecimento limitado sobre robôs ou computadores.

Igualmente, em [Hibino et al., 2010] é possível notar que foi desenvolvido um sistema de navegação para robôs móveis, utilizando o reconhecimento de indicações de um humano. Uma pessoa podia fazer com que o robô navegasse de forma intuitiva até um lugar simplesmente apontando a posição desejada com a mão, enquanto que o robô móvel reconhecia as indicações com uma câmera tipo olho de peixe colocado no topo dele e navegava para o local dos pontos indicados fazendo uso de um simples controlador com *feedback* visual.

De outra forma, [Sato et al., 2011] apresentaram uma interface de navegação para vários robôs móveis autônomos, introduzindo um conceito de agrupamento. Essa interface oferece interação com vários robôs, na qual o sistema exibe o mapa do ambiente do entorno dos robôs no monitor e o operador pode graficamente colocar uma posição de destino ou uma trajetória desejada para um conjunto de vários robôs autônomos.

Yu and Nawroj, 2012 desenvolveram uma interface cérebro-computador (BCI) para navegar um robô por meio de atividades cerebrais. Um operador interagia com comandos de navegação exibidos em uma tela de computador e enviando os comandos ao robô fazendo uso das informações adquiridas do cérebro. Sinais de eletroencefalograma foram adquiridos por um sistema de aquisição de dados e analisados para identificar a resposta do operador determinando assim o comando de navegação. Esse comando é então transmitido através de uma rede local para guiar o robô. Lichtenstern et al., 2013, criaram uma interface intuitiva para navegação remota de um sistema múltiplos robôs projetada para facilitar a interação entre um único usuário e um grande número de robôs equipados com câmeras. Com essa interface, o usuário, usando um display acoplado à sua cabeça, é capaz de ver as transmissões de vídeo em tempo real provenientes das câmeras dos robôs e, se desejar, selecionar um robô para controlá-lo remotamente mediante o movimento da sua cabeça.

Nesta dissertação, a navegação do robô foi realizada a partir do reconhecimento de instruções de voz, nas quais mencionam-se o lugar para onde o robô deve deslocar-se. No próximo capítulo será apresentada a metodologia utilizada para o desenvolvimento do

sistema de navegação em ambientes internos.

3 *Metodologia*

Neste capítulo são descritas as ferramentas, os procedimentos e os métodos utilizados para o desenvolvimento de um sistema de navegação em ambientes internos para o robô móvel de grande altura MARIA (Figura 3.3). Como visto no Capítulo 2, existem várias técnicas empregadas para realizar um sistema de navegação. Neste trabalho é proposta uma metodologia diferente para a resolução deste problema.

O sistema de navegação permite ao robô deslocar-se desde uma pose inicial até uma pose final, dentro de um ambiente interno previamente mapeado ao receber instruções de voz. Para conseguir que o robô navegue dentro do ambiente, foi realizado um mapeamento do ambiente tanto bidimensional como tridimensional. Para controlar o movimento do robô pelo cenário utilizou-se um controle manual fazendo uso do teclado. Enquanto o robô se locomovia realizou-se simultaneamente a representação do ambiente em um mapa bidimensional, fazendo uso de uma grade de ocupação e do sensor laser e, uma representação em grade de ocupação tridimensional, baseada em octrees utilizando o sensor Kinect.

A partir dos mapas completos obtidos do ambiente, realiza-se uma filtragem do mapa tridimensional ao longo do eixo de coordenadas z , excluindo dessa forma o teto e o chão, transformando os pontos 2D em uma imagem e gerando uma projeção bidimensional em grade de ocupação que considera os obstáculos até a altura do robô, como ilustra-se no diagrama de fluxo da Figura 3.1.

Fazendo uso do mapa bidimensional projetado, foram selecionados pontos referências do ambiente com o objetivo de obter uma base de dados desses pontos e seus nomes associados, considerando-os como pontos objetivos ou referências que o robô deverá atingir segundo ordens obtidas na etapa do reconhecimento de voz. O módulo de reconhecimento de voz permite identificar a instrução que o usuário dá ao robô para que atinga um lugar. O lugar identificado é procurado na base de dados que é salva no momento de selecionar os pontos referenciais e com ela é encontrada as coordenadas deste lugar.

Com as coordenadas do ponto identificadas é realizado o planejamento da trajetória pelo método de frentes de onda permitindo gerar uma trajetória livre de colisões até atingir o ponto objetivo desejado pelo usuário. Todo esse processo é repetido iterativamente como é mostrado na Figura 3.2.

Basicamente, o sistema de navegação proposto é a integração de todas as etapas mencionadas anteriormente que são mostradas no esquema da Figura 3.1 que funciona *offline* e na Figura 3.2 executada *online*. Cada módulo observado nas Figuras serão detalhados mais detalhadamente nas próximas sub-seções.

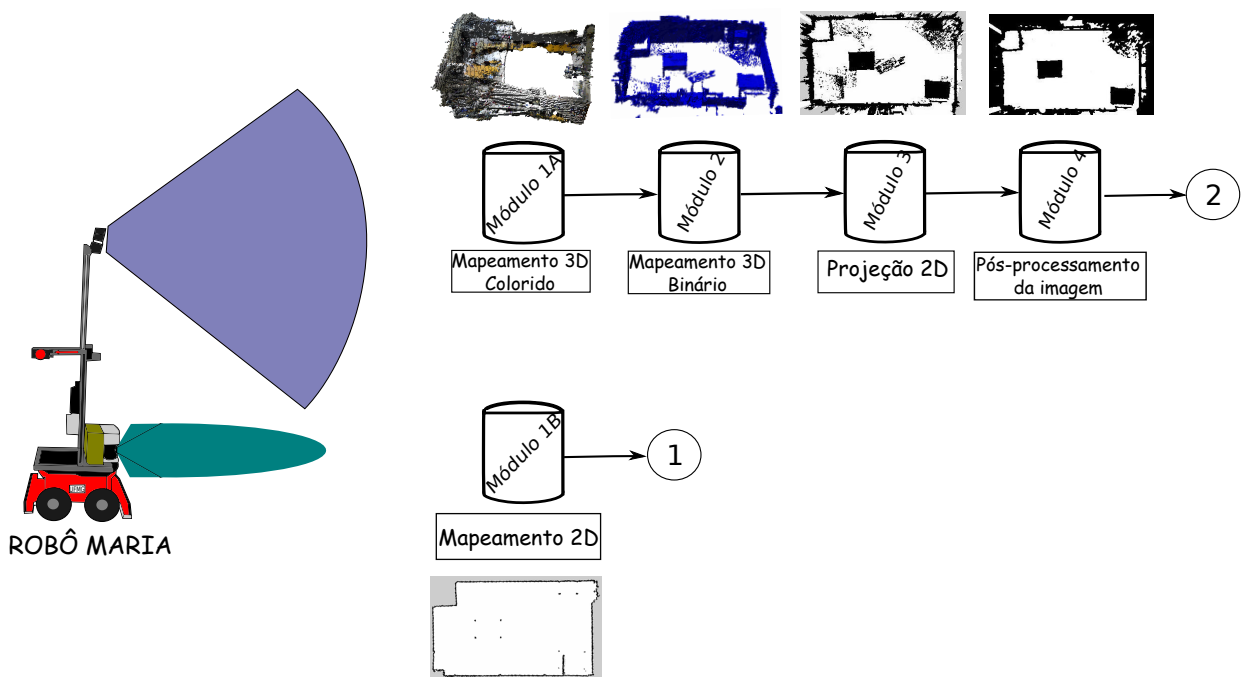


Figura 3.1: Esquema de conexão dos módulos implementados para o funcionamento do sistema de navegação *offline*. Módulo 1A e 1B realizam simultaneamente o mapeamento 3D e 2D. O Módulo 2 converte o mapa 3D colorido em um mapa baseado em octrees e realiza o filtro do mapa. O módulo 3 realiza a projeção bidimensional a partir do módulo 2. O módulo 4 realiza o processamento da imagem gerado a partir módulo 3.

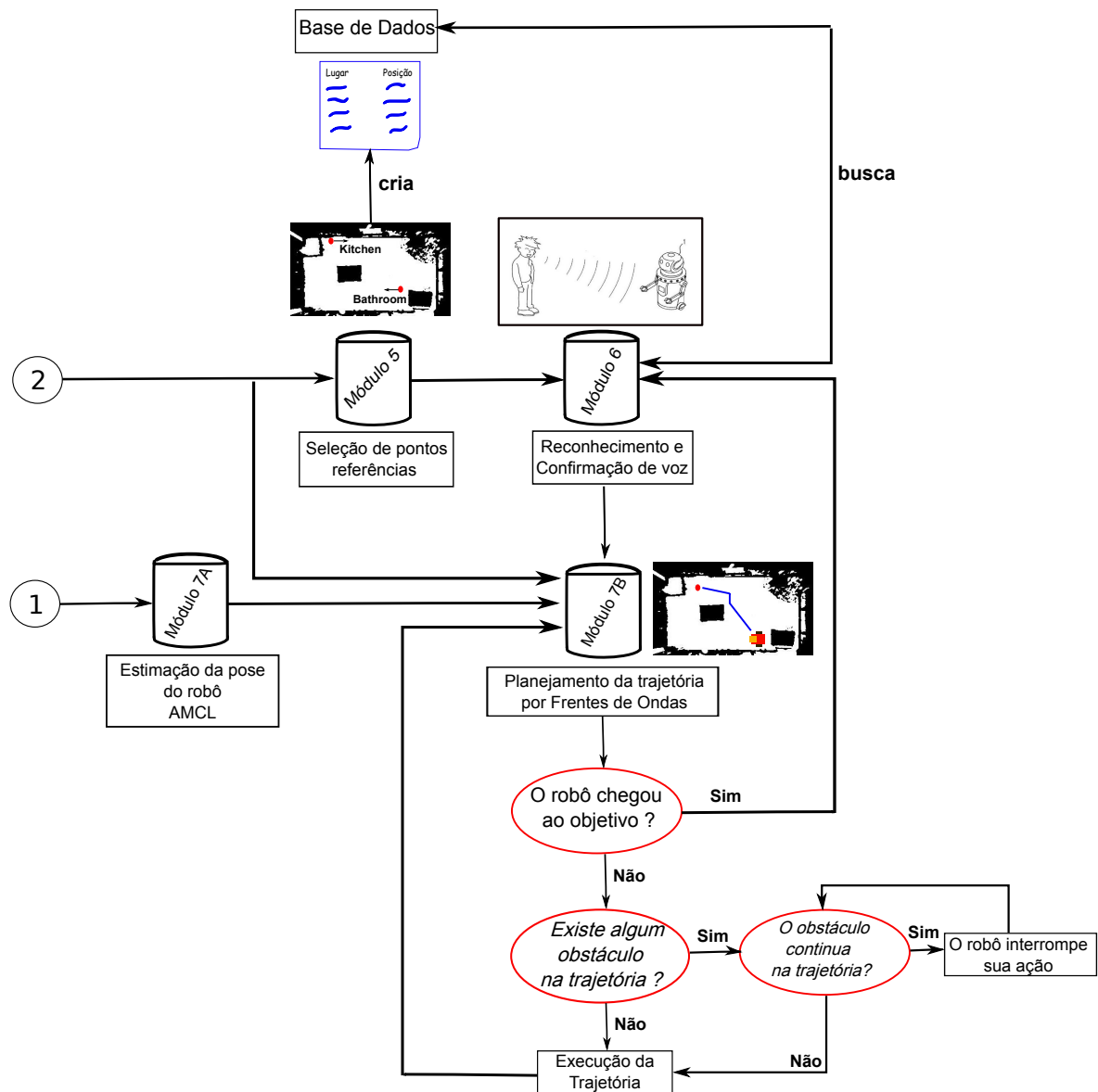


Figura 3.2: Esquema de conexão dos módulos implementados para o funcionamento do sistema de navegação *online*. Módulo 5 recebe informações do Módulo 4 e permite a seleção de pontos referências do ambiente pelo usuário. O Módulo 6 permite o reconhecimento e verificação das instruções de voz. O Módulo 7A E 7B trabalham simultaneamente para a estimação da pose do robô e o planejamento da trajetória, enquanto o robô se locomove pelo ambiente.

3.1 Plataforma Robótica

Várias plataformas robóticas já foram desenvolvidas e são usadas na pesquisa sobre a interação humano-robô. Essas plataformas geralmente são compostas por sensores e atuadores, que em conjunto são denominados o *hardware* do sistema. O *hardware* permite que seja feita a captura de dados do próprio robô e do ambiente no qual ele está inserido,

bem como modificá-lo e/ou efetuar o seu deslocamento dentro dele [Ortiz, 2013].

A seguir é apresentada a plataforma robótica utilizada neste projeto, bem como os elementos que a compõem (Figura 3.3). Além disso, descreve-se brevemente as ferramentas computacionais que permitiram desenvolver o sistema de navegação proposto. É importante destacar que a manipulação dos objetos está fora do escopo deste trabalho. No entanto, nada impede que um módulo de manipulação possa ser acoplado futuramente.

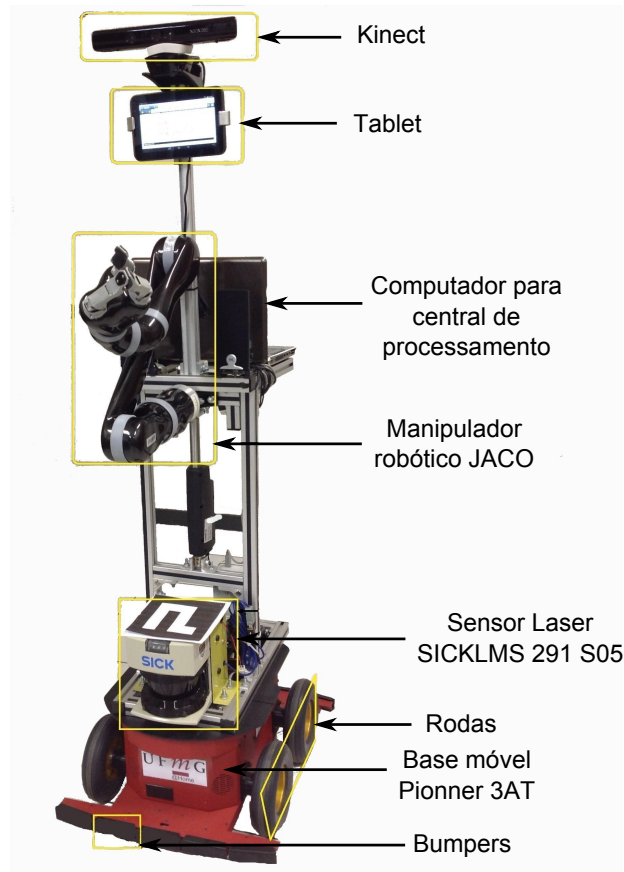


Figura 3.3: Descrição geral da plataforma robótica MARIA com todos os componentes utilizados neste projeto.

3.1.1 Laser SICK LMS-291

O laser SICK LMS-291 (*Laser Measurement Sensor*) é um sensor de medição de distância que permite coletar dados do ambiente por meio de repetidas emissões de luz (Figura 3.4). Utiliza um raio laser infravermelho de classe I, inofensivo para o olho humano, mesmo em tempos de exposição prolongados.

O sensor oferece medições de distância em uma faixa de 180° com um alcance configurável de até 80 metros. Neste trabalho foi utilizado um alcance de 8 metros, resolução



Figura 3.4: Laser SICK LMS-291.

de um grau e o período de amostragem é de 13ms (Figura 3.5).

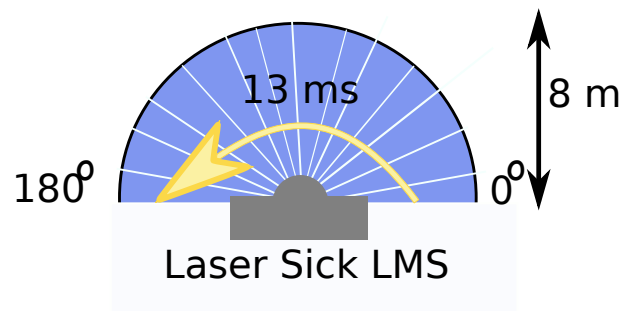


Figura 3.5: Configurações da varredura, alcance e período de amostragem do laser SICK LMS-291 para este projeto.

A tabela abaixo apresenta algumas características importantes desse dispositivo.

Tabela 3.1: Características técnicas do laser SICK LMS-291.

Características	SICK LMS-291S05
Comprimento de onda	Infravermelho (905 nm)
Classe Laser	1 (EN/IEC 60825-1)(<i>eye-safe</i>)
Ângulo de Abertura	180°
Frequência de Exploração(max)	75 Hz
Resolução Angular	1°
Máximo Alcance	80 m
Alcance Configurado	8 m
Tempo de Resposta	13 ms
Resolução	10 mm
Erro Sistemático	± 35mm
Erro Estatístico	± 10mm
Interface de Dados	RS422
Velocidade de Transferência	38,4 kBaud
Peso	4,5 kg
Dimensões	156 mm x 155 mm x 210 mm

3.1.2 Sensor Kinect

O sensor Kinect é uma câmera do tipo RGB-D utilizada como controlador para jogos. Esse sensor foi desenvolvido pela Microsoft para ser utilizado no vídeo game Xbox 360 e em PC's. O módulo Kinect tem cerca de 23 cm de comprimento como visto na Figura 3.6 e, é composto principalmente dos seguintes recursos:

- Uma câmera RGB (Resolução 640x480, 30fps VGA), para codificar vídeo e imagens digitais.
- Um sensor de profundidade 3D que permite que o acessório escaneie o ambiente a sua volta em três dimensões e, é composto de:
 - Um projetor de luz infravermelha.
 - Uma câmera infravermelha composta por um sensor CMOS monocromático de luz infravermelha (Resolução 640x480, 30fps).
- Quatro Microfones (16bit taxa de amostragem: 16KHz), utilizados para reconhecimento de voz.

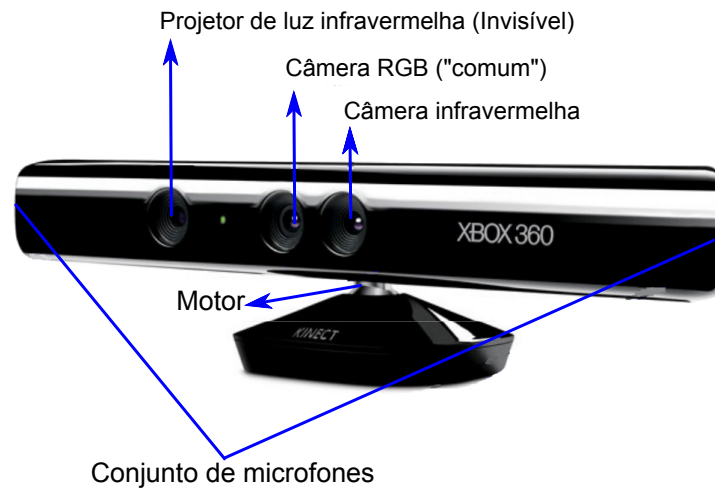


Figura 3.6: Descrição dos elementos que compõem o sensor Kinect. Fonte: <http://nti.ceavi.udesc.br/>.

- Um motor para inclinar o dispositivo automaticamente para cima ou para baixo.

O funcionamento desse dispositivo está restrito ao alcance da visão do sensor de profundidade. Contudo, existem faixas nas quais o sensor não consegue registrar as imagens em 3D adequadamente. Como se observa na Figura 3.7, o sensor de profundidade não consegue registrar as imagens no intervalo de 0 a 0,8 m e, mais de 4 m.

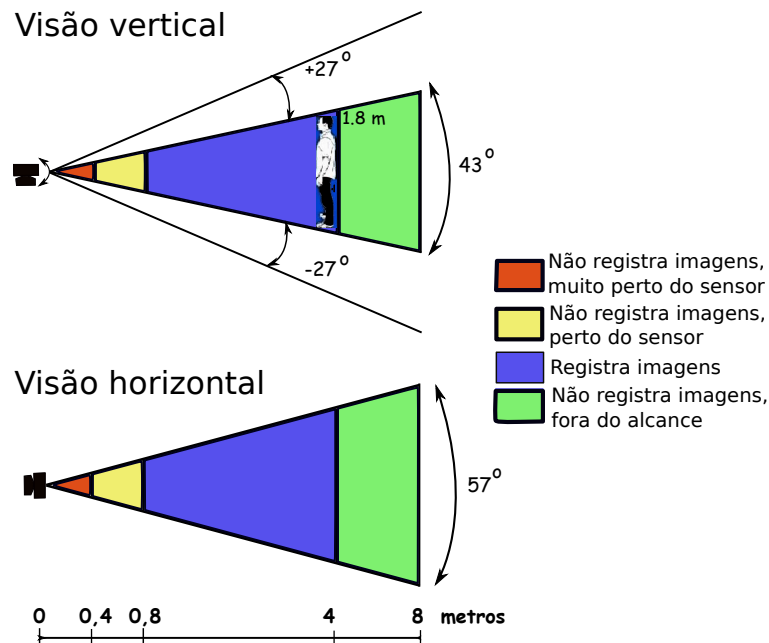


Figura 3.7: Campo de visão horizontal e vertical do sensor de profundidade do Kinect.

A seguir é apresentada uma tabela com as principais características do equipamento.

Tabela 3.2: Características técnicas do sensor Kinect.

Kinect	Especificações
Ângulo de Visão	43° vertical por 57° horizontal
Faixa de Inclinação Vertical	$\pm 27^\circ$
Taxa de <i>Frames</i>	30 fps (frames por segundo)
Formato de Áudio	16-kHz, 24 bits PCM (<i>Mono Pulse Code Modulation</i>)

3.1.3 Pioneer 3-AT

O robô Pioneer 3-AT (Figura 3.8) é uma plataforma móvel não-holonômica. As dimensões desse são: 0,625m de comprimento e 0,49m de largura.



Figura 3.8: Plataforma móvel Pioneer 3-AT.

Os elementos que compõem a base móvel não-holonômica são os seguintes:

- **Painel de controle**, consiste em um painel de acesso ao microcontrolador do robô e é composto por vários botões de controle, LEDs indicadores de estado e uma porta serial RS-232 com um conector de 9 pinos.
- **Motores**, o robô está dotado de 4 motores de corrente contínua reversíveis e que permitem ao mesmo mover-se com uma velocidade de até 1 m/s.
- **Baterias**, o robô é alimentado por três baterias de 12V e 9Ah cada. Assim, o robô dispõe um total de 252 W/h.

A plataforma Pioneer 3-AT, pode receber comandos de velocidade e enviar o estado dos *bumpers*, das baterias e dos motores, além da pose do robô. Para estimação da pose do robô são utilizados os *encoders* e um girômetro.

Na próxima seção deste capítulo serão apresentadas as ferramentas computacionais do sistema robótico supracitado.

3.2 Ferramentas Computacionais

Para que um robô móvel seja funcional e interaja de forma autônoma com o ambiente, é necessário a implementação de várias técnicas que possam de fato prover essa autonomia. Por tal motivo, foram integrados vários componentes de software na plataforma robótica MARIA. O objetivo é compor um sistema embarcado que faça uso de vários recursos disponíveis na literatura como pacotes de *software* e bibliotecas de acesso a recursos de *hardware* (drivers). Os recursos de *software* utilizados para compor o sistema desenvolvido serão descritos de maneira pontual nas seções subsequentes.

3.2.1 ROS (Robot Operating System)

O ROS³ é um meta sistema operacional que fornece bibliotecas e ferramentas para ajudar aos desenvolvedores de *software* a criarem aplicações para robôs. Ele fornece abstrações para diversos recursos, tais como: controle de acesso a dispositivos, implementação de funcionalidades genéricas de controle, trocas de mensagens entre processos, sincronismo e gerenciamento de pacotes, etc. O ROS é livre e de código aberto tendo sua distribuição regida pela licença BSD (*Berkeley Software Distribution*).

Entre os principais pacotes que compõem a plataforma ROS e que foram utilizados na implementação do sistema embarcado no robô MARIA estão os seguintes:

- *GMapping*⁴ é um *datalogger* de partículas *Rao-Blackwellized*⁵ utilizado para criar um mapa de grade de ocupação com os dados provenientes de um sensor laser. Gmapping fornece o SLAM (Mapeamento e Localização Simultâneos) como um nó ROS chamado *slam_gmapping*. Usando *slam_gmapping* pode-se criar um mapa em grade de ocupação 2D a partir dos dados coletados por um robô móvel.
- *Map server*⁶ é um nó de ROS que lê um mapa de um diretório e o disponibiliza através de um serviço de ROS. Além disso, oferece um utilitário que permite recuperar

³Disponível em <http://wiki.ros.org/>

⁴Disponível em <http://wiki.ros.org/gmapping>

⁵Rao-Blackwellized é uma técnica para explorar a estrutura de estado-espaço e assim, reduzir o número de partículas na hora de estimar a trajetória do robô

⁶Disponível em http://wiki.ros.org/map_server

dados de mapas e salvá-los em um formato pgm e um arquivo yaml.

- *OpenNI Launch*⁷ é um pacote que contém as bibliotecas de acesso aos dispositivos do tipo *OpenNI*. Tais recursos, são compatíveis com o Microsoft Kinect no ROS. O pacote cria um nó que transforma os dados do dispositivo em nuvens de pontos, mapas de disparidade, e outras representações adequadas para o processamento e visualização.
- *Pocketsphinx*⁸ é uma biblioteca de reconhecimento de fala, que permite fornecer acesso ao reconhecedor de voz *CMU Pocket Sphinx*.
- *Sound Play*⁹ é uma biblioteca que fornece um nó ROS na qual são traduzidos texto em sons. O nó suporta sons internos, reprodução de arquivos WAV e OGG.
- ROSARIA¹⁰ fornece ao ROS, mecanismos que permitem reconhecer várias informações e características no que tange as plataformas robóticas móveis de *MobileRobots Inc.* e *ActivMedia*, inclusive o Pioneer 3-AT. O conjunto de dados fornecidos pela biblioteca são: velocidade, pose estimada, estado dos *bumpers*, controle de aceleração, dentre outras.
- *Stage*¹¹ é uma interface de simulação de robôs e sensores para ambientes bidimensionais que permite a criação do ambiente de trabalho para o robô. Isto proporciona ao usuário uma ferramenta de alto nível para desenvolver o controlador do robô, facilitando a sua programação.
- Rviz¹² é uma ferramenta de visualização 3D para ROS.
- OpenCV é uma biblioteca de livre distribuição para desenvolvimento de aplicações na área de visão computacional. Essa biblioteca vem sendo utilizada em inúmeras aplicações como, por exemplo, em sistemas de segurança com detecção de movimento, em controle de processos onde requer detecção de objetos, dentre outras.

⁷Disponível em http://wiki.ros.org/openni_launch

⁸Disponível em <http://wiki.ros.org/pocketsphinx>

⁹Disponível em http://wiki.ros.org/sound_play

¹⁰Disponível em <http://wiki.ros.org/ROSARIA>

¹¹Disponível em <http://wiki.ros.org/stage>

¹²Disponível em <http://wiki.ros.org/rviz>

3.2.2 AMCL (*Monte Carlo Localization*)

O AMCL¹³ é um algoritmo usado para a localização dos robôs utilizando filtro de partículas [Hiemustra and Nederveen, 2007]. Dado um mapa do ambiente, o AMCL estima a posição e orientação de um robô que se move e detecta o ambiente. Esse algoritmo utiliza um filtro de partículas para representar a distribuição dos estados prováveis, com cada partícula representando um estado possível, ou seja, uma hipótese de onde o robô está. O algoritmo tipicamente começa com uma distribuição aleatória uniforme de partículas ao longo do espaço de configurações, o que significa que o robô não tem nenhuma informação sobre onde ele está e assume que é igualmente provável que ele esteja em qualquer ponto do espaço. Sempre que o robô se move, ele muda as partículas para prever seu novo estado após o movimento, e quando o robô detecta algo, as partículas são re-amostradas com base na estimativa Bayesiana recursiva, ou seja, quão bem os dados detectados reais correlacionam com o estado previsto. Em última instância, as partículas devem convergir para a pose real do robô. Um exemplo do funcionamento desse algoritmo é mostrado na Figura 3.9.

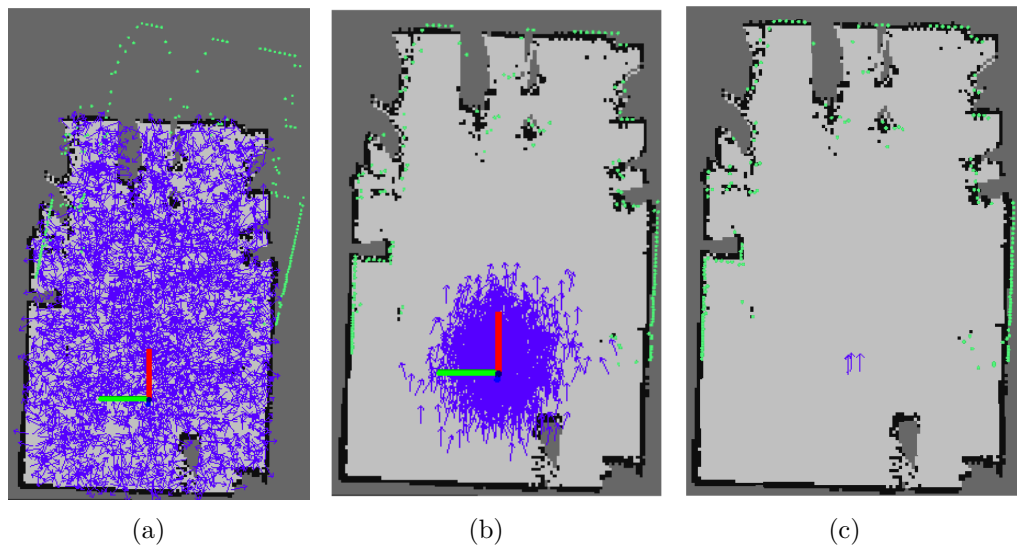


Figura 3.9: Exemplo do funcionamento do AMCL: (a) espalhamento de partículas (b) convergência para a região de localização do robô; (c) posição estimada encontrada.

¹³Disponível em <http://wiki.ros.org/amcl>

3.2.3 RGBDSLAM

O RGBDSLAM¹⁴ é uma aplicação desenvolvida para trabalhar na plataforma ROS, que registra as nuvens de pontos geradas pelos sensores RGB-D como o Kinect, permitindo adquirir rapidamente modelos 3D em forma de nuvens de pontos de ambientes internos. O funcionamento dessa aplicação está sujeito à utilização de recursos visuais e algoritmos como:

- SURF (*Speeded-Up Robust Features*). O algoritmo SURF é utilizado para a extração e descrição de características, independente da escala e rotação [Bay et al., 2006], por meio das técnicas de processamento de imagens estatísticas, incluindo matrizes hessianas para medições do detector e um descritor com base numa amostragem aleatória.
- RANSAC (*RANdom SAmple Consensus*). O RANSAC é um método iterativo que pode ser utilizado para extrair linhas retas de uma leitura de amostras de um sensor de distâncias [Chum et al., 2003]. Além disso, pode-se usar esse método para distinguir linhas retas em um ambiente tridimensional, tais como ambientes formados por paredes, tetos, pisos e móveis que podem ser encontrados em um ambiente interno. Um exemplo da extração de linhas mediante RANSAC é mostrado na Figura 3.10.

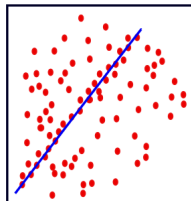


Figura 3.10: Exemplo de extração de linhas pelo método RANSAC.

- ICP (*Iterative Closest Point*). O ICP é um algoritmo utilizado para minimizar as diferenças entre duas nuvens de pontos e é comumente empregado para alinhar superfícies bidimensionais ou tridimensionais de capturas de imagens diferentes, para a localização de robôs [Segal et al., 2009]. Os pontos são associados de acordo com os critérios do vizinho mais próximo e a transformação entre nuvens é estimada por uma função de mínimos quadrados. Portanto, uma nuvem de pontos é definida como referência e mantida fixa, enquanto a outra é transformada utilizando os parâmetros estimados em pontos anteriores. O processo é repetido iterativamente até convergir.

¹⁴Disponível em <http://wiki.ros.org/rgbdsllam>

- HOGMAN. O algoritmo HOGMAN é utilizado para o cálculo da posição mais provável do robô. Esse algoritmo utiliza uma representação do espaço do problema, nesse caso o grafo das posições e das restrições em diferentes níveis. Assim, utilizando uma representação não-euclidiana do espaço tridimensional como simplificação da modelagem do ambiente, o algoritmo consegue uma modelagem susceptível para ser dividida em camadas, sendo que a complexidade do modelo avança conforme se aprofunda na estrutura em níveis.

Através da Figura 3.11 observam-se as quatro etapas executadas pelo algoritmo RGBDSLAM, sendo que na primeira, as características são extraídas das informações obtidas pelo sensor e combina pares de imagens adquiridas por meio do algoritmo SURF. Na segunda etapa, o algoritmo RANSAC estima a transformação 3D entre os pares de imagens obtidas. Enquanto que, na terceira etapa, é refinada a posição estimada usando uma implementação do algoritmo ICP. Na quarta etapa, o gráfico é então otimizado com HOGMAN para reduzir os erros de posição acumulados [Pons, 2012].

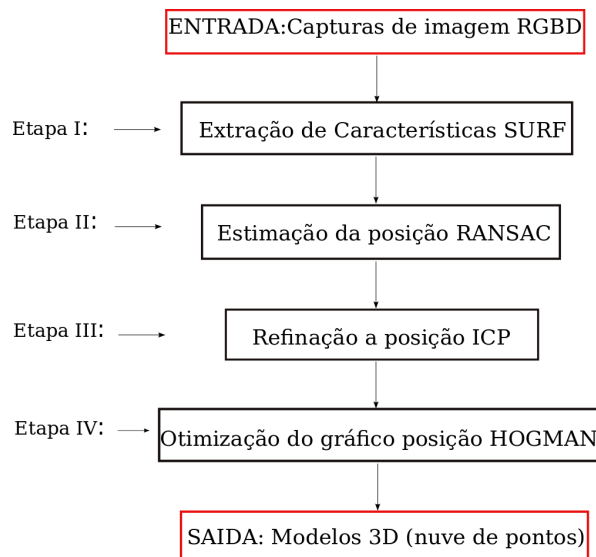


Figura 3.11: Esquema hierárquico de execução dos processos vinculados à aplicação RGBDSLAM.

3.2.4 OctoMap

O OctoMap é uma biblioteca que implementa uma abordagem de mapeamento 3D em grade de ocupação, proporcionando estruturas de dados e algoritmos de mapeamento em C++, particularmente adequados para robótica [Hornung et al., 2013]. A implementação do mapa é baseada em uma *octree* e é projetado para atender os seguintes requisitos:

- **Modelo 3D Completo.** O mapa é capaz de modelar ambientes arbitrários sem conhecimento prévio sobre ele, representando as áreas ocupadas bem como o espaço livre. Áreas desconhecidas do ambiente estão implicitamente codificadas no mapa, enquanto a distinção entre o espaço ocupado e livre é essencial para a navegação segura dos robôs.
- **Atualizável.** É possível adicionar informação ou novas leituras dos sensores em qualquer momento. Modelagem e atualização são feitas de forma probabilística. Além disso, vários robôs são capazes de contribuir para criar o mesmo mapa e um mapa previamente gravado é extensível quando novas áreas são exploradas.
- **Flexível.** O mapa é multi-resolução, de modo que, por exemplo, um planejador de alto nível é capaz de usar um mapa não aperfeiçoado, enquanto um planejador local pode operar usando uma resolução fina.
- **Compacto.** O mapa é armazenado de forma eficiente, tanto na memória quanto no disco. É possível gerar arquivos compactados para uso posterior ou troca conveniente entre robôs, mesmo sob as restrições de largura de banda.

3.3 Desenvolvimento do sistema de navegação

Uma vez explicada a composição da plataforma robótica e das ferramentas computacionais utilizadas, torna-se necessário apresentar cada etapa mostrada no esquema das Figuras 3.1 e 3.2 e, como eles estão ligados uns com outros para a execução da navegação.

3.3.1 Mapeamento

Nesta dissertação utilizou-se a ideia de que a forma mais eficiente de se navegar manipuladores móveis é utilizar mapas bidimensionais que levem em consideração os obstáculos de grande altura presentes no caminho do robô e que, os mapas tridimensionais sejam usados apenas quando o robô precisar realizar uma tarefa de detecção e manipulação de objetos. Mantém-se a ideia de que a localização e a navegação em 2D possuem soluções bem conhecidas e muito mais eficientes que aquelas obtidas em 3D.

Para mapear o ambiente utilizou-se o sensor laser para capturar as características do ambiente, os dados da odometria do robô e do algoritmo *gmapping* que é um filtro de partículas Rao-Blackwellized altamente eficiente para a criação de um mapa em grade de ocupação bidimensional. A abordagem Rao-Blackwellized utiliza um filtro de partículas

em que cada partícula carrega um mapa individual do ambiente. O algoritmo *gmapping* precisa de uma transformação entre os sistemas de referência da base móvel e do sensor laser, pois essa transformação define os deslocamentos tanto em termos de translação como de rotação entre os diferentes sistemas de referência.

A rotação do sistema de referência do laser coincide com a referência da base móvel, mas existe uma traslação em x de 15,5 cm e em z de 20,2 cm. Esse procedimento detalhado é considerado como o modulo IB, na qual a resolução escolhida para a grade de ocupação foi de 5 cm por cada pixel e o mapa bidimensional gerado foi utilizado para estimar a posição do robô dentro do ambiente enquanto se locomove.

Por outro lado, o sensor Kinect foi utilizado junto com o pacote RGBDSLAM para capturar as características do ambiente em forma de um mapa tridimensional colorido. RGBDSLAM permite adquirir rapidamente modelos 3D coloridos de objetos e cenas de interior com uma câmera Kinect. Ele utiliza recursos visuais SURF ou SIFT para combinar pares de imagens adquiridas, e usa RANSAC para estimar de forma robusta a transformação 3D entre eles. Para conseguir o processamento on-line, a imagem atual é comparada apenas contra um subconjunto das imagens anteriores. Posteriormente, ele constrói um grafo cujos nós correspondem a pontos de vista da câmera e cujos limites correspondem às transformações 3D estimadas. O gráfico é então otimizado com HOG-Man para reduzir os erros acumulados.

RGBDSLAM precisou também da transformação entre o sistema de referência da base móvel e o sistema de referência do Kinect, para que tanto o mapa bidimensional como o tridimensional estejam alinhados ao sistema de referência da base móvel.

O sistema de referência do Kinect sofreu uma rotação $\mathcal{R}_z = -90^\circ$ e $\mathcal{R}_y = -90^\circ$, com relação à referência da base móvel e, uma traslação em z de 150 cm e em x de 14 cm. Por meio da Figura 3.12, pode-se observar os sistemas de referência atribuídos a cada elemento da plataforma robótica que foram utilizados para o alinhamento no momento da geração dos mapas. O modulo IA e IB observados na Figura 3.1 são executados simultaneamente.

Uma vez que o ambiente foi percorrido e o mapa 3D colorido foi concluído, gerou-se o mapa em grade de ocupação 3D binário baseado em octree com a utilização da biblioteca OctoMap com o objetivo de modelar ambientes com representação de áreas, bem como o espaço livre e ocupado e áreas desconhecidas do ambiente são implicitamente codificada no mapa. Portanto, obtendo esse mapa é possível editá-lo com a mesma biblioteca e considerar somente os obstáculos até a altura do robô, eliminando o conteúdo em todas as grades do mapa localizadas por acima do sistema de referência do Kinect, nominando

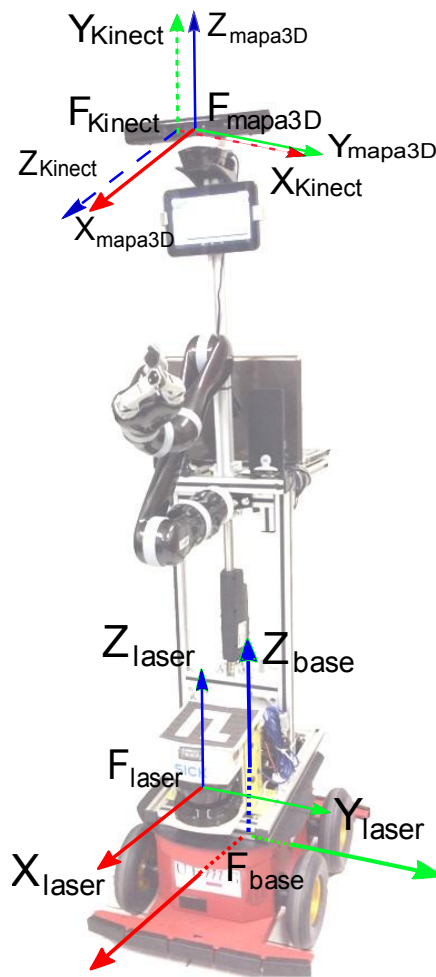


Figura 3.12: Sistemas de referência atribuídos a cada elemento da plataforma robótica MARIA.

este procedimento como modulo II. Essa altura segundo a referência do mapa foi de -165 cm em z .

Com a edição do mapa tridimensional terminada, gera-se no modulo III a projeção do mapa tridimensional binário gerando uma grade de ocupação bidimensional. Essa projeção considera os obstáculos que o mapa tridimensional encontrou até a altura do robô.

A projeção esta baseada na filtragem dos valores ao longo do eixo z , excluindo dessa forma todas as grades por cima da altura do robô e, o chão que é um tipo de grade que não pode considerar-se na hora da projeção, gerando-se dessa forma a projeção bidimensional em grade de ocupação. Todo esse procedimento detalhado anteriormente foi realizado no sistema operacional ROS baixo o protocolo que o regula.

Uma vez projetado o mapa, foi conveniente gerar um modulo IV onde a grade de

ocupação é transformada em uma representação onde cada célula tem o valor binário de 0 (zero) ou 1 (um) com o objetivo de se utilizar a implementação de um algoritmo de planejamento de caminhos que considere todos os obstáculos presentes no ambiente e evite colidir com eles. Assim, 0 (zero) e 1 (um) indicam a presença e ausência de obstáculos respectivamente. Utilizando técnicas de processamento de imagens e com ajuda da biblioteca OpenCV, processou-se o mapa projetado. Primeiramente a imagem foi suavizada com um filtro da mediana de tamanho três com o objetivo de eliminar o ruído capturado do ambiente e, logo depois segmentou-se essa imagem para simplificar e modificar a representação da imagem em outra mais significativa e mais fácil de analisar, onde se estabeleceu um limiar de transformação como se mostra a seguir:

$$dsp(x, y) = \begin{cases} 0 & \text{Se } src(x, y) < \tau \\ 1 & \text{Caso contrário} \end{cases} \quad (3.1)$$

onde, $dsp(x, y)$ é o valor da intensidade do pixel na imagem binária, $src(x, y)$ é o valor da intensidade do pixel na grade de ocupação e τ é o valor limiar para a transformação.

Além disso, foi necessário considerar o espaço de configurações do robô. Portanto, dilatou-se o mapa para alcançar esse objetivo. Para definir esse espaço de configurações considerou-se o tamanho do robô com todos os dispositivos que o compõem, lateral e frontalmente. Portanto, foi escolhido o maior raio do robô considerando-o como se fosse circular, permitindo assim que o robô não colida com os obstáculos a seu redor ao momento de locomover-se, como observa-se na Figura 3.13.

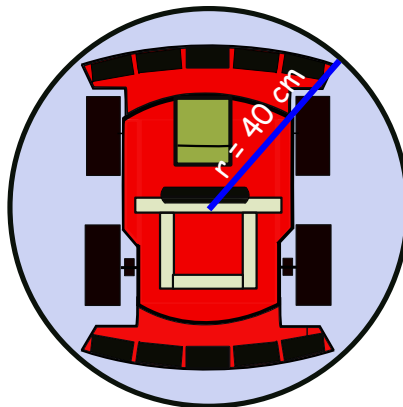


Figura 3.13: Representação do robô MARIA para determinar o espaço de configurações.

Conhecendo o tamanho do robô gerou-se o espaço de configurações, dilatando a imagem processada do mapa bidimensional projetado. O valor do raio do robô foi de 40 cm que, para torna-lo pixels, levou-se em consideração a resolução do mapa projetado. Dessa

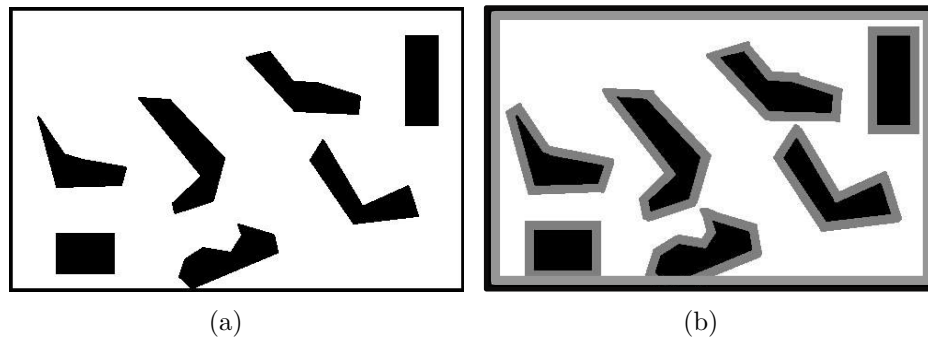


Figura 3.14: Exemplo da criação do espaço de configurações: (a) Imagem original; (b) Espaço de configurações.

forma, mostra-se na Figura 3.14 um exemplo onde se considera o espaço de configurações do robô.

Uma vez finalizado o pré-processamento da imagem, essa deu início à próxima etapa, ou seja, a seleção de pontos referenciais.

3.3.2 Seleção de pontos referenciais no ambiente

A imagem processada obtida na subseção anterior é a representação do ambiente que considera os obstáculos até a altura do robô. Por esse motivo, essa imagem foi utilizada para realizar a seleção dos pontos objetivos que serão considerados referenciais no ambiente, ou seja, pontos finais no ambiente que o robô devia atingir quando o usuário se referir a um lugar. Para que o usuário realize a seleção desses pontos foi necessário primeiro que o mapa já considere o espaço de configurações do robô, evitando que os pontos selecionados sejam de uma posição onde que o robô possa colidir com os obstáculos a seu redor. Esse espaço de configurações permitiu tratar ao robô como um ponto.

A imagem dilatada foi mostrada ao usuário na tela do computador para a seleção dos pontos com o *mouse*. Uma vez selecionado um ponto, o algoritmo implementado permitiu criar uma base de dados com as coordenadas dos pontos objetivos selecionados com referência a um sistema inercial do ambiente real, colocado na parte esquerda inferior do mapa que o representa.

Quando o ponto era selecionado as coordenadas em pixels desse ponto estavam com respeito ao sistema de referência da imagem, ou seja, canto superior esquerdo como mostra na Figura 3.15 (a), então foi realizada a conversão desses pixels à referência do mapa do ambiente real como visto na Figura 3.15 (b).

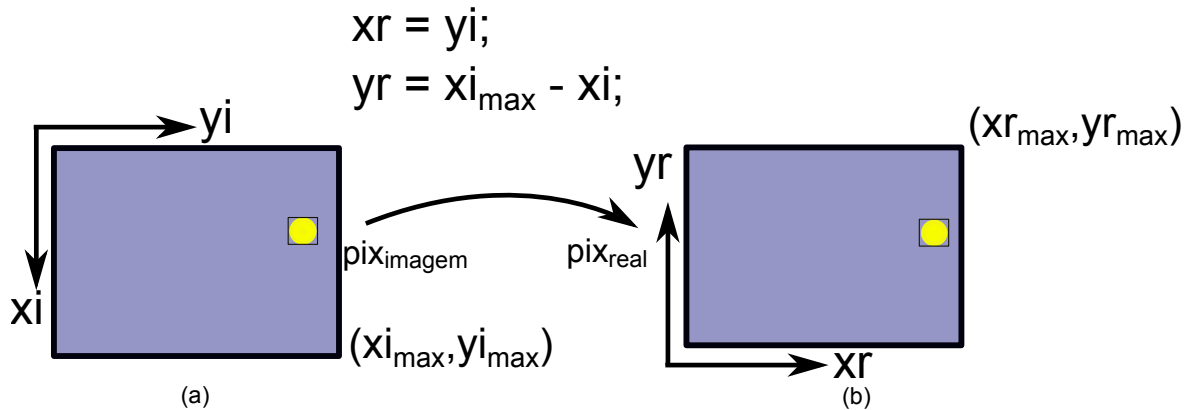


Figura 3.15: Conversão dos sistemas de referência: (a) da imagem binária para (b) o ambiente real.

Desse modo, procedeu-se a salvar na base de dados a posição convertida para o novo sistema de referência associado a um nome que permitia identificar aquele ponto objetivo.

Para que esse procedimento seja executado com êxito, apresentou-se ao usuário uma interface no computador, permitindo selecionar os pontos objetivos e digitando os nomes de cada lugar. Esses dados são salvos em uma base de dados, sendo esse processo repetido para todos os pontos que o usuário desejava ter como referência. A base de dados foi utilizada na próxima etapa que é o reconhecimento de voz.

3.3.3 Reconhecimento e Confirmação de voz

Para o reconhecimento da fala, foi utilizada a biblioteca *Pocketsphinx*, com a qual se implementou uma estratégia que permitiu reconhecer essa fala em forma de instruções de voz. Para a confirmação por parte do robô foram utilizadas as bibliotecas *sound play* e *festival*.

Sound play permite traduzir textos em sons e fazer sínteses de voz via *festival*. A biblioteca *Pocketsphinx* fornece acesso ao reconhecedor de voz *CMU Sphinx*. Assim, o reconhecedor requer um modelo de linguagem e um arquivo de dicionário com todas as palavras que serão identificadas. Desse modo, a ferramenta de base do conhecimento *Sphinx*¹⁵ constrói um conjunto de arquivos com modelos lexicais e de linguagem para os decodificadores *Sphinxs*, a partir de um arquivo que contém as palavras ou sentenças que o decodificador reconhecerá. As palavras no arquivo devem ser colocadas uma em cada

¹⁵Disponível em <http://www.speech.cs.cmu.edu/tools/lmtool-new.html>

linha, sem precisar de pontuação e no idioma inglês. A desvantagem dessa ferramenta é precisamente que está configurada somente para o funcionamento no idioma inglês.

Para que as instruções de voz sejam reconhecidas e se execute a tarefa falada, foi implementado um programa com uma estratégia que permitiu detectar as palavras reconhecidas pela biblioteca e concatenar em uma instrução de voz com uma ordem específica, permitindo assim evitar erros na escuta no decodificador.

Um exemplo de uma ordem geral para reconhecer a instrução é mostrada a seguir:

1. Maria, **2.** go, **3.** to, **4.** the, **5.** bedroom.

Uma variante da instrução seria simplesmente falar a primeira e a última palavra (e.g. Maria bedroom), e o programa completa automaticamente a instrução.

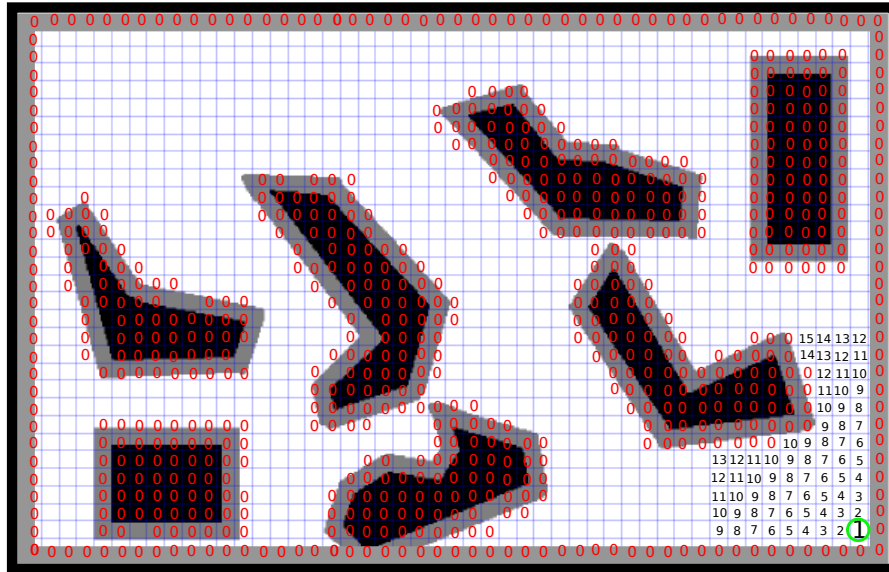
Quando a instrução é identificada, a plataforma robótica realiza uma pergunta, espera a confirmação por parte do usuário para a instrução. Uma vez reconhecida a instrução é identificado o nome do lugar onde o robô deve atingir. Com o conhecimento do lugar, usou-se a base de dados da subseção anterior para procurar por esse lugar mencionado e a posição onde esse fica dentro do ambiente. Desse modo, uma vez encontrada a posição do lugar é enviado o ponto objetivo para o planejamento da trajetória.

3.3.4 Planejamento da trajetória

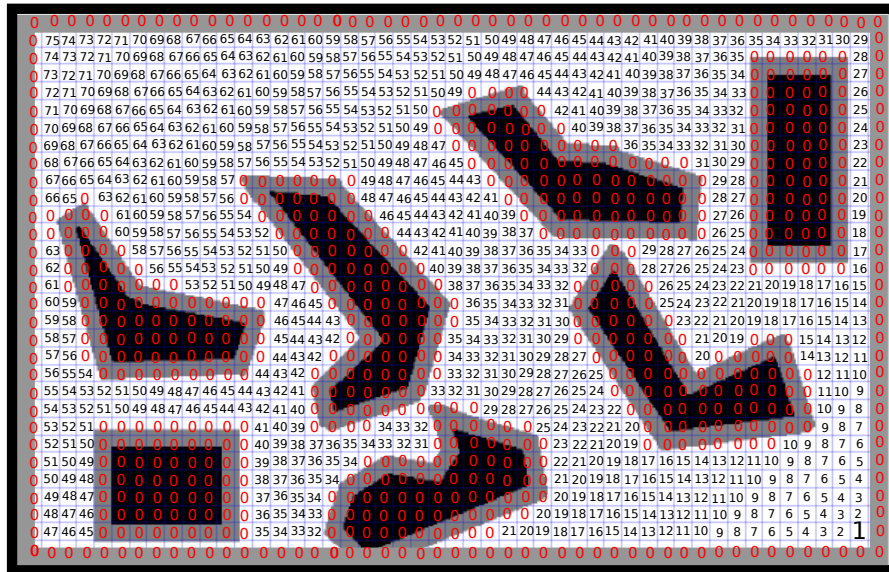
A etapa do reconhecimento de voz envia a esta etapa as coordenadas do ponto objetivo que o usuário deseja que o robô atinga. Com essa informação, inicia-se a execução do algoritmo de planejamento de caminho por frentes de ondas no mapa binário que considera o espaço de configurações do robô. Assim, para a execução desse algoritmo foi necessário criar uma grade no espaço de configurações. De tal modo, permitiu-se discretizar o mapa binário com uma grade onde se podia rotular as células para o crescimento da frente de onda. O tamanho de cada célula foi de 8×8 pixels, o mesmo valor do raio do robô.

Logo depois, rotulou-se a célula que continha o ponto objetivo com o valor de 1 e, a partir daí se considerou uma vizinhança de 4 para o crescimento da frente de onda. Os vizinhos da célula objetivo que tinham um valor de intensidade de 255, ou seja, células livres de obstáculos, foram colocados com o rótulo 2. Em seguida, todas as células adjacentes a 2 que tinham intensidade 255 foram rotulados com 3 e assim por diante, até preencher por completo todas as células livres no mapa do ambiente.

Um exemplo do procedimento da rotulação da frente de ondas em um mapa é mostrado na Figura 3.16.



(a)



(b)

Figura 3.16: Exemplo de funcionamento do algoritmo por frentes de ondas (a) rotulação do ponto objetivo e dos vizinhos mais próximos; (b) rotulação completa do ambiente.

Uma vez finalizada a rotulação, a geração da trajetória até o objetivo foi realizada partindo da estimativa da posição inicial do robô. Essa estimativa da posição dentro do mapa do ambiente foi obtida utilizando o algoritmo de localização Monte Carlo (AMCL). Logo depois, procedeu-se à busca do rótulo que identifica a posição em uma estrutura em lista que contém os rótulos e as posições em metros de cada célula do mapa. Com esse rótulo e a utilização de uma vizinhança 8 se procedeu a procura do menor rótulo, mostrando dessa maneira o próximo ponto que o robô devia ir. Assim pela frente, seguindo uma sequência decrescente até chegar à célula de rótulo 1 que corresponde à posição desejada.

Na Figura 3.17 é mostrada um exemplo da geração de uma trajetória em um ambiente, utilizando a distância diagonal.

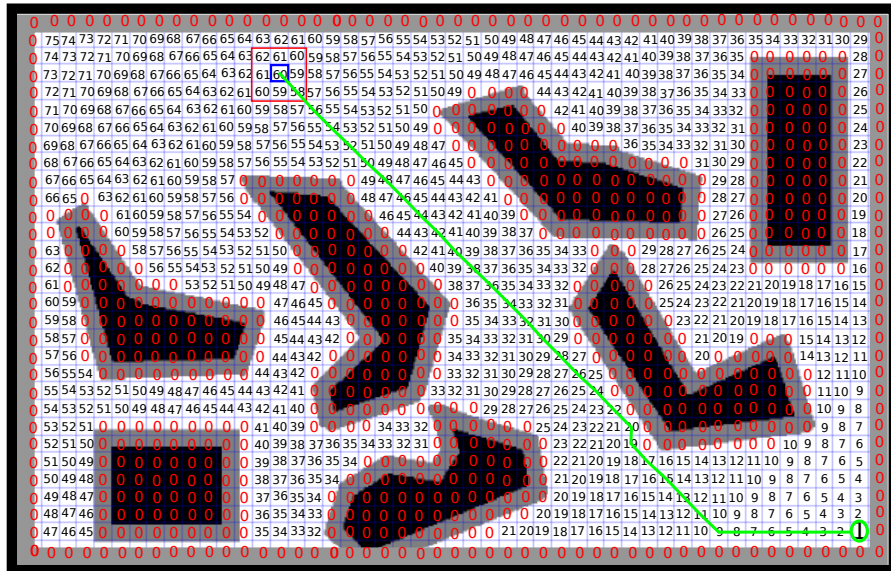
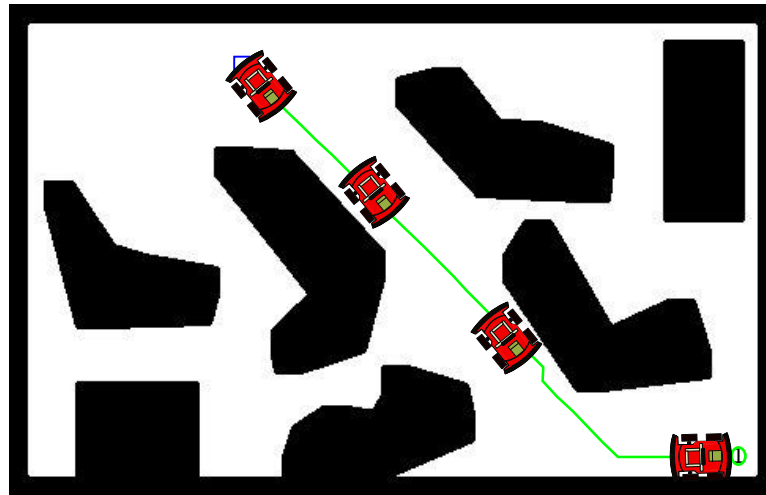
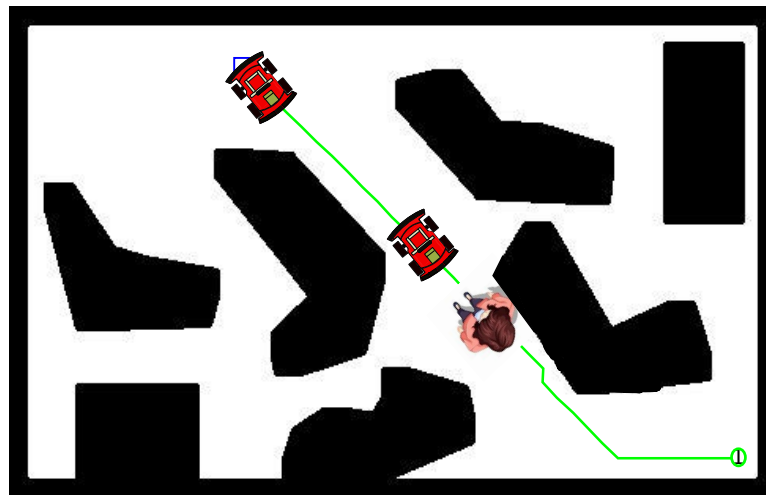


Figura 3.17: Exemplo da geração de trajetória pelo planejador até a posição desejada.

A trajetória é executada pelo robô até chegar ao ponto desejado pelo usuário quando não existir nenhum obstáculo não mapeado nesse caminho que impeça a sua execução. O mapa criado no início do processo é considerado um mapa estático, portanto, para evitar obstáculos que estiverem fora do mapa é realizada simultaneamente com a execução da trajetória uma verificação do próximo ponto a ser alcançado pelo robô. Com a informação do mapa e dos dados adquiridos com o sensor laser, procede-se sempre a perguntar se o próximo ponto se mantém livre ou ocupado. Se estiver ocupado, a um metro de distância o robô gera comandos de voz pedindo licença, para que esse obstáculo possa se retirar. Se o obstáculo continuar na mesma posição a uma distância de 40 cm, o robô para até que ele saia dessa posição. Enquanto o obstáculo não se retirar, a plataforma robótica



(a)



(b)

Figura 3.18: Execução da trajetória pelo robô (a) sem restrições; (b) Interrompida por um obstáculo fora do ambiente previamente modelado.

permanecerá parada. Se o obstáculo sair o robô continuará normalmente sua execução. Um exemplo desta etapa é mostrada na Figura 3.18.

4 *Resultados Experimentais*

Neste capítulo serão apresentados os resultados obtidos a partir da metodologia mostrada no capítulo anterior. Para cada etapa definida e descrita, são expostos os resultados obtidos a fim de demonstrar o funcionamento do sistema de navegação.

Para os resultados mostrados a seguir, a central de processamento de dados da plataforma MARIA foi um notebook HP com as seguintes características: processador Intel Core 2 Duo T5900, 2,20Ghz, 4GB de memória RAM e sistema operacional Linux Ubuntu 12.04 LTS.

4.1 Resultados do mapeamento

Como descrito na metodologia, para permitir o controle do robô pelo cenário desejado no momento do mapeamento, implementou-se um *software* para fazer que o robô se locomova pelo ambiente e assim permitir que o sensor laser capture as características do entorno. Definiu-se uma velocidade de 0,08 rad/s para o giro direito e esquerdo do robô e uma velocidade de deslocamento de 0,002 m/s. Foram escolhidos três cenários distintos em ambientes internos (rotulados como cenários I-III) para realizar o mapeamento.

4.1.1 Cenário I

O mapeamento do cenário I foi realizado na sala de convivência dos alunos de Mestrado do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais. Neste cenário, objetivou-se apresentar o mapeamento de um ambiente interno tipicamente estruturado com mesas, cadeiras, sofá, etc, semelhante a uma casa. A Figura 4.1 mostra o cenário envolvido, visto de várias perspectivas, enquanto que a Figura 4.2 permite visualizar a planta baixa do ambiente com os objetos contidos nele e, a trajetória realizada durante o mapeamento. É importante salientar que no cenário em questão, foram considerados obstáculos de várias dimensões, de forma a aferir um maior nível de

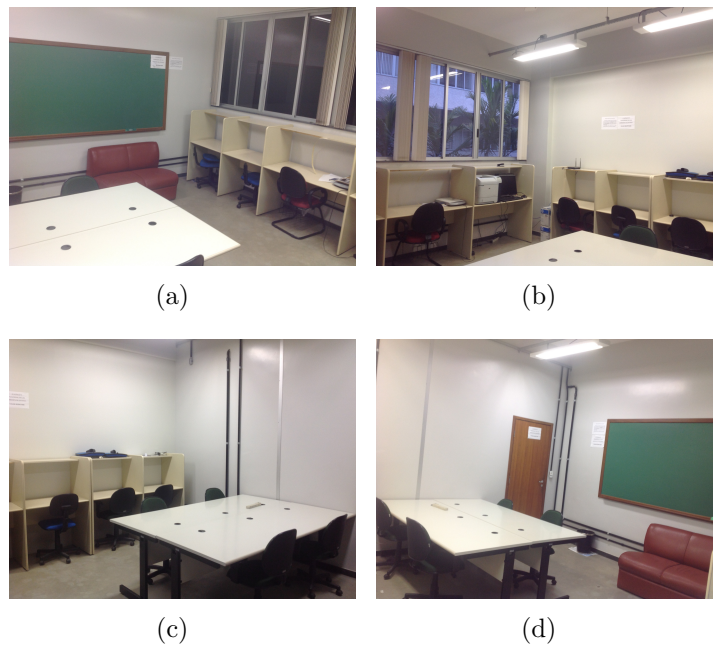


Figura 4.1: Cenário I, observado de quatro pontos de vista diferentes.

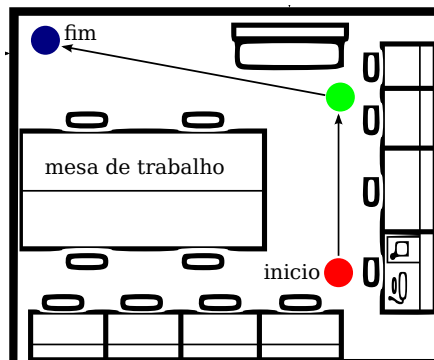


Figura 4.2: Planta baixa e representação da trajetória realizada no cenário I.

complexidade, bem como tornar o ambiente mais próximo de um cenário doméstico real.

No processo de mapeamento, a plataforma robótica se posicionou em um zero absoluto definido no cenário como *início* e, rotacionou-se em torno de seu próprio eixo de maneira a capturar a primeira visão do mapa a ser criado. O robô então se deslocou seguindo as setas mostradas na Figura 4.2 e em cada círculo colorido girou novamente em torno de seu eixo. Esse processo foi repetido até a conclusão da trajetória mostrada.

Obteve-se simultaneamente um mapa bidimensional utilizando o sensor laser e um mapa tridimensional colorido obtido com o sensor Kinect, sendo esses coerentes com o ambiente e espacialmente sincronizados como pode ser observado na Figura 4.3 (a) e na Figura 4.4 (a) e (c) respectivamente. A partir desse último mapa, conseguiu-se, também,

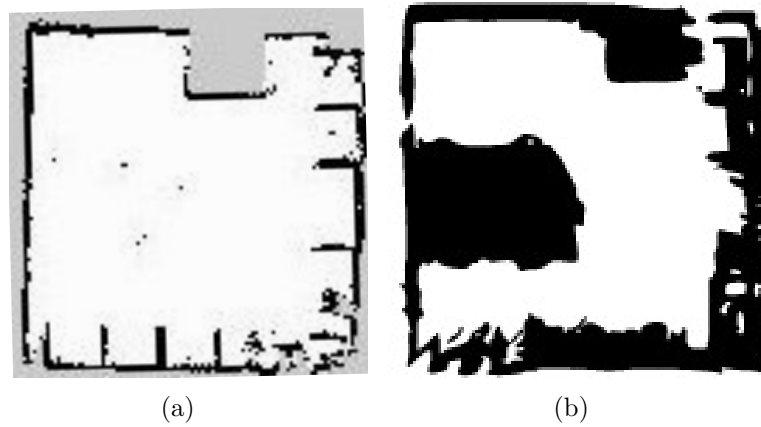


Figura 4.3: Mapas bidimensionais do Cenário I. (a) Capturado pelo laser; (b) Projeção obtida a partir do mapa 3D binário.

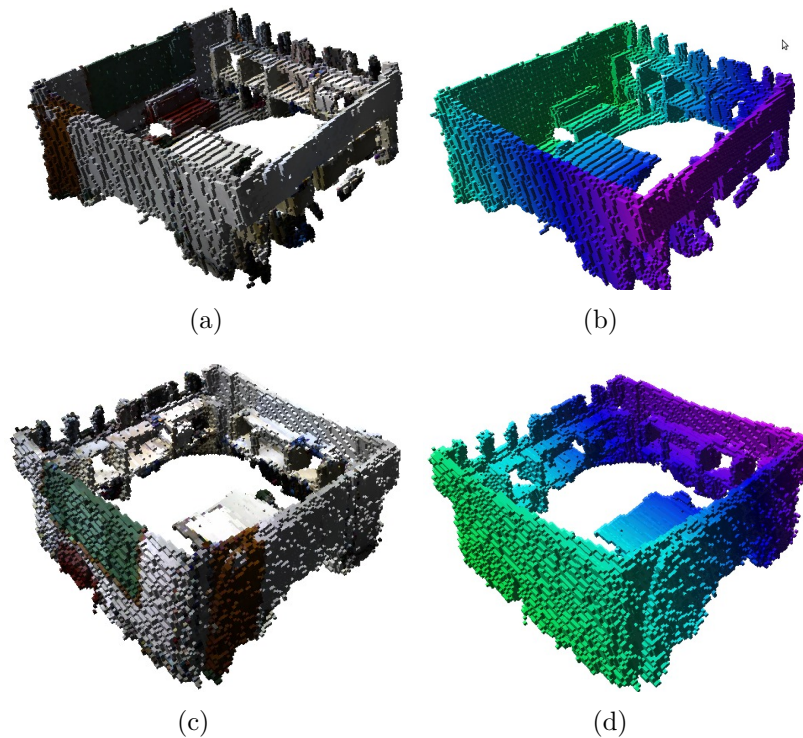


Figura 4.4: Mapas tridimensionais do Cenário I. (a) e (c) Mapas 3D coloridos obtidos com a aplicação RGBDSLAM; (b) e (d) Mapas 3D binários usados para obter a projeção em 2D.

o mapa 3D binário mostrado na Figura 4.4 (b) e (d) que permitiu a criação da projeção bidimensional vista na Figura 4.3 (b).

Considerando que o principal problema nesta etapa consiste em avaliar a representação do ambiente tendo em conta a altura do robô, percebe-se que o mapa 2D obtido pelo sensor laser conforme ilustrado na Figura 4.3 (a), não detecta os obstáculos cuja altura encontra-

se acima do seu campo de visão. Esse problema foi contornado com o uso do sensor Kinect e do mapa binário da Figura 4.4 (b), criando a projeção desse último e obtendo o mapa 2D projetado da Figura 4.3 (b).

No mapa 2D projetado é possível observar que a altura da plataforma MARIA foi considerada. A partir dessa representação, tem-se uma melhor estimativa dos espaços livres e ocupados, o que permitiu um planejamento de caminhos seguro para o robô. Conforme ilustrado na Figura 4.3 (b), pode-se identificar que a região mais escura representa os obstáculos encontrados, por exemplo, no setor esquerdo observa-se uma mesa, enquanto que na Figura 4.3 (a) consegue-se apenas perceber as pernas da mesa e das cadeiras que estavam ao seu redor.

É importante destacar que o tempo de captura dos dados está intimamente relacionado ao tempo que o usuário leva para navegar com o robô pelo ambiente. Após o mapeamento, o tempo gasto para salvar os mapas obtidos foi em média: 3 segundos para salvar o mapa 2D do laser, 13 minutos para salvar o mapa colorido em 3D e 5 segundos para salvar o mapa binário. Para enviar a nuvem de pontos ao OctoMap o tempo foi de 1 minuto. Por fim, o tempo de processamento computacional da imagem 3D e a projeção para 2D é de aproximadamente 1,5 minutos.

4.1.2 Cenário II

Outro ambiente considerado para o mapeamento foi o Laboratório de Sistemas de Computação e Robótica (CORO) da Universidade Federal de Minas Gerais. Assim como na sub-seção anterior, deseja-se obter o mapeamento de um ambiente interno tipicamente estruturado com mesas, cadeiras, cadeira de rodas, etc. Esse cenário apresenta alguns obstáculos diferentes dos considerados no cenário I, o qual faz dele um ambiente propício de testes.

A Figura 4.5 ilustra uma visão geral do espaço mapeado em várias perspectivas e a Figura 4.6 mostra a planta baixa desse ambiente.

Como no mapeamento anterior, o robô deslocou-se na trajetória definida pelas setas, girando no seu próprio eixo em cada círculo marcado, conforme ilustrado na Figura 4.6, até finalizar a trajetória estabelecida.

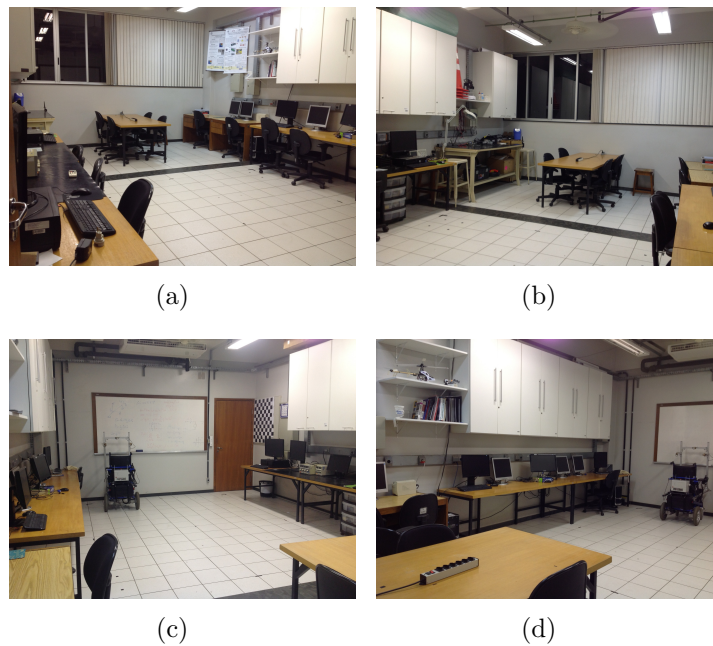


Figura 4.5: Cenário II, observado de quatro pontos de vista diferentes.

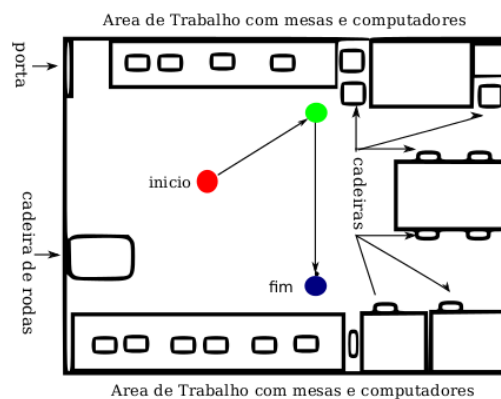


Figura 4.6: Planta baixa e representação da trajetória realizada no cenário II.

Após a realização da aquisição dos dados desse processo, foram obtidos os mapas correspondentes ao cenário II mostrados nas Figuras 4.7 e 4.8, onde se evidencia, assim como no cenário anterior, que a projeção criada a partir do mapa 3D considera os obstáculos não detectados no mapa gerado pelo laser.

Por meio das características visuais dos mapas, pode-se notar que eles podem ser considerados satisfatórios, visto que correspondem a representação dos obstáculos encontrados no ambiente. Dessa forma, possibilitam a navegação, além de manter uma boa representação do cenário. É importante destacar que o tempo de captura dos mapas deste cenário foi igual que no cenário I.



Figura 4.7: Mapas 2D encontrados para o Cenário II. (a) Capturado pelo laser (b) Projeção obtido a partir do mapa 3D binário.

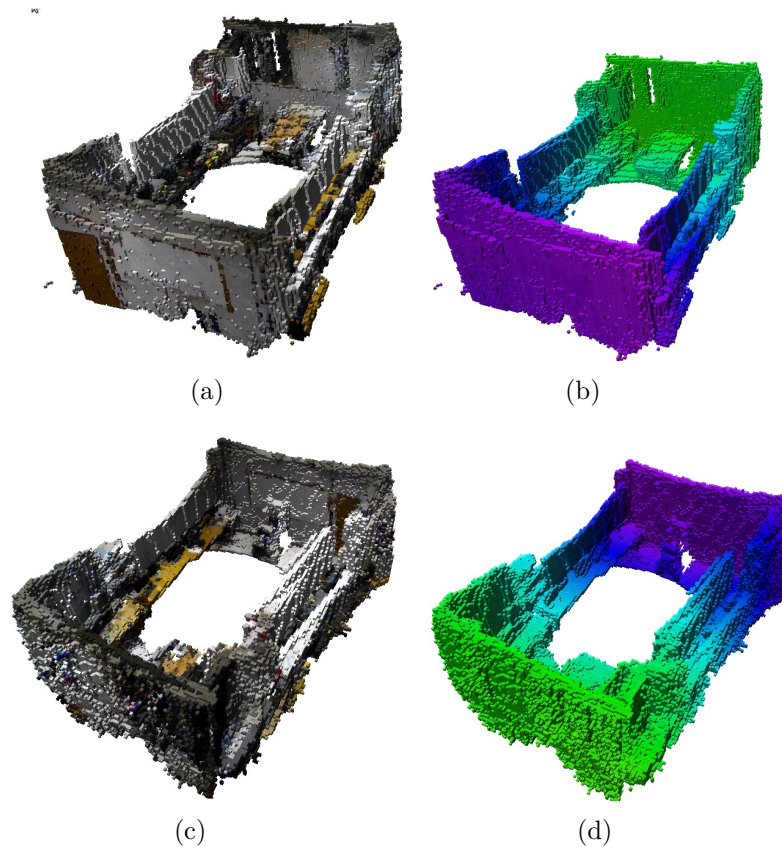


Figura 4.8: Mapas tridimensionais do Cenário II (a) Mapa 3D colorido obtido com ajuda da aplicação RGBDSLAM (b) Mapa 3D binário usado para obter a projeção em 2D.

4.1.3 Cenário III

O mapeamento do terceiro cenário é considerado mais completo, pois considera um espaço interno maior incluindo corredores e várias salas de trabalho. Nesse caso, tomou-se como cenário as dependências do programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Minas Gerais. O objetivo do mapeamento desse cenário foi

considerar não somente um lugar interno composto de mesas, cadeiras, dentre outras coisas, mas também corredores e várias salas internas, permitindo que funcione em um ambiente interno humano completo.

O cenário envolvido inclui os cenários mostrados nas sub-seções anteriores, além da sala de convivência dos alunos de Doutorado, os corredores e obstáculos que o contornam, como observado na Figura 4.9. A planta baixa desses cenários é mostrada em forma geral na Figura 4.10, assim como a trajetória realizada para obter os mapas do seu ambiente.

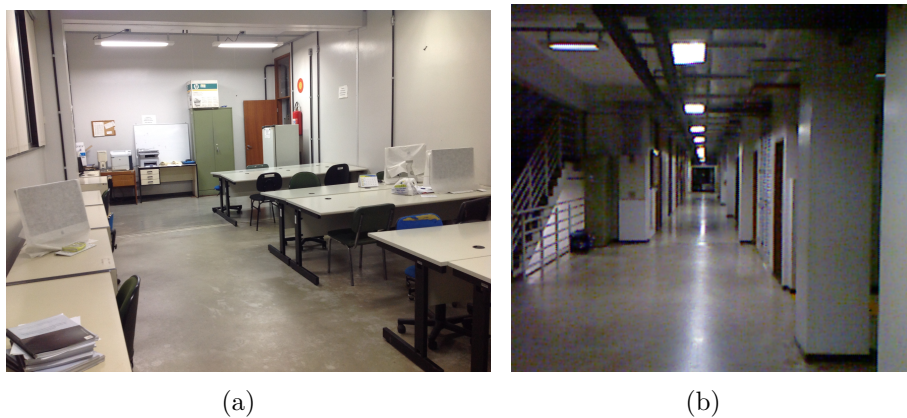


Figura 4.9: Cenário III, pontos de vista diferentes.

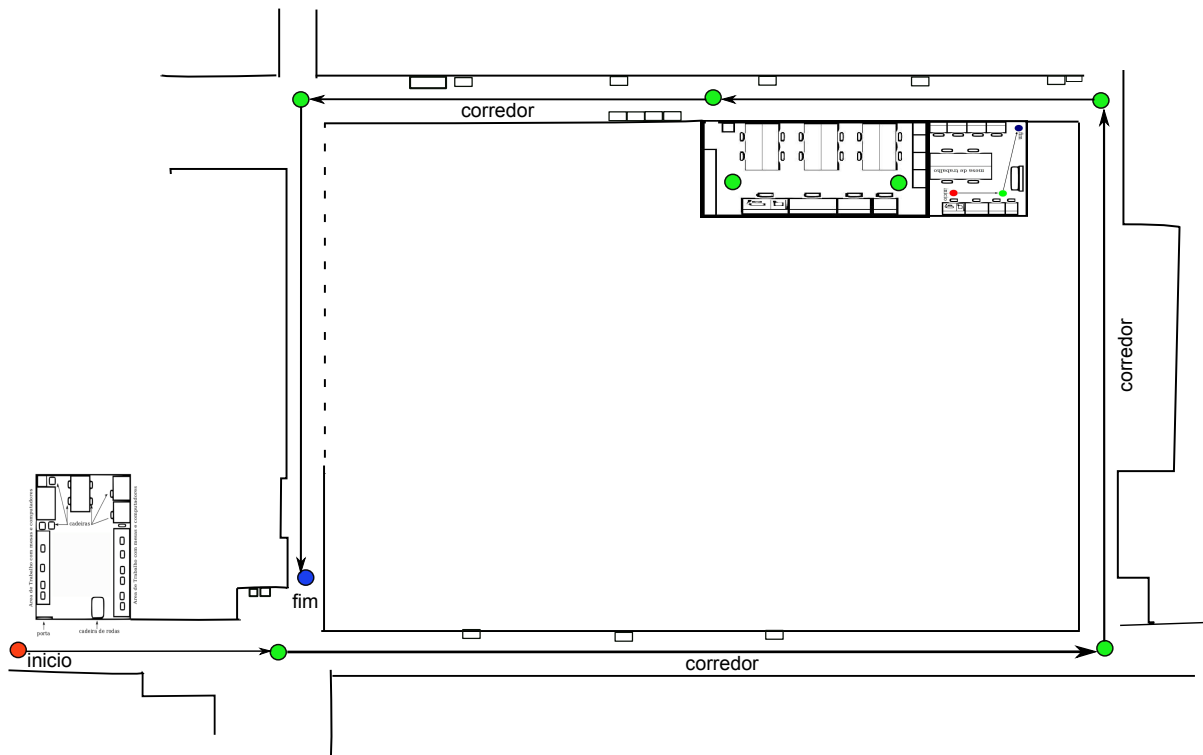
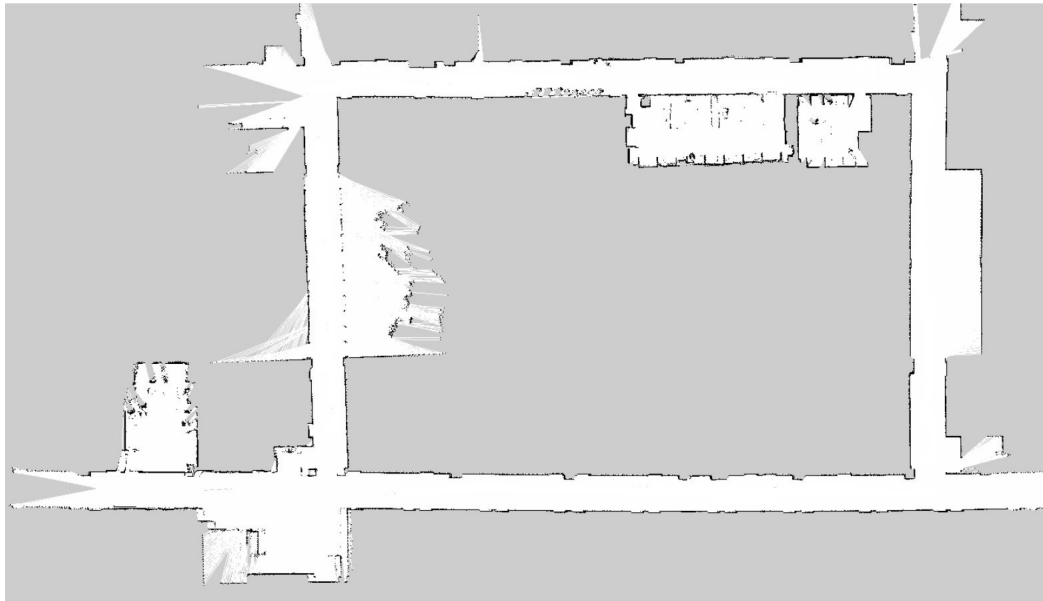
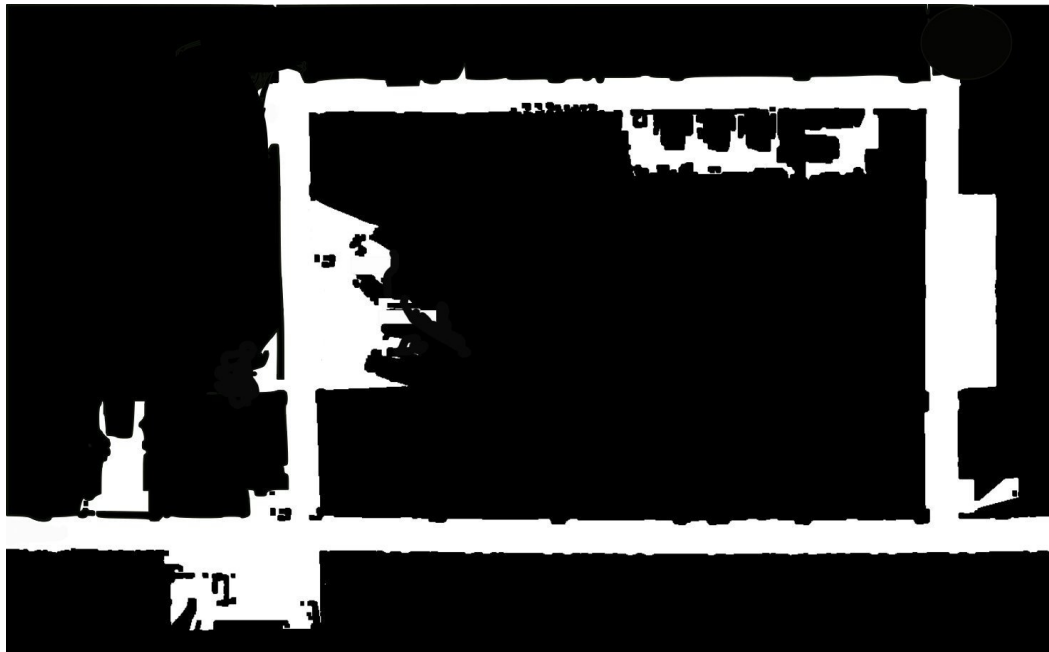


Figura 4.10: Planta baixa e representação da trajetória realizada no cenário III.

Os mapas obtidos tanto bidimensionalmente com o laser e tridimensionalmente com o Kinect quanto os mapas binários e suas projeções são mostradas nas Figuras 4.11 e 4.12.



(a)



(b)

Figura 4.11: Mapas bidimensionais do cenário III. (a) Obtido com ajuda do laser (b) Projeção obtida a partir do mapa 3D binário.

Pode-se observar que o laser captura o ambiente na sua maioria, mas dentro dos lugares internos são mais úteis as projeções que consideram os obstáculos encontrados até a altura do robô.

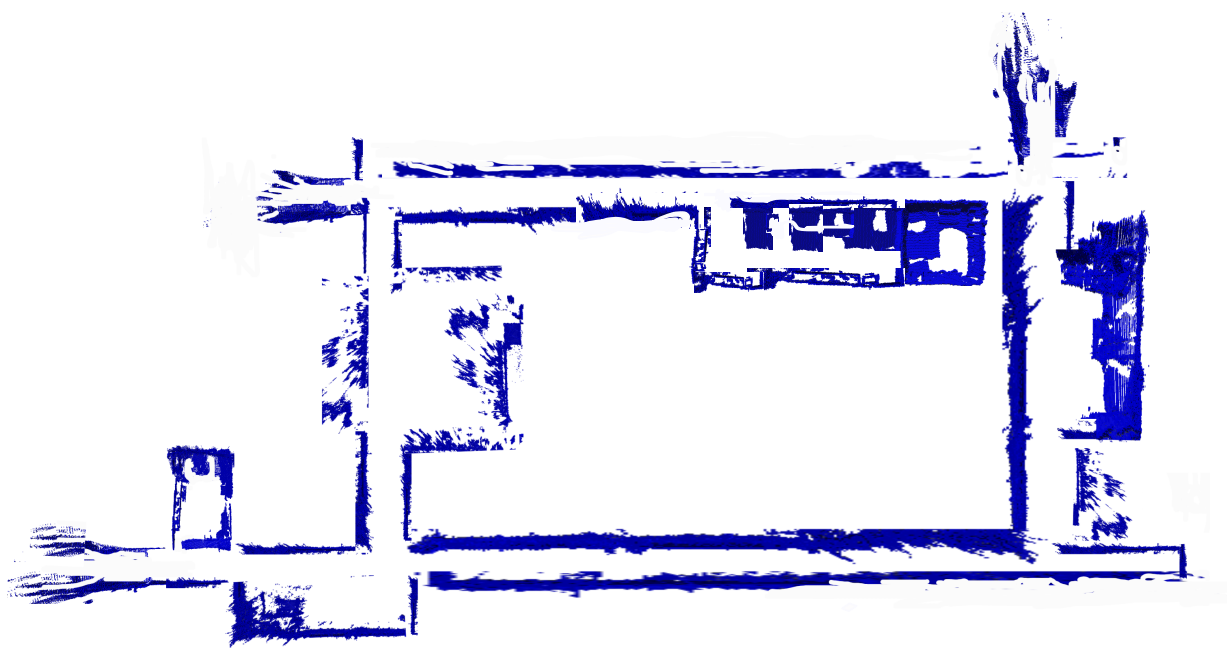


Figura 4.12: Mapa tridimensional binário do Cenário III, vista superior, usado para projeção em 2D.

Similarmente ao cenário I, é importante destacar que o tempo de captura dos dados está relacionado ao tempo que o usuário leva para navegar com o robô pelo ambiente. O tempo gasto para salvar os mapas obtidos foi em média: 3 segundos para salvar o mapa 2D do laser, 30 minutos para salvar o mapa colorido em 3D e 7 segundos para salvar o mapa binário. Para enviar a nuvem de pontos ao OctoMap o tempo foi de 3 minutos. Por fim, o tempo de processamento da imagem 3D e a projeção para 2D é de aproximadamente 1,5 minutos.

4.2 Resultados de Localização

A partir do mapa do cenário II obtido na seção anterior e, fazendo o uso dos dados fornecidos pelo sensor laser e pela odometria do robô, foi obtida a estimativa do posicionamento do robô no ambiente mediante a utilização do sistema de localização probabilística AMCL implementado no ROS. Para validar esse sistema de localização, foram utilizadas quatro câmeras localizadas no teto do cenário II (Figura 4.13), para estimar a pose de um marcador fiducial localizado no robô por meio da utilização da biblioteca `fiducialros`¹⁶ e, compará-lo com a estimativa da posição obtida com o AMCL. Essa biblioteca fornece a

¹⁶Disponível em <http://coro.cpdee.ufmg.br/index.php/software/160-no-ros-para-ler-informacoes-de-fiducial-do-player-no-ros>



Figura 4.13: Posicionamento das câmeras no cenário II.

posição e orientação desse marcador dentro do campo de visão das câmeras.

Na Figura 4.14, é possível identificar uma sequência de passos do algoritmo AMCL para estimar a *pose* do robô dentro do ambiente. Inicialmente, o algoritmo espalha partículas pelo ambiente e, por meio da comparação com os dados adquiridos pelo sensor laser no mapa encontrado na seção anterior e com ajuda do movimento do robô, o algoritmo converge as partículas para a posição real do robô.

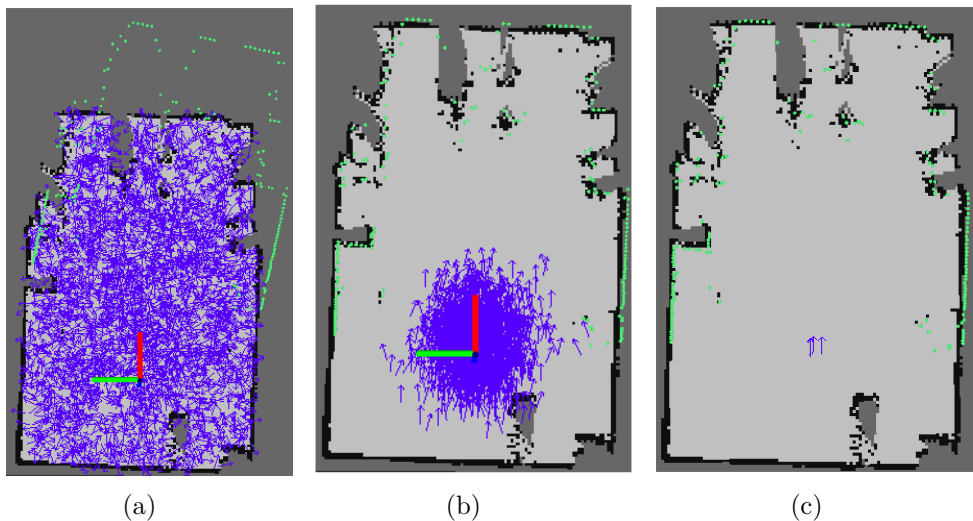


Figura 4.14: Exemplo do funcionamento do AMCL (a) espalhamento de partículas (b) convergência para a região de localização do robô (c) posição estimada encontrada.

Observa-se na Figura 4.15, a trajetória obtida a partir da estimação da posição do robô com AMCL (curva *azul*), assim como a posição do marcador obtida pelas câmeras (curva *vermelha*), permitindo a comparação entre elas e a verificação do funcionamento do algoritmo AMCL.

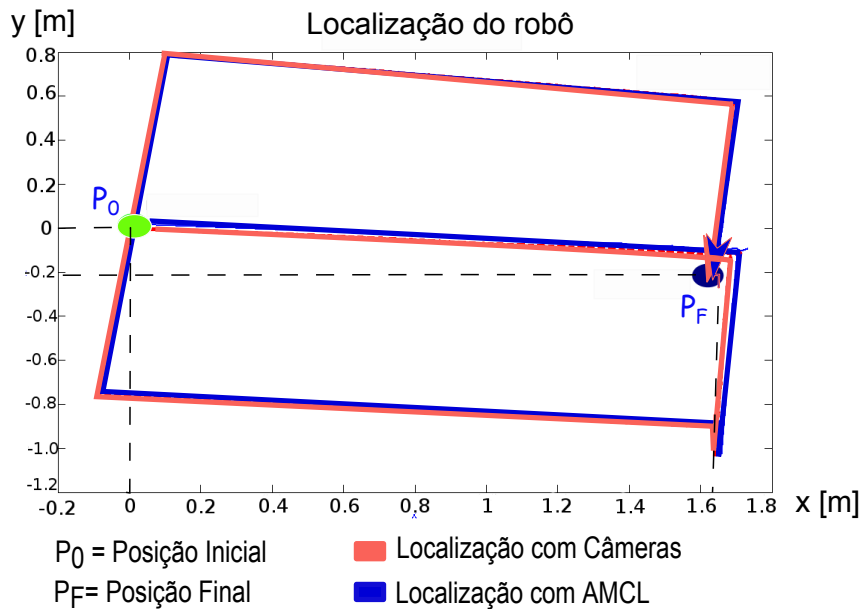
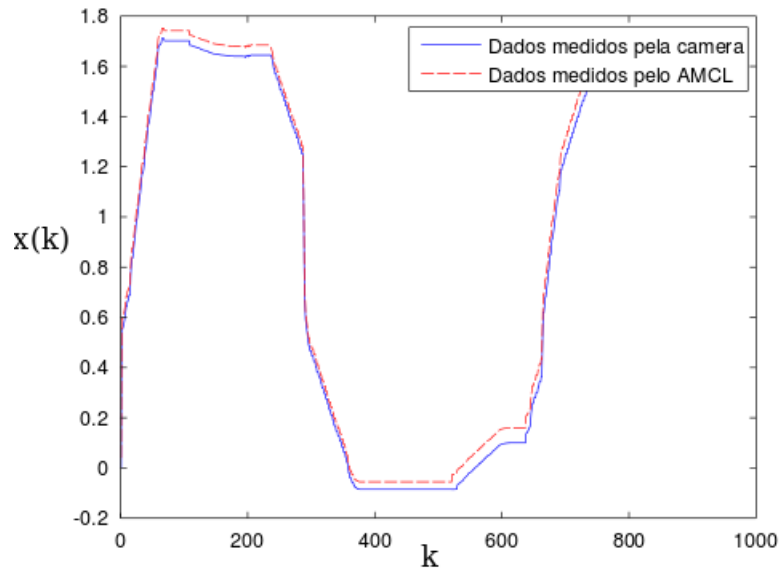
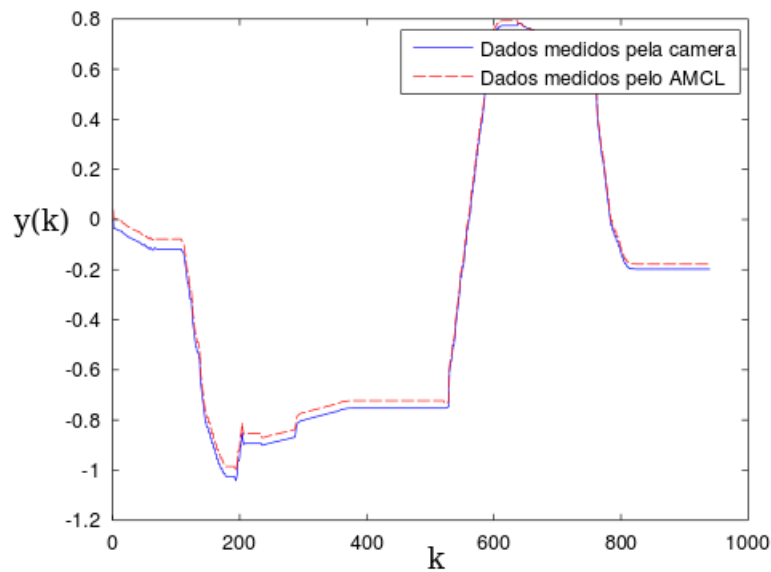


Figura 4.15: Experimento e comparação do Posicionamento do robô dentro do cenário II.

Observa-se na Figura 4.16 que o erro encontrado é consideravelmente pequeno, visto que os dados obtidos com o sistema de localização AMCL são aproximadamente iguais aos obtidos com as câmeras. O erro médio encontrado para o caminho seguido foi de $0,0145\sim\text{m}$. Isso demonstra que os procedimentos realizados e os algoritmos utilizados alcançaram os objetivos estabelecidos, ou seja, permitiram uma correta localização do robô dentro do ambiente avaliado.



(a)



(b)

Figura 4.16: Comparação entre as estimativas de posição baseadas no mapa obtido com AMCL e aquelas obtidas pelo sistema visual externo ao robô (a) na posição x ; (b) na posição y .

4.3 Resultados do Reconhecimento de Voz

Para validar o sistema de reconhecimento de voz realizou-se experimentos com um grupo de pessoas para encontrar a taxa de sucesso do módulo. O experimento teve como objetivo medir a capacidade de reconhecimento do módulo para cada instrução do usuário,

obtendo dessa forma uma taxa de sucesso geral do módulo.

Realizou-se o experimento com 10 pessoas, que foram denotadas com as letras *a* até *j*. Cada pessoa podia realizar até 5 tentativas de repetição em cada instrução para que o módulo possa reconhecer a instrução. Cada tentativa teve uma porcentagem de taxa de sucesso do módulo como é observado na Tabela 4.1. Ao incrementar o numero de tentativas a taxa de sucesso diminui. Portanto, na Tabela 4.2 observa-se a taxa de sucesso das pessoas com cada instrução e a taxa de sucesso do módulo de reconhecimento de voz bem como o erro na sua detecção.

Tabela 4.1: Taxa de sucesso do modulo de reconhecimento de voz de acordo com o numero de tentativas.

Numero de Tentativas	Taxa de Sucesso do Módulo de Reconhecimento de Voz
1	100%
2	75%
3	50%
4	25%
5	0%

Tabela 4.2: Resultados Experimentais no Reconhecimento de Voz.

Student	Kitchen	Bathroom	Bedroom	Living room	Dining room	Laundry room	Office	Média
a	100%	100%	100%	100%	100%	100%	100%	100%
b	50%	100%	100%	100%	100%	75%	25%	78.57%
c	100%	75%	75%	0%	25%	0%	0%	39.29%
d	75%	50%	100%	0%	25%	0%	0%	35.71%
e	100%	100%	100%	100%	100%	75%	100%	96.43%
f	100%	100%	100%	100%	100%	75%	100%	96.43%
g	100%	50%	100%	0%	50%	100%	75%	67.86%
h	100%	0%	75%	0%	100%	25%	75%	53.57%
i	100%	100%	75%	75%	100%	0%	100%	78.57%
j	75%	100%	100%	0%	100%	100%	0%	67.86%
Média	90%	77.5%	92.5%	47.5%	80%	55%	57.5%	
Taxa de Sucesso do módulo de reconhecimento de voz								71.43%

A taxa de sucesso do módulo encontrada verifica que o módulo funciona para a aplicação apresentada. O erro encontrado assume-se que tem relação com o sotaque da pessoa e o nível de voz, porque o módulo esta configurado para funcionar com a língua inglesa.

4.4 Resultados do Planejamento de Caminhos

Com o objetivo de avaliar a eficiência do algoritmo de planejamento de caminhos implementado neste trabalho, mostra-se nesta seção os resultados de aplicação desse algoritmo em vários ambientes simulados, observados nas Figuras 4.17-4.20, assim como também em ambientes reais como aqueles vistos nas Figuras 4.21 e 4.22. Na Figura 4.17 o mapa considerado para a simulação foi desenhado manualmente, enquanto que em 4.18, 4.19 e 4.20, os mapas utilizados são os resultados do mapeamento mostrados na seção anterior. Como observado em todas as Figuras mencionadas anteriormente, o algoritmo de planejamento criou e gerou a trajetória a ser seguida pelo robô até o ponto desejado e, o robô conseguiu executar a tarefa sem maiores inconvenientes.

Na Figura 4.19 foi utilizado o mapa projetado do cenário I da Seção 4 para planejar o caminho do robô até o ponto desejado. A fim de permitir uma comparação e verificar a importância desse mapa, na Figura 4.18 utilizou-se o mapa do mesmo ambiente mas gerado pelo laser para atingir ao mesmo objetivo. Isto permite mostrar que o planejamento realizado no mapa projetado permite evitar os obstáculos que o mapa realizado com o laser não conseguiu observar, como mostrado na Figura 4.18, onde a trajetória gerada passaria por onde existe um obstáculo (mesa), colidindo assim com o ambiente real.

Pode-se observar na Figura 4.20, o planejamento do caminho na simulação, realizado no mapa do ambiente real que gerou a trajetória desde a sala de Mestrado até a sala de Doutorado e, na Figura 4.21, a execução da trajetória no ambiente real do cenário I.

Os resultados da execução do experimento completo no ambiente real é apresentado na Figura 4.22. O experimento apresenta a sala de Mestrado como ponto inicial do robô e a sala de Doutorado como ponto final. Além de mostrar as etapas realizadas para a execução da navegação como: a estimação da posição do robô, a seleção de pontos referenciais no mapa do ambiente, o reconhecimento e verificação das instruções de voz, a execução da tarefa, o reconhecimento de obstáculos não mapeados, a ação de parada tomada pelo robô até que o obstáculo saia do caminho e, a finalização da tarefa ao chegar ao ponto desejado quando o obstáculo sai do caminho.



Figura 4.17: Execução do planejador de caminhos em simulação, utilizando um mapa desenhado similar a um ambiente doméstico. A tarefa foi ir de uma pose inicial até uma pose desejada, (a) trajetória completa mostrada no mapa dilatado; (b) robô saindo da pose inicial; (c), (d), (e), (f) e (g) robô executando a trajetória; (h) robô chegando a pose desejada.

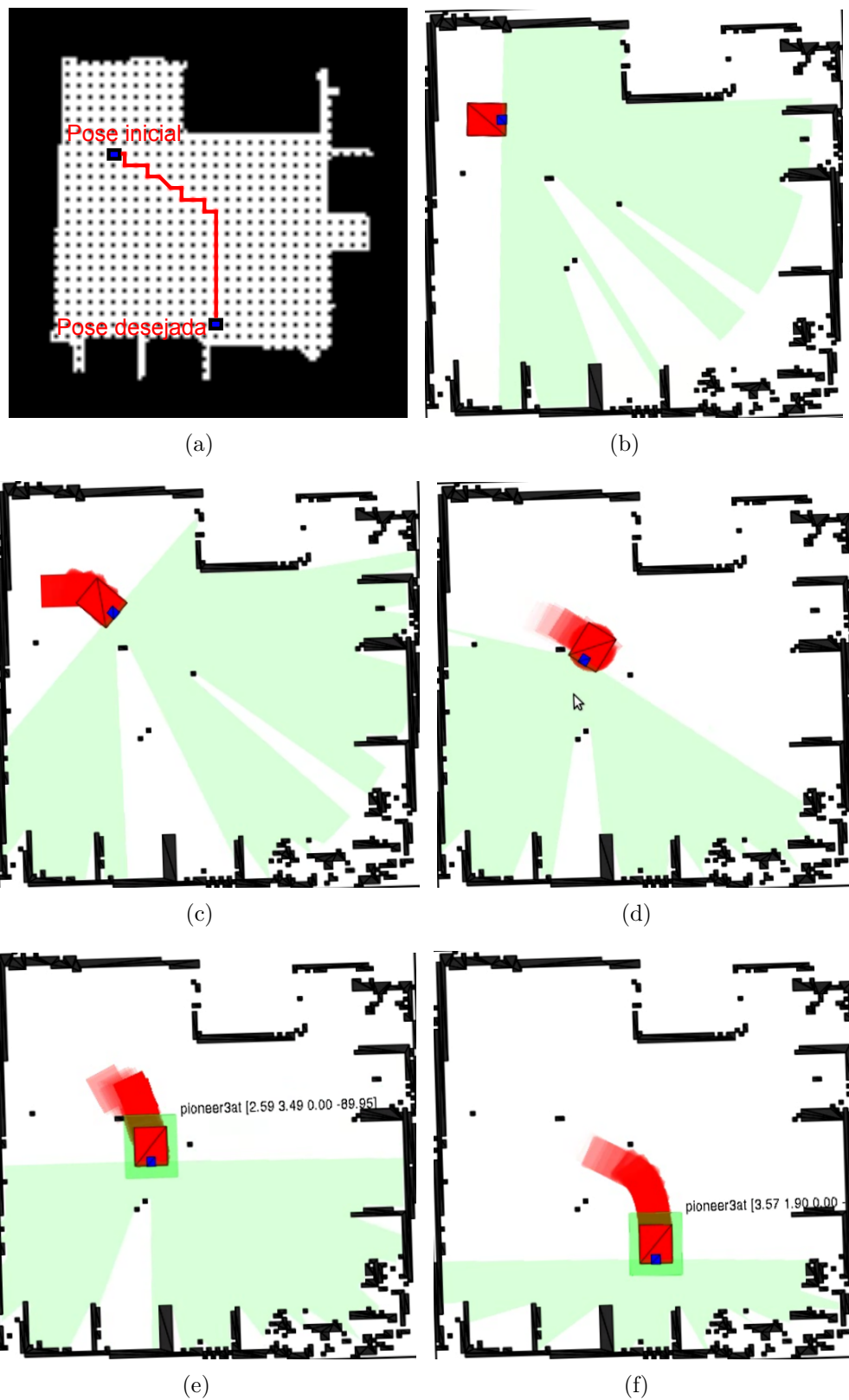


Figura 4.18: Execução do planejador de caminhos em simulação, utilizando o mapa 2D do cenário I gerado com laser que não considera os obstáculos até o tamanho do robô MARIA, (a) trajetória completa mostrada no mapa dilatado; (b) robô saindo da pose inicial; (c), (d) e (e) robô executando a trajetória; (f) robô chegando a pose desejada.

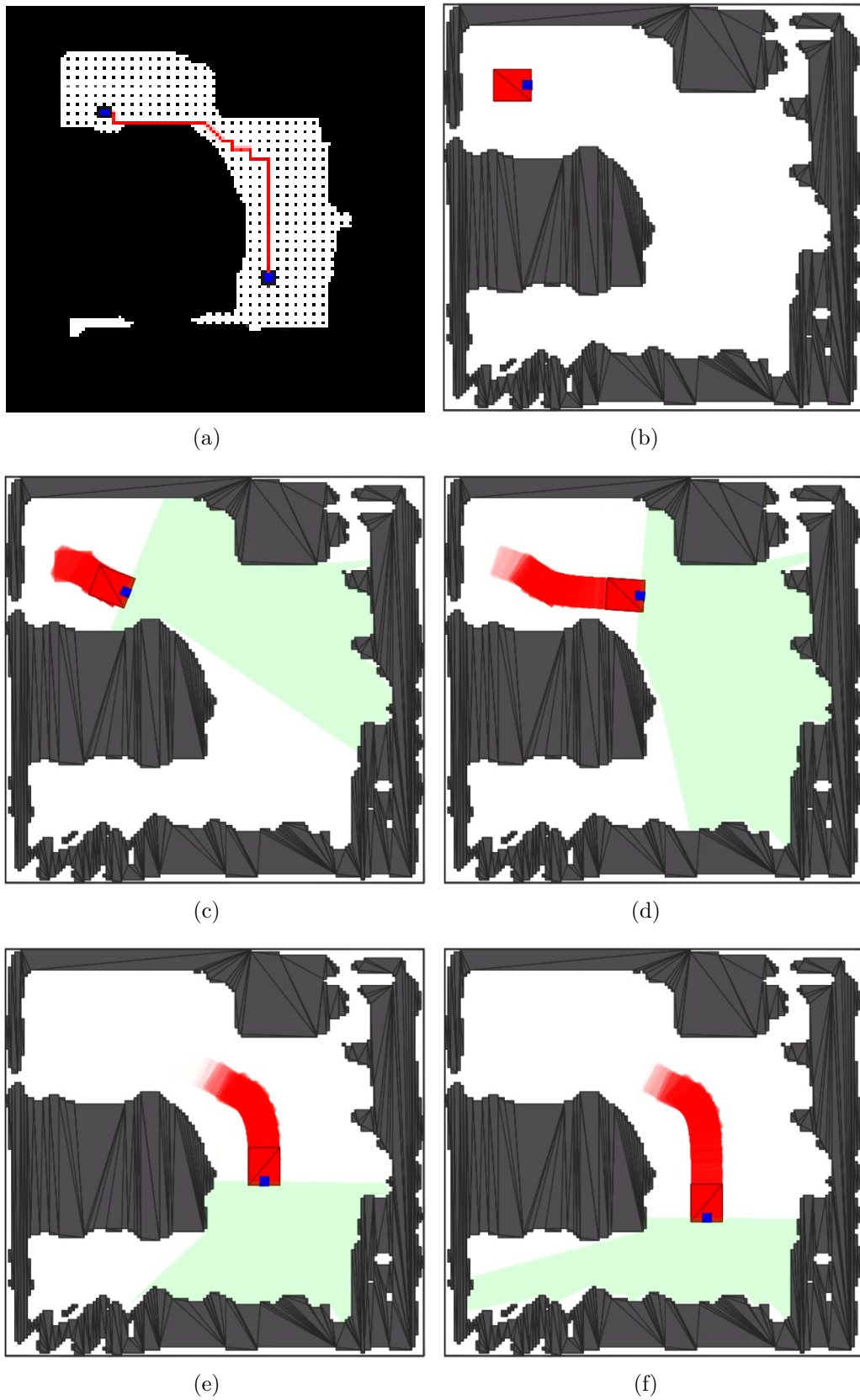


Figura 4.19: Execução do planejador de caminhos em simulação, utilizando o mapa 2D projetado do cenário I que considerando os obstáculos até o tamanho do robô MARIA, (a) trajetória completa mostrada no mapa dilatado; (b) robô saindo da pose inicial; (c), (d) e (e) robô executando a trajetória; (f) robô chegando a pose desejada.



(a)



(b)

(c)



(d)

(e)



(f)

(g)

Figura 4.20: Execução do planejador de caminhos em simulação, utilizando o mapa 2D projetado do cenário III, considerado completo pela sua complexidade. A tarefa foi ir de uma sala à outra: (a) trajetória completa; (b) robô saindo da pose inicial; (c), (d), (e) e (f) robô executando a trajetória; (g) robô chegando a pose desejada.

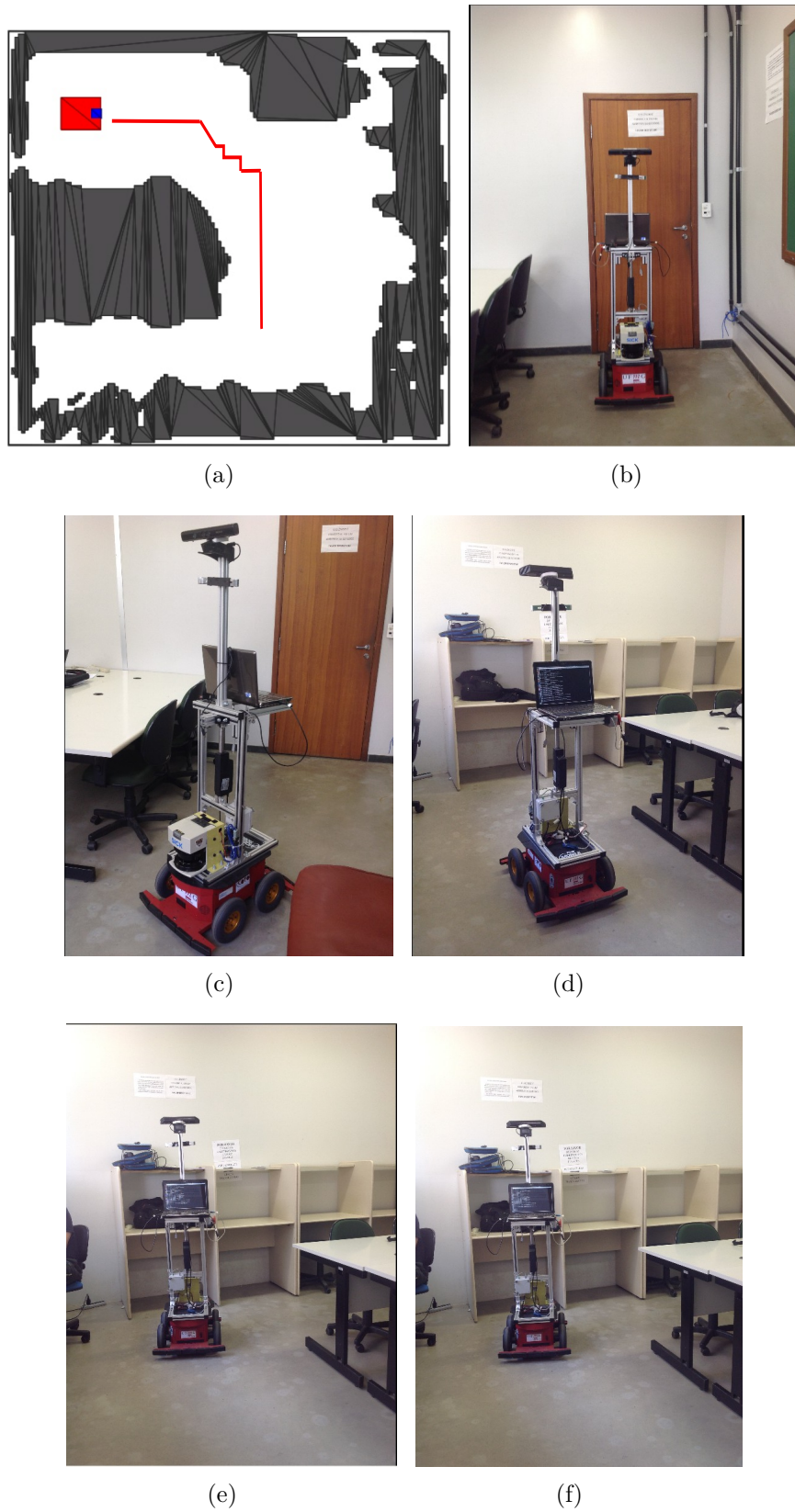


Figura 4.21: Execução do planejador de caminhos no ambiente real do cenário I pelo robô MARIA, mostrando o desvio de obstáculos detectados na projeção bidimensional (a) trajetória completa; (b) robô saindo da pose inicial; (c), (d) e (e) robô executando a trajetória; (f) robô chegando a pose desejada.

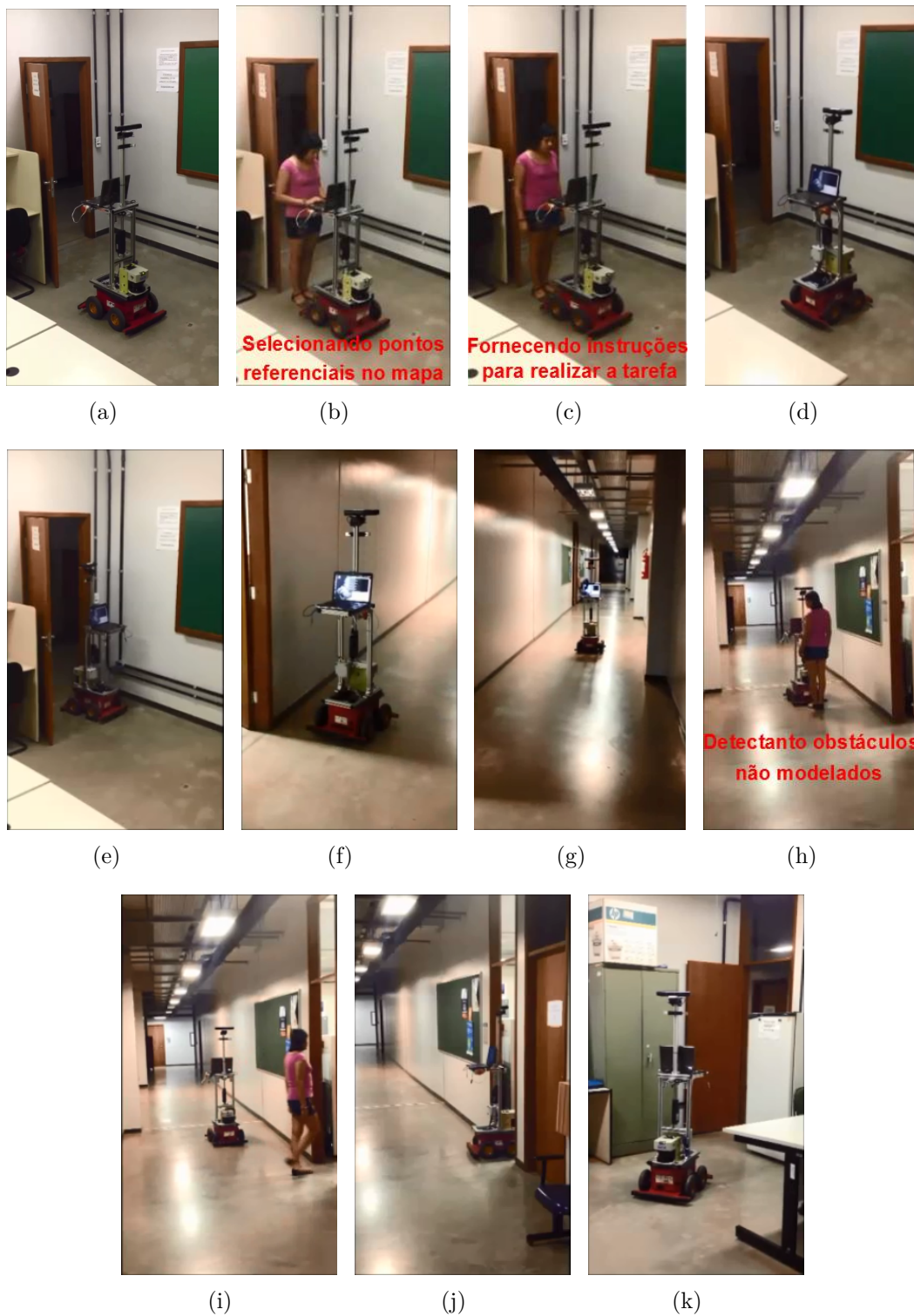


Figura 4.22: Sequência completa do experimento realizado no ambiente real: (a) robô localizando-se; (b) seleção de pontos referenciais no mapa do ambiente; (c) fornecendo instruções de voz para que realize a tarefa; (d) robô saindo da pose inicial; (e), (f) e (g) robô executando a trajetória; (h) detectando obstáculos não mapeados; (i) e (j) robô continuando com a tarefa; (k) robô chegando a pose desejada.

5 *Conclusões e Trabalhos Futuros*

Esta dissertação apresentou o desenvolvimento de um sistema de navegação em ambientes internos para um robô pessoal de grande altura chamado MARIA. Essa plataforma é composta por uma base móvel não holonômica Pioneer 3-AT, um sensor laser SICK LMS 291, um sensor Kinect, um tablet e um braço manipulador Jaco. O sistema de navegação desenvolvido permitiu ao robô deslocar-se de uma posição e orientação iniciais até uma posição e orientação finais dentro de um ambiente interno previamente mapeado, ao receber instruções de voz. A metodologia envolveu: (i) a geração de mapas do ambiente mediante a aquisição dos dados sensoriais de um sensor Kinect e um sensor de proximidade a laser, (ii) a estimativa da posição no ambiente mediante o algoritmo de localização Monte Carlo (AMCL), (iii) a implementação de um planejador de caminhos por frentes de ondas, (iv) a geração e a execução de rotas livres de colisão e, (v) o reconhecimento e verificação de instruções de voz.

O mapa bidimensional gerado pelo laser foi utilizado para estimar a posição do robô dentro do ambiente, enquanto que o mapa tridimensional gerado com o uso do Kinect foi utilizado para realizar uma projeção bidimensional dos obstáculos que se encontraram no ambiente até o tamanho do robô, para ser utilizada na etapa da implementação do algoritmo de planejamento de caminhos. A metodologia manteve a ideia de que a forma mais eficiente de se navegar manipuladores móveis é utilizar mapas bidimensionais (2D) que tenham em consideração os obstáculos de grande altura presentes no caminho do robô e que, os mapas tridimensionais sejam usados apenas quando o robô precisa realizar uma tarefa de detecção e manipulação de objetos.

A metodologia desta pesquisa foi validada a partir de experimentos realizados em um ambiente de simulação e na plataforma robótica MARIA. Esses mostraram que os mapas obtidos foram coerentes com os respectivos ambientes e cenários de testes e que o algoritmo AMCL estimou a pose do robô bem próxima da real, mas que ainda mantém pequenos erros que dependendo da precisão com que se deseje que trabalhe o robô seria considerado um problema. O algoritmo de planejamento de caminhos respondeu de forma

satisfatória para encontrar uma trajetória até o ponto desejado e, o manipulador móvel respondeu a comandos de voz para a execução da tarefa de forma efetiva. Além disso, o manipulador móvel detectou a presença de objetos não modelados que se encontravam na trajetória que o robô seguia para chegar ao ponto desejado. Por meio de comandos de voz, o robô solicitava ao objeto que se encontrava na rota que desobstruísse o caminho.

A proposta apresentada nesta dissertação mostra uma metodologia de integração diferente para a resolução do sistema de navegação se comparada com os trabalhos existentes na literatura. Este trabalho foi inspirado na competição RoboCup@Home que visa o desenvolvimento de sistemas para interação humano-robô. Portanto, visou-se o desenvolvimento deste sistema para que o robô possa se locomover dentro de ambientes internos. Embora neste trabalho não tenha sido realizado a manipulação dos objetos propriamente dita, nada impede que um módulo de manipulação possa ser acoplado futuramente.

Um aspecto interessante que pode ser observado no desenvolvimento do trabalho, refere-se ao baixo custo computacional demandado para a geração dos mapas, o que tornou o processo bastante viável. Percebe-se ainda que a representação obtida pelos mapas, foi muito próxima da visão humana. Convém salientar que, devido ao ângulo limitado de visão do sensor Kinect, existe uma dificuldade de detectar o entorno a um ângulo de abertura horizontal maior de 57° e vertical de 43° . Assim, em ambientes com grandes dimensões o robô poderia ter inconvenientes para realizar o mapeamento. Contudo, com o uso do laser esse problema é minimizado, visto que consegue-se construir um mapa bidimensional com um alcance maior. De igual forma, o campo de visão horizontal é mais amplo, o que acaba contornando as dificuldades que o Kinect apresenta para ambientes longos.

O principal objetivo da obtenção dos mapas é permitir que o robô consiga alcançar autonomia no seu ambiente de trabalho. Partindo desses mapas, conseguiu-se definir a localização e estrutura dos obstáculos, o que possibilitou programar ao robô de forma que o mesmo possa localizar-se no cenário em questão, planejar sua trajetória, navegar sem colidir com obstáculos e explorar o ambiente como um todo.

Outro ponto a ser considerado é o tempo que o algoritmo de planejamento de caminhos implementado leva para rotular o mapa todo, pois este cresce exponencialmente com o tamanho do mapa. Em mapas de grande dimensão poderia se levar um grande tempo para rotular, o que impediria a execução da tarefa com rapidez pelo robô. Uma solução viável é modificar o tamanho da célula na grade de ocupação dependendo do mapa a ser usado. Contudo, uma vez rotulado o algoritmo de frentes de ondas retorna uma solução

se essa existir em um tempo finito ou caso contrario indica que não existe uma solução.

Como trabalhos futuros sugeresse a melhoria do algoritmo de estimação da pose do robô dentro de ambientes com tendencia a maiores mudanças. Além de se realizar a busca e manipulação de objetos dentro do ambiente para que a plataforma robótica possa ser usada para ajudar aos seres humanos.

Referências

- [Abdelkrim and Issam, 2014] Abdelkrim, N. and Issam, K. (2014). Fuzzy logic controllers for Mobile robot navigation in unknown environment using Kinect sensor. *21th International Conference on Systems, Signals and Image Processing (IWSSIP)*, (May).
- [Adorno, 2008] Adorno, B. V. (2008). Planejamento Probabilístico de Rotas no Espaço de Configuração e sua Aplicação em Robótica Móvel. Master’s thesis, Universidade de Brasília.
- [Ahn et al., 2007] Ahn, S., Lee, K., Chung, W. K., and Oh, S. R. (2007). SLAM with visual plane: Extracting vertical plane by fusing stereo vision and ultrasonic sensor for indoor environment. In *IEEE International Conference on Robotics and Automation*, pages 4787–4794.
- [Albán-Peñañiel and Pereira, 2014] Albán-Peñañiel, D. and Pereira, G. A. S. (2014). Geração de mapas para localização e navegação de um manipulador móvel usando múltiplos sensores. In *XX Congresso Brasileiro de Automática 2014 (CBA 2014)*, pages 1–8, Belo Horizonte, MG, Brazil.
- [Barraquand and Latombe, 1991] Barraquand, J. and Latombe, J.-C. (1991). Robot Motion Planning: A Distributed Representation Approach.
- [Barrera, 2010] Barrera, A. (2010). *Mobile Robots Navigation*. [S.l.]: In-Teh, 2010.
- [Bay et al., 2006] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded up robust features. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3951 LNCS, pages 404–417.
- [Burgard et al., 1998] Burgard, W., Cremers, A., and Fox, D. (1998). The interactive museum tour-guide robot. *15th Conference on Artificial intelligence/Innovative applications of artificial intelligence AAAI*, pages 11–18.
- [Choset, 2005] Choset, H. M. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementation*.
- [Chum et al., 2003] Chum, O., Matas, J., and Kittler, J. (2003). Locally Optimized RANSAC. *Annual Symposium of the German Pattern Recognition Society (DAGM)*, 2781:236–243.
- [Chung et al., 2004] Chung, W., Kim, G., Kim, M., and Lee, C. (2004). Integrated navigation system for indoor service robots in large-scale environments. *IEEE International Conference on Robotics and Automation*, (April).

- [Correa and Sciotti, 2012] Correa, D. and Sciotti, D. (2012). Mobile robots navigation in indoor environments using kinect sensor. *2th Brazilian Conference on Critical Embedded Systems*, pages 36–41.
- [de Souza, 2002] de Souza, S. C. B. (2002). Planejamento de Trajetória para um Robô Móvel com Duas Rodas Utilizando um Algoritmo A-Estrela Modificado. Master’s thesis, Universidade Federal do Rio de Janeiro.
- [Dryanovski and Morris, 2010] Dryanovski, I. and Morris, W. (2010). Multi-volume occupancy grids: An efficient probabilistic 3D mapping model for micro aerial vehicles. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, (1):1553–1559.
- [Durrant-Whyte et al., 2003] Durrant-Whyte, H., Majumder, S., Thrun, S., de Battista, M., and Scheduling, S. (2003). A Bayesian Algorithm for Simultaneous Localisation and Map Building.
- [Eder, 2003] Eder, R. (2003). Navegação Em Ambientes Semiestruturados Por Um Sistema Robótico Autônomo Em Tempo Real. Master’s thesis, IME, Rio de Janeiro.
- [Fu et al., 2009] Fu, S., Hou, Z., and Yang, G. (2009). An indoor navigation system for autonomous mobile robot using wireless sensor network. *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pages 227–232.
- [Gallegos and Rives, 2010] Gallegos, G. and Rives, P. (2010). Indoor SLAM based on composite sensor mixing laser scans and omnidirectional images. In *IEEE International Conference on Robotics and Automation*, pages 3519–3524.
- [Gockley R, 2006] Gockley R, M. M. J. (2006). Encouraging physical therapy compliance with a hands-off mobile robot. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 150–155.
- [Goldstein et al., 1987] Goldstein, M., Pin, F., Saussure, G., and Weisbin, C. (1987). 3-D world modeling based on combinatorial geometry for autonomous robot navigation. *IEEE International Conference on Robotics and Automation*.
- [Gross et al., 2012] Gross, H. M. A., Schroeter, C. A., Mueller, S. A., Volkhardt, M. A., Einhorn, E. A., Bley, A. B., Langner, T. B., Merten, M. B., Huijnen, C. C., Van Den Heuvel, H. C., and Van Berlo, A. C. (2012). Further progress towards a home robot companion for people with mild cognitive impairment. In *IEEE International Conference on Systems Man and Cybernetics*, pages 637–644.
- [Hata, 2010] Hata, A. (2010). Mapeamento de ambientes externos utilizando robôs móveis. Master’s thesis, Universidade de São Paulo.
- [Hayashi, 2007] Hayashi, E. (2007). A navigation system with a self-drive control for an autonomous robot in an indoor environment. In *16th IEEE International Conference on Robot and Human Interactive Communication*, pages 246–251.
- [Hibino et al., 2010] Hibino, F., Takahashi, Y., Yoshida, K., and Maeda, Y. (2010). Human Pointing Navigation of Mobile Robot with Spherical Vision System. *Joint 5th International Conference on Soft Computing and Intelligent Systems and 11th International Symposium on Advanced Intelligent Systems SCIS & ISIS*, pages 217–221.

- [Hiemustra and Nederveen, 2007] Hiemustra, P. and Nederveen, A. (2007). Monte Carlo localization. *Ad Hoc Networks*, 6(5):718–733.
- [Hirose and Chugo, 2011] Hirose, K. and Chugo, D. (2011). Service robots navigation using pictographs detection for indoor environment. *37th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 2170–2175.
- [Hornung et al., 2012] Hornung, A., Phillips, M., Gil Jones, E., Bennewitz, M., Likhachev, M., and Chitta, S. (2012). Navigation in three-dimensional cluttered environments for mobile manipulation. *IEEE International Conference on Robotics and Automation*, pages 423–429.
- [Hornung et al., 2013] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.
- [Isozaki and Chugo, 2011] Isozaki, N. and Chugo, D. (2011). Camera-based AGV navigation system for indoor environment with occlusion condition. *IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 778–783.
- [Jahanbin, 1988] Jahanbin, M.R., F. F. (1988). Path Planning Using a Wave Simulation Technique in the Configuration Space. *GERO, J.S. Artificial intelligence in engineering: robotics and process. Amsterdam: Elsevier.*
- [Jeong et al., 2009] Jeong, H., Kim, S., and Kwak, Y. (2009). Development of autonomous robot system for indoor messy environments. *International Joint Conference (ICCAS-SICE)*, pages 3409–3413.
- [Kavraki et al., 1996] Kavraki, L. E., Švestka, P., Latombe, J. C., and Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580.
- [Kriegman et al., 1987] Kriegman, D., Triendl, E., and Binford, T. (1987). A mobile robot: Sensing, planning and locomotion. *IEEE International Conference on Robotics and Automation*.
- [Kuipers B., 1988] Kuipers B. (1988). A robust, qualitative method for spatial learning in unknown environments.
- [Kwang-Hyun et al., 2008] Kwang-Hyun, P. K.-H. P., Hyong-Euk, L. H.-E. L., Youngmin, K. Y. K., and Bien, Z. (2008). A Steward Robot for Human-Friendly Human-Machine Interaction in a Smart House Environment. *IEEE Transactions on Automation Science and Engineering*, 5(1):21–25.
- [Kwon et al., 2006] Kwon, J.-W., Yang, D.-H., Chwa, D., and Hong, S.-K. (2006). Wavefront method-based local-path planning for a mobile robot with a vision system. In *SICE-ICASE, 2006. International Joint Conference*, pages 1250–1254.
- [LaValle, 2006] LaValle, S. M. (2006). Planning Algorithms. *Methods*, 2006:842.

- [Lichtenstern et al., 2013] Lichtenstern, M., Angermann, M., Frassl, M., Berthold, G., Julian, B. J., and Rus, D. (2013). Pose and Paste - An Intuitive Interface for Remote Navigation of a Multi-Robot System. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1632–1639.
- [Lv et al., 2008] Lv, X., Zhang, M., and Li, H. (2008). Robot Control Based on Voice Command. *International Conference on Automation and Logistics*, pages 2490–2494.
- [Marder-Eppstein et al., 2010] Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., and Konolige, K. (2010). The Office Marathon: Robust navigation in an indoor office environment. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 300–307.
- [Marks et al., 2009] Marks, T. K., Howard, A., Bajracharya, M., Cottrell, G. W., and Matthies, L. H. (2009). Gamma-SLAM: Visual SLAM in unstructured environments using variance grid maps. *Journal of Field Robotics*, 26(1):26–51.
- [Merat and Wu, 1987] Merat, F. and Wu, H. W. H. (1987). Generation of object descriptions from range data using feature extraction by demands. *IEEE International Conference on Robotics and Automation*.
- [Mohamed and Capi, 2012] Mohamed, Z. and Capi, G. (2012). Development of a New Mobile Humanoid Robot for Assisting Elderly People. In *International Symposium on Robotics and Intelligent Sensors*, volume 41, pages 345–351.
- [Moravec and Elfes, 1985] Moravec, H. and Elfes, A. (1985). High resolution maps from wide angle sonar. *Proceedings. 1985 IEEE International Conference on Robotics and Automation*.
- [Murphy, 2000] Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press, Cambridge, MA, USA, 1st edition.
- [Okumura et al., 2010] Okumura, J., Takei, T., and Tsubouchi, T. (2010). Navigation in indoor environment by an autonomous unicycle robot with wide-type wheel. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 154–159.
- [Ortiz, 2013] Ortiz, D. (2013). Navegação autônoma de robôs móveis e detecção de intrusos em ambientes internos utilizando sensores 2D e 3D. Master’s thesis, Universidade de São Paulo.
- [Ottoni, 2000] Ottoni, G. L. (2000). Planejamento De Trajetórias Para Robôs Móveis. *Curso de Engenharia de Computação, Fundação Universidade Federal do Rio Grande*.
- [Paredes, 2014] Paredes, A. C. (2014). Signage system for the navigation of autonomous robots in indoor environments. *IEEE Transactions on Industrial Informatics*, 10(1):680–688.
- [Pereira, 2003] Pereira, J. (2003). Avaliação e Correção do Modelo Cinemático de Robôs Móveis Visando a Redução de Erros no Seguimento de Trajetórias. Master’s thesis, Universidade do Estado de Santa Catarina.

- [Pimenta et al., 2006] Pimenta, L., Fonseca, A., Pereira, G., Mesquita, R., Silva, E., Caminhas, W., and Campos, M. (2006). Robot navigation based on electrostatic field computation. *IEEE Transactions on Magnetics*, 42(4).
- [Pons, 2012] Pons, J. V. N. (2012). Localización y generación de mapas del entorno (SLAM) de un robot por medio de una Kinect.
- [Prado, 2013] Prado, M. (2013). Planejamento de trajetória para estacionamento de veículos autônomos. Master’s thesis, Universidade de São Paulo.
- [Ruhnke et al., 2011] Ruhnke, M., Kümmerle, R., Grisetti, G., and Burgard, W. (2011). Highly accurate maximum likelihood laser mapping by jointly optimizing laser points and robot poses. In *IEEE International Conference on Robotics and Automation*, pages 2812–2817.
- [Russell and Norvig, 2003] Russell, S. J. and Norvig, P. (2003). Artificial Intelligence: A Modern Approach. *second edition, Prentice Hall, Upper Saddle River, New Jersey*.
- [Santana et al., 2008] Santana, P., Cândido, C., Santos, P., Almeida, L., and Barata, J. (2008). The ares robot: case study of an affordable service robot. In *European Robotics Symposium*.
- [Sato et al., 2011] Sato, N., Kon, K., and Matsuno, F. (2011). Navigation Interface for Multiple Autonomous Mobile Robots with Grouping Function. *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 32–37.
- [Sato et al., 2008] Sato, N., Mizumoto, H., Shiroma, N., Inami, M., and Matsuno, F. (2008). Touch-Pen Interface with Local Environment Map for Mobile Robot Navigation. *Annual Conference SICE*, pages 1632–1637.
- [Segal et al., 2009] Segal, A., Haehnel, D., and Thrun, S. (2009). Generalized-ICP. In *Robotics: Science and Systems*.
- [Seo and Kim, 2013] Seo, D. J. and Kim, J. (2013). Development of autonomous navigation system for an indoor service robot application. In *13th International Conference on Control, Automation and Systems (ICCAS)*, pages 204–206.
- [Shim et al., 2010] Shim, B., Kang, K.-w., and Lee, W. (2010). An Intelligent Control of Mobile Robot Based on Voice Command. *International Conference on Control, Automation and Systems*, pages 2107–2110.
- [Souza, 2012] Souza, A. d. S. (2012). *Mapeamento robótico 2, 5-D com representação em grade de ocupação-elevação*. PhD thesis, Universidade Federal do Rio Grande do Norte.
- [Surmann et al., 2003] Surmann, H., Nüchter, A., and Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198.
- [Thrun, 2002] Thrun, S. (2002). Robotic Mapping: A Survey. *Science*, 298(February):1–35.

- [Thrun et al., 1999] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). MINERVA: a second-generation museum tour-guide robot. *IEEE International Conference on Robotics and Automation*, 3:1999–2005.
- [Thrun et al., 2005] Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. Number 3. Cambridge, MA, USA: The MIT Press.
- [Vincent, 1987] Vincent, A. d. S. (1987). Visual navigation for a mobile robot: Building a map of the occupied space from sparse 3-D stereo data. *IEEE International Conference on Robotics and Automation*.
- [Wen et al., 2014] Wen, C., Qin, L., Zhu, Q., Wang, C., and Li, J. (2014). Three-Dimensional Indoor Mobile Mapping With Fusion of Two-Dimensional Laser Scanner and RGB-D Camera Data. *IEEE Geoscience and Remote Sensing Letters*, 11(4):843–847.
- [Yu and Nawroj, 2012] Yu, Y. and Nawroj, A. (2012). Mobile Robot Navigation Through a Brain Computer Interface. *IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–4.
- [Zaman et al., 2011] Zaman, S., Slany, W., and Steinbauer, G. (2011). ROS-based mapping, localization and autonomous navigation using a pioneer 3-DX robot and their relevant issues. *Electronics, Communications and Photonics Conference (SIECPC)*, pages 1–4.
- [Zhixiao Yang, 2006] Zhixiao Yang, Sumin Jiao, D. Z. A. G. (2006). A Novel Graphic Interface Design for the Navigation of Remotely Controlled Rescue Robot. *IEEE International Conference on Robotics and Biomimetics*, pages 589–594.