



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E COMPUTAÇÃO



# **Localização e Mapeamento Simultâneos de Ambientes Planos Usando Visão Monocular e Representação Híbrida do Ambiente.**

**André Macêdo Santana**

Orientador: Prof. Dr. Adelardo Adelino Dantas de Medeiros

**Tese de Doutorado** apresentada ao Programa de Pós Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Doutor em Ciências.

Natal-RN, fevereiro de 2011

Seção de Informação e Referência  
Catalogação da Publicação na Fonte. UFRN / Biblioteca Central Zila Mamede

Santana, André Macedo.

Localização e mapeamento simultâneos de ambientes planos usando visão monociliar e representação híbrida do ambiente / André Macêdo Santana. – Natal, RN, 2010.

151 f. : il.

Orientador: Adelardo Adelino Dantas de Medeiros.

Tese (doutorado) – Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. SLAM – Tese. 2. Visão Monocular – Tese. 3. Grade de Ocupação Visual – Tese. 4. Mapa Híbrido – Tese. 5. Filtro de Kalman – Tese. I. Medeiros, Adelardo Adelino Dantas de. II. Universidade Federal do Rio Grande do Norte. III. Título.

**Localização e Mapeamento Simultâneos de Ambientes Planos Usando Visão  
Monocular e Representação Híbrida do Ambiente.**  
**Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN**

**André Macêdo Santana**

Tese de Doutorado aprovada em 11 de fevereiro de 2011 pela banca examinadora composta pelos seguintes membros:

---

Dr. Adelardo Adelino Dantas de Medeiros (orientador) ..... UFRN

---

Dra. Silvia Silva da Costa Botelho (examinadora externa) ..... FURG

---

Dra. Anna Helena Reali Costa (examinadora externa) ..... USP

---

Dr. Adrião Duarte Dória Neto (examinador interno) ..... UFRN

---

Dr. Pablo Javier Alsina (examinador interno) ..... UFRN

---

Dr. Diogo Pinheiro Fernandes Pedrosa ..... UFRN

*A vida só pode ser compreendida  
olhando-se para trás; mas só pode  
ser vivida olhando-se para a frente.*

---

# Agradecimentos

---

O primeiro gesto é de agradecimento a Deus - do qual todas as coisas dependem - e que, com a sua infinita bondade, concedeu-me a oportunidade de, na aspérrima caminhada, experimentar a inefável sensação de galgar um patamar a mais.

Depois, o agradecimento aos meus pais, Antônio e Vera que, na segurança de um lar bem estruturado, plasmaram a minha personalidade e fincaram o balizamento do meu caráter.

Logo a seguir, acode-me a imagem querida da minha doce companheira, Carol, sempre acreditando em mim, impelindo-me a constantes buscas de aperfeiçoamento, inculcandom-me o salutar desejo de superar as minhas próprias e humanas limitações.

Minhas irmãs, Janaína e Sâmia também são peças fundamentais para esta conquista.

É impossível não agradecer de forma veemente a pessoa do professor Adelardo Medeiros, meu orientador. Obrigado pelo apoio irrestrito, pela paciência, pelos incentivos diversos, pelos inúmeros conselhos e ensinamentos e, pela confiança no meu trabalho.

Muitas outras pessoas também merecem meus agradecimentos: meus amigos do Departamento de Informática e Estatística da UFPI; meu grande amigo de convivência diária Anderson Abner; o professor Pablo Alsina que costumo dizer que foi meu "co-orientador"; professor Luiz Affonso pelos incentivos e longos "papos".

Enfim, a TODOS os colegas do Laboratório de Robótica, batizado carinhosamente por nós de "LAR", que contribuíram direta ou indiretamente para esta conquista.

Obrigado a todos!!

---

# Resumo

---

O objetivo desta tese é apresentar uma técnica de SLAM (Localização e Mapeamento Simultâneos) adequada para ambientes planos com linhas presentes no chão, de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de visão monociliar. Inicialmente, é feita uma etapa de segmentação para classificar as partes da imagem em “chão” e “não-chão”. Em seguida, o processamento de imagem identifica linhas na parte “chão” e os parâmetros dessas linhas são mapeados para o mundo, usando uma matriz de homografia. Finalmente, as linhas identificadas são usadas como marcos no SLAM, para construir um mapa de características.

Em paralelo, a pose corrigida do robô, a incerteza em relação à pose e a parte “não-chão” da imagem são usadas para construir uma grade de ocupação, gerando um mapa métrico com descrição dos obstáculos.

A utilização simultânea dos dois tipos de mapa obtidos (métrico em grade e de características) dá maior autonomia ao robô, permitindo acrescentar tarefas de planejamento em simultâneo com a localização e mapeamento.

Resultados práticos são apresentados para validar a proposta.

**Palavras-chave:** *SLAM, Visão Monocular, Grade de Ocupação Visual, Mapa Híbrido, Filtro de Kalman.*



---

# Abstract

---

The goal of this work is to propose a SLAM (Simultaneous Localization and Mapping) solution based on Extended Kalman Filter (EKF) in order to make possible a robot navigates along the environment using information from odometry and pre-existing lines on the floor. Initially, a segmentation step is necessary to classify parts of the image in “floor” or “non floor”. Then the image processing identifies floor lines and the parameters of these lines are mapped to world using a homography matrix. Finally, the identified lines are used in SLAM as landmarks in order to build a feature map.

In parallel, using the corrected robot pose, the uncertainty about the pose and also the part “non floor” of the image, it is possible to build an occupancy grid map and generate a metric map with the obstacle’s description.

A greater autonomy for the robot is attained by using the two types of obtained map (the metric map and the features map). Thus, it is possible to run path planning tasks in parallel with localization and mapping.

Practical results are presented to validate the proposal.

**Keywords:** *SLAM, Monocular Vision, Occupancy Grid Maps, Hybrid Map, Extended Kalman Filter.*



---

# Sumário

---

<b>Sumário</b>	<b>8</b>
<b>Lista de Figuras</b>	<b>11</b>
<b>Lista de Tabelas</b>	<b>15</b>
<b>Lista de Algoritmos</b>	<b>17</b>
<b>1 Introdução</b>	<b>19</b>
1.1 Localização . . . . .	20
1.2 Mapeamento . . . . .	21
1.3 SLAM . . . . .	22
1.4 Motivação e Justificativa . . . . .	26
1.5 Objetivo . . . . .	27
1.6 Organização do trabalho . . . . .	29
<b>2 Localização e Mapeamento Simultâneos</b>	<b>31</b>
2.1 Correlações e incertezas . . . . .	31
2.2 Soluções para SLAM . . . . .	33
2.3 Filtro de Kalman . . . . .	35
2.4 Formulação Probabilística do SLAM . . . . .	38
2.5 Evolução dos trabalhos em SLAM . . . . .	43
2.6 Resumo do Capítulo . . . . .	45
<b>3 SLAM Visual</b>	<b>47</b>
3.1 Navegação Visual . . . . .	47
3.2 Representações de ambientes . . . . .	50
3.3 Trabalhos em SLAM visual . . . . .	55
3.4 Resumo do Capítulo . . . . .	60
<b>4 Sistema Proposto</b>	<b>61</b>
4.1 Visão Geral . . . . .	61
4.2 Trabalhos Relacionados . . . . .	63
4.3 Fundamentação Teórica . . . . .	67
4.4 Modelagem . . . . .	69
4.5 Processamento de Imagem . . . . .	74
4.6 Resumo do Capítulo . . . . .	77

<b>5 Resultados SLAM Visual</b>	<b>79</b>
5.1 Cenário A: Linhas Paralelas . . . . .	79
5.2 Cenário B: Linhas Concorrentes . . . . .	84
5.3 Cenário C: Fechamento de laço . . . . .	88
5.4 Cenário D: Ambiente externo . . . . .	92
5.5 Resultados Comparativos . . . . .	95
5.6 Conclusões . . . . .	96
<b>6 Grade de Ocupação Visual</b>	<b>99</b>
6.1 Grade de Ocupação . . . . .	99
6.2 Trabalhos Relacionados . . . . .	102
6.3 Sistema Proposto . . . . .	105
6.4 Resultados experimentais . . . . .	114
<b>7 Conclusões e Perspectivas</b>	<b>121</b>
<b>8 Produção Científica</b>	<b>123</b>
<b>Referências bibliográficas</b>	<b>125</b>
<b>A Modelagem do Sensor</b>	<b>143</b>

---

# **Lista de Figuras**

---

1.1	Etapas de um processo de navegação robótica. . . . .	20
1.2	Etapa de um processo de localização baseado em marcos. . . . .	21
1.3	Processo de construção de um mapa. . . . .	22
1.4	Problema de SLAM. . . . .	23
1.5	Esquema simplificado de SLAM. . . . .	25
1.6	Esquema simplificado do sistema proposto nesta Tese. . . . .	28
2.1	Sensores normalmente utilizados em SLAM. . . . .	33
2.2	Soluções para SLAM. . . . .	35
2.3	Resumo esquemático do Capítulo 2 . . . . .	46
3.1	Sistemas visuais de percepção. . . . .	49
3.2	Representações do ambiente . . . . .	50
3.3	Construção de um mapa topológico. . . . .	51
3.4	Representação de um feixe de sonar em uma grade de ocupação. . . . .	52
3.5	Principais tipos de características extraídas em imagens. . . . .	54
3.6	Resumo esquemático do Capítulo 3 . . . . .	60
4.1	Esquema do SLAM visual proposto. . . . .	62
4.2	Transformada de Hough. . . . .	68
4.3	Variáveis do modelo geométrico do robô. . . . .	69
4.4	Sistemas de coordenadas Fixo ( <i>F</i> ) e Móvel ( <i>M</i> ). . . . .	73
4.5	Imagen do piso do ambiente interno . . . . .	75
4.6	Imagen do piso do ambiente externo . . . . .	75
4.7	Filtragem das linhas . . . . .	76
4.8	Segmentação e filtragem das linhas . . . . .	76
4.9	Cálculo da Homografia: (a) Padrão, (b) Detecção automática dos pontos. . . . .	77
5.1	Cenário A: LAUT-UFRN . . . . .	79
5.2	Descrição do ambiente - Cenário A . . . . .	80
5.3	Cálculo da Homografia - Cenário A . . . . .	80
5.4	Imagen do piso - Cenário A . . . . .	81
5.5	Falha no processamento de imagem - Cenário A . . . . .	81
5.6	Sequencia de identificação das retas - Cenário A . . . . .	82
5.7	Comportamento da variância das linhas - Cenário A . . . . .	83
5.8	Comportamento da variância da pose do robô - Cenário A . . . . .	83
5.9	Cenário B: DCA-UFRN . . . . .	84

5.10	Descrição do ambiente - Cenário B . . . . .	85
5.11	Cálculo da homografia - Cenário B . . . . .	85
5.12	Imagen do piso - Cenário B . . . . .	86
5.13	Influência da iluminação - Cenário B . . . . .	86
5.14	Comportamento da variância da pose do robô - Cenário B . . . . .	87
5.15	Cenário C: DFTE-UFRN . . . . .	88
5.16	Descrição do ambiente - Cenário C . . . . .	88
5.17	Tempos do Sistema de Visão. . . . .	89
5.18	Cálculo da homografia - Cenário C . . . . .	89
5.19	Imagen do piso - Cenário C . . . . .	90
5.20	Influência da iluminação - Cenário C . . . . .	90
5.21	Trajetória percorrida pelo robô - Cenário C . . . . .	91
5.22	Comportamento da coordenada y da pose do robô . . . . .	91
5.23	Comportamento da variância da linha de fechamento de laço . . . . .	92
5.24	Cenário D: Supermercado Nata-RN . . . . .	92
5.25	Trajetória percorrida pelo robô - Cenário D . . . . .	93
5.26	Imagen do piso - Cenário D . . . . .	94
5.27	Falha no processamento de imagem - Cenário D . . . . .	94
5.28	Falha no processamento de imagem - Cenário D . . . . .	94
5.29	Trajetória usando SLAM com linhas e SLAM com pontos. . . . .	96
5.30	Comparação entre as variâncias da coordenada $x_R$ da pose do robô. . . . .	97
5.31	Comparação entre as variâncias da coordenada $y_R$ da pose do robô. . . . .	97
5.32	Comparação entre as variâncias da coordenada $\theta_R$ da pose do robô. . . . .	97
6.1	Representação de um feixe de sonar em uma grade de ocupação. . . . .	100
6.2	Esquema de construção da grade de ocupação. . . . .	106
6.3	Área de atuação do algoritmo <i>Mean Shift</i> . . . . .	107
6.4	Funcionamento do algoritmo <i>Mean Shift</i> . . . . .	107
6.5	Gráfico do tempo de processamento do sistema de segmentação. . . . .	108
6.6	Classificação - formas simples . . . . .	108
6.7	Classificação - forma mais complexa . . . . .	109
6.8	Efeito da inclinação da câmera em um conjunto de pontos. . . . .	109
6.9	Efeito da inclinação da câmera no mapeamento imagem-mundo. . . . .	110
6.10	Região de mapeamento imagem-mundo. . . . .	110
6.11	Ilustração do funcionamento do algoritmo de classificação. . . . .	111
6.12	Comportamento da variância do sensor. . . . .	112
6.13	Construção de grade visual. . . . .	113
6.14	Descrição do ambiente do primeiro experimento. . . . .	116
6.15	Imagen do piso - Experiência 1 . . . . .	116
6.16	Grade de ocupação. . . . .	117
6.17	Mapa híbrido: Grade de ocupação e Mapa de Características. . . . .	117
6.18	Descrição do ambiente. . . . .	118
6.19	Imagen do piso - Experiência 2 . . . . .	118
6.20	Grade de ocupação. . . . .	119
6.21	Mapa híbrido: Grade de ocupação e Mapa de Características. . . . .	119

A.1	Representação normal da reta.	143
A.2	Sistemas de coordenadas Fixo ( $F$ ) e Móvel ( $M$ ).	146



---

# **Lista de Tabelas**

---

4.1	Simbolos das equações de Kalman . . . . .	67
5.1	Resultados experimentais: Cenário A. . . . .	84
5.2	Resultados experimentais: Cenário B. . . . .	87
5.3	Resultados comparativos: Linhas vs Pontos . . . . .	95
6.1	Símbolos do algoritmo <i>grade_ocupaçao</i> . . . . .	114



---

# **Lista de Algoritmos**

---

1	<i>Filtro de Kalman Discreto.</i>	37
2	<i>Filtro de Kalman Estendido.</i>	38
3	<i>Grade de Ocupação Visual</i>	115



---

# Capítulo 1

## Introdução

---

O desenvolvimento tecnológico vem proporcionando um significativo aumento de qualidade de vida para os humanos. Um exemplo clássico desses avanços são os robôs. De uma forma mais ampla, os robôs podem ser definidos como máquinas que, além de serem capazes de reproduzir tarefas associadas aos seres humanos e movimentos implícitos em sua construção, complementam a parte mecânica com dispositivos eletrônicos inteligentes de suporte, os quais lhes dão um certo grau de autonomia [Medeiros 1998].

Craig (2004) classifica os robôs em duas grandes classes: robôs manipuladores e robôs móveis. Os manipuladores robóticos têm forte atuação nas indústrias, sobretudo na automobilística. A classe de robôs móveis é constituída por robôs que tem a capacidade de locomoção no espaço tridimensional (robôs aeroespaciais e robôs subaquáticos) ou no espaço planar (robôs terrestres).

As aplicações envolvendo robôs terrestres vêm crescendo significativamente nos últimos anos devido a uma classe de tarefas que esses podem realizar, já que, ao contrário dos manipuladores, possuem capacidade de se locomover livremente pelo espaço de trabalho, limitados apenas por eventuais obstáculos. Além disso, as rodas permitem uma maior praticidade de locomoção em ambientes planos, dão um maior suporte estático ao robô e o simples acionamento dos motores faz com que ele se move.

A aplicação prática de robôs móveis junto a diferentes atividades em nossa sociedade vem demonstrando o quanto promissor é o futuro desta área. Por exemplo, seu uso em aplicações domésticas (e.g. aspiradores de pó e cortadores de grama robóticos), industriais (e.g. transporte automatizado e veículos de carga autônomos), urbanas (e.g. transporte público, cadeiras de rodas robotizadas), militares (e.g. sistemas de monitoramento aéreo remoto) e de segurança e defesa civil e militar (e.g. controle e patrulhamento de ambientes, resgate e exploração em ambientes hostis), demonstra a grande gama de aplicações atuais dos robôs móveis e os interesses econômicos envolvidos em relação ao seu desenvolvimento e aplicação [Wolf et al. 2009].

Um fator essencial para um sistema robótico é torná-lo autônomo. Robôs autônomos devem apresentar reações às mudanças do ambiente, comportamento inteligente, integração entre vários sensores, resolução de múltiplas tarefas, robustez, capacidade de operar sem falha, programabilidade, modularidade, flexibilidade, expansibilidade, adaptabilidade e raciocínio global.

Ainda no contexto de autonomia, destaca-se o problema da navegação que pode ser definido como os procedimentos necessários para permitir que um sistema robótico saia de uma configuração inicial e alcance uma configuração final percorrendo um determinado caminho planejado, desviando de obstáculos quando existirem, obedecendo a restrição temporal imposta para o cumprimento dessa tarefa. Esse processo é comumente dividido em sub-problemas que podem ser abstraídos em cinco níveis hierarquizados de autonomia: *Mapeamento do Ambiente, Localização, Planejamento de Caminho, Geração de Trajetória e Execução de Trajetória* [Alsina et al. 2002].



Figura 1.1: Etapas de um processo de navegação robótica.

Quando as duas primeiras etapas do processo de navegação devem ser realizadas de forma simultânea geram o problema de SLAM (*Simultaneous Localization and Mapping*). Para facilitar o entendimento, antes de introduzir o SLAM serão comentadas as etapas de Localização e Mapeamento de forma separada.

## 1.1 Localização

Uma importante etapa no processo de navegação de robôs móveis autônomos é a tarefa de localização. A partir de um conhecimento acumulado sobre o ambiente e utilizando as leituras atuais dos sensores, o robô deve ser capaz de determinar e manter atualizada a sua posição e orientação em relação a este ambiente, mesmo que os sensores apresentem erro e/ou ruído. Em outras palavras, localizar um robô consiste em determinar sua pose ou postura (posição e orientação) no espaço de trabalho a cada instante de tempo.

Borenstein & Feng (1996) classificam os métodos de localização em duas grandes categorias: métodos de localização relativa, que dão a pose do robô em relação a uma pose inicial e métodos de localização absoluta, que dão a pose global do robô e não necessitam das posições calculadas anteriormente.

O modelo mais simples de localização envolve apenas parâmetros internos do robô, e por isso não depende da obtenção de informações do ambiente ao seu redor. Normalmente, as informações odométricas são obtidas através de sensores acoplados às rodas dos robôs, de forma a calcular o deslocamento dos mesmos a partir do movimento destas. Na prática, existem diversos fatores que causam erros nesse tipo de dispositivo como a imprecisão mecânica do sistema e irregularidades do terreno em que o robô atua. Outro fator que dificulta e pode até inviabilizar a localização do robô a partir da odometria é o fato de que os pequenos erros causados durante o cálculo da posição do robô vão se acumulando ao longo do tempo.

Em se tratando de localização absoluta, Borenstein & Feng (1996) definem marcos como elementos distintos que o robô é capaz de reconhecer através de seus sensores. Geralmente os marcos estão em posições fixas e conhecidas em relação ao referencial do ambiente. As informações dos sensores são utilizadas para reconhecer os marcos e determinar as suas posições com relação ao referencial do robô. Conhecendo a posição dos marcos com relação ao referencial do ambiente e com relação ao referencial do robô, é possível determinar a pose do robô com relação ao referencial do ambiente.

A figura 1.2 ilustra uma etapa do processo de localização. Nessa ilustração, o sistema de localização do robô móvel detecta o ponto de interseção entre as retas existentes no piso de um ambiente a partir de imagens providas por uma câmera. Sabendo a posição deste ponto no sistema de coordenadas global, o sistema calcula a pose do robô neste sistema de coordenadas.

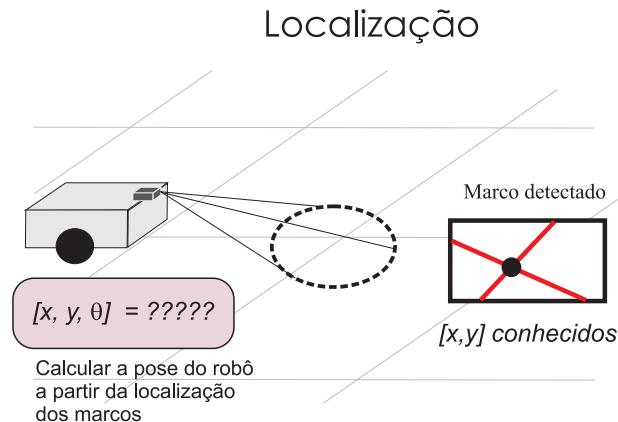


Figura 1.2: Etapa de um processo de localização baseado em marcos.

Vale ressaltar que as imprecisões dos sensores do robô podem comprometer a etapa de localização. Assim, o grande desafio na solução do problema de localização está no fato de que tanto as informações sobre o ambiente como os dados fornecidos pelos sensores são normalmente limitados e imprecisos. Nesse contexto, técnicas probabilísticas têm sido amplamente utilizadas na solução desse problema [Thrun et al. 2005].

## 1.2 Mapeamento

No nível de Mapeamento, o robô deve coletar dados do seu entorno, utilizando sensores, visando gerar modelos computacionais com as principais características estruturais do ambiente. Em outras palavras, é necessário que os robôs sejam equipados com dispositivos de percepção capazes de fornecer algum tipo de informação de seu entorno, de modo que, a partir de algum processamento, essas informações sejam usadas para construir um mapa do ambiente. Além disso, para que um sistema robótico possa construir um mapa consistente, é necessário determinar a posição e orientação do robô com relação a algum referencial fixo no mundo. Esse processo de obtenção de dados sensoriais, seu subsequente processamento e mais a inferência da posição e orientação exata do robô, tendo como objetivo a construção de um mapa é chamado de Mapeamento Robótico.

Para melhor formalizar o problema de mapeamento robótico, devemos estabelecer algumas hipóteses (ou restrições) básicas. A primeira hipótese é que o robô possui sensores proprioceptivos e exteroceptivos que o permitem coletar dados sobre si mesmo (posição e orientação) e sobre o ambiente, respectivamente. A segunda é que este sistema de percepção deve permitir, a todo momento, o conhecimento exato da posição e orientação do robô em relação a algum sistema de coordenadas ou referencial fixo global adotado.

A figura 1.3 ilustra uma etapa do processo de construção de um mapa.

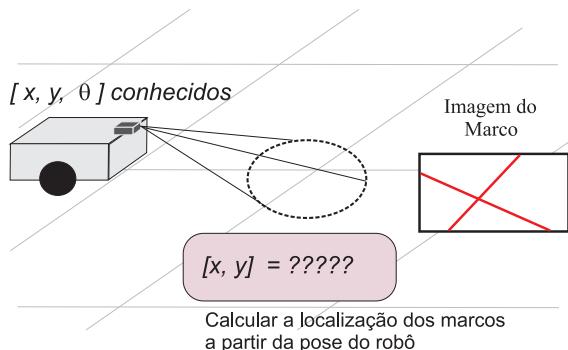


Figura 1.3: Processo de construção de um mapa.

Nessa ilustração, o sistema de mapeamento do robô móvel detecta o ponto de interseção entre as retas existentes no piso de um ambiente, a partir de imagens providas por uma câmera. Sabendo a posição e orientação do robô, o sistema calcula a localização dos pontos detectados e armazena esta informação de forma sistemática em uma estrutura computacional que irá representar o mapa do ambiente. Nesse caso, o modelo espacial será constituído simplesmente das coordenadas dos pontos detectados.

A hipótese de que o sistema de percepção tem o conhecimento exato da posição e orientação do robô em relação a algum referencial global adotado nem sempre é verdadeira, visto que, na maioria das vezes, para se obter a posição e orientação do robô dentro do seu ambiente é necessário um mapa, a priori. Aqui surge um conflito: para se obter um mapa consistente é necessário o conhecimento exato da posição e orientação do robô e para ter condições de inferir a posição e orientação exatas do robô é necessário um mapa do ambiente. Note que há uma interdependência entre localização (inferência da posição e orientação do robô no ambiente) e mapeamento e para tratar este conflito de forma simultânea surgiram as técnicas de SLAM.

### 1.3 SLAM

O termo SLAM é indicado como uma abreviatura para Localização e Mapeamento Simultâneos e foi originalmente desenvolvida por Leonard & Durrant-Whyte (1991) com base em trabalhos anteriores de Smith et al. (1987). A técnica de SLAM está preocupada com o problema da construção de um mapa de um ambiente desconhecido por um robô móvel que ao mesmo tempo utiliza as informações do mapa para calcular sua pose.

Uma compreensão intuitiva do processo de SLAM pode ser conduzida por um exemplo hipotético. Considere um robô móvel simples: um conjunto de rodas ligado a um motor, uma câmera e atuadores físicos para controlar a velocidade e a direção da unidade; agora imagine o robô sendo utilizado remotamente por um operador para mapear locais inacessíveis; os acionadores permitem que o robô se mova, e a câmera fornece informação visual suficiente para o operador entender onde e como os objetos estão em torno do robô. O que o homem está fazendo é um exemplo de SLAM: determinar a localização de objetos no ambiente é um exemplo de mapeamento, e inferir a pose do robô com relação a esses objetos é um exemplo de localização.

O problema de SLAM pode ser pensado como um “problema do ovo e da galinha” (*chicken or egg problem*) pois: para a localização é necessário o conhecimento do mapa enquanto que uma estimativa exata da pose do robô é necessária para construir esse mapa. A figura 1.4 ilustra o problema de SLAM.

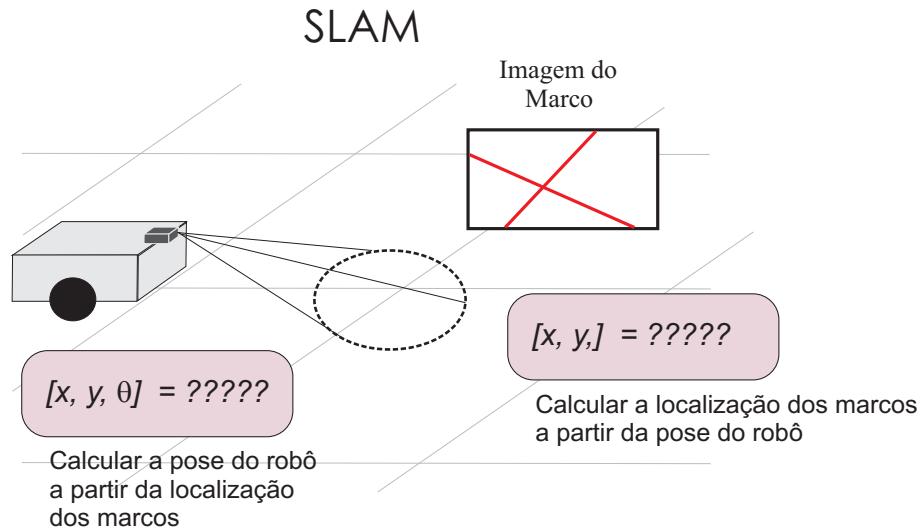


Figura 1.4: Problema de SLAM.

Dois dos principais trabalhos desenvolvidos sobre este tema provaram que erros na localização do robô estão correlacionados com os erros nas observações dos marcos [Smith & Cheesman 1987, Durrant-Whyte 1988]. A correlação existe porque um erro na localização terá um efeito sobre a percepção universal da localização de todos os outros marcos. Compreender e utilizar a relação entre os erros de observação e de localização do robô é a principal motivação por trás do problema de localização e mapeamento simultâneos.

Sabe-se que as informações obtidas a partir de sensores são inevitavelmente imprecisas, devido a limitações físicas e a fenômenos que não podem ser previstos e, portanto, modelados computacionalmente. Dessa forma, as estimativas de localização e mapeamento obtidas pelo robô a partir dessas informações possuirão erros que estabelecem uma diferença entre a realidade e como ela é percebida pelo robô. Se esses erros não forem tratados chegará o momento em que a imprecisão será tão grande que já não faz sentido utilizar essas estimativas, invalidando todo o processo.

Neste sentido, soluções probabilísticas lidam com o problema de SLAM através da minimização simultânea das incertezas envolvidas em cada uma das estimativas, reconhecendo que ambas são dependentes entre si. Resultados precisos da localização são necessários para o estabelecimento de um mapa preciso e vice-versa, e dessa forma não faz sentido abordar apenas um dos problemas, é necessário resolvê-los simultaneamente, eliminando sistematicamente os erros antes que eles se acumulem.

Um processo de SLAM, onde o mapa é formado por um conjunto de marcos, é composto de várias partes: *Extração de Marcos, Associação de Dados, Estimação de Estados, Atualização dos Estados e Atualização dos Marcos*.

Marcos Naturais ou *Landmarks* são características que podem ser facilmente observadas e distinguidas no ambiente. Eles são utilizados pelo robô para calcular sua pose absoluta. Uma forma de entender como os marcos são utilizados é imaginar seus olhos vendados. Se você se move, de olhos vendados em uma casa, você pode chegar e tocar os objetos ou paredes, para que você não se perca. Essas características podem ajudá-lo a estabelecer uma estimativa de onde você está. De forma semelhante, os robôs procuram por características no ambiente onde estão navegando para ajudá-lo a se localizar.

Consideram-se bons marcos naturais os que podem ser reobservados (permitindo, por exemplo, ser visualizados ou detectados de posições diferentes) e que possam ser facilmente identificados sem misturá-los. Em outras palavras, se um marco for observado em um momento posterior, ele deve ser facilmente distinto dos demais marcos já detectados.

De forma resumida, os pontos-chave para a escolha adequada dos marcos são os seguintes: os marcos deverão ser facilmente reobserváveis, devem ser distinguíveis entre si, devem ser abundantes no ambiente e devem ser estáticos. É importante comentar que depois de decidir qual marco utilizar o robô deve ser capaz de extraí-lo de forma confiável e que essa extração depende fortemente de quais são os sensores utilizados pelo robô.

No SLAM, quando um marco é reconhecido em um instante posterior, o robô adquire novas informações não apenas sobre a posição do marco no ambiente, mas também informações relativas à sua própria localização. Com isso, ele consegue refinar simultaneamente ambas, tornando a localização e o mapeamento globalmente mais precisos.

O problema da correspondência ou *Matching* é verificado quando um marco é reobservado. Para ilustrar tal problema considere o exemplo: para nós humanos, uma cadeira pode ser considerada um marco; digamos que entramos em uma sala e vemos uma cadeira específica; saímos da sala e em seguida (algum tempo mais tarde), voltamos à sala; se olharmos e verificarmos a existência de uma cadeira então estamos associando-a cadeira vista anteriormente - definindo uma correspondência.

Isso parece simples, mas a correspondência pode ser bastante complicada. Digamos que a sala tinha duas cadeiras praticamente idênticas. Quando retornarmos à sala podemos não distinguir com precisão quais são as cadeiras vistas anteriormente pois ambas têm a mesma aparência. Uma solução seria assumir que a cadeira mais à direita é a cadeira mais à direita vista anteriormente e que à cadeira mais à esquerda é a cadeira mais à esquerda vista anteriormente. Agora imagine que você entra na sala e visualiza as cadeiras de um outro ângulo. A correspondência torna-se mais difícil ainda.

Na prática, os problemas da associação de dados são: o marco pode não ser observado a cada passo, pode-se identificar um marco uma vez e ele não ser mais observado, um marco pode ser associado indevidamente a outro marco visto anteriormente e, um marco pode ser reobservado e não reconhecido como um mesmo já visto anteriormente. Ressalta-se que mesmo considerando que os marcos sejam estáticos e abundantes no ambiente é importante definir uma política adequada de correspondência.

No SLAM baseado em marcos, as características dos marcos serão inseridas no vetor de estados à medida que eles são identificados, caso não haja correspondência deles com os já detectados. Se houver uma reobservação de um marco detectado anteriormente apenas uma atualização das medidas é feita.

Segundo Thrun et al. (2005) duas abordagens podem ser usadas para calcular a correspondência entre os marcos. A primeira, determinística, consiste em utilizar a Distância Euclidiana dado que os marcos encontram-se espaçados no ambiente. A segunda, probabilística, consiste em utilizar como fator de correspondência a Distância de Mahalanobis. Outra métrica também utilizada é a distância de *Hamming* [Hamming 1950].

Apresentados alguns conceitos relacionados ao problema de SLAM utilizaremos a figura 1.5 para mostrar a idéia básica do seu funcionamento.

A figura 1.5 (a) mostra o robô em uma configuração inicial que possibilita a observação de um marco (ponto de interseção entre as linhas). No início da navegação o robô assume sua pose conhecida ( $[x, y, \theta] = [0, 0, 0^\circ]$ ) e como ele identifica pela primeira vez um marco, este é inserido diretamente no vetor de estados. Em seguida, o robô se desloca e um marco é observado (figura 1.5 (b)). É feita uma etapa de correspondência para verificar se o marco detectado é antigo ou novo. Caso seja detectado um novo marco ele é inserido no “mapa de representação” do ambiente e então o robô calcula sua nova pose com base na odometria e na localização deste marco. Caso o marco detectado já exista no mapa, é realizada uma atualização da pose do robô e da localização dos marcos.

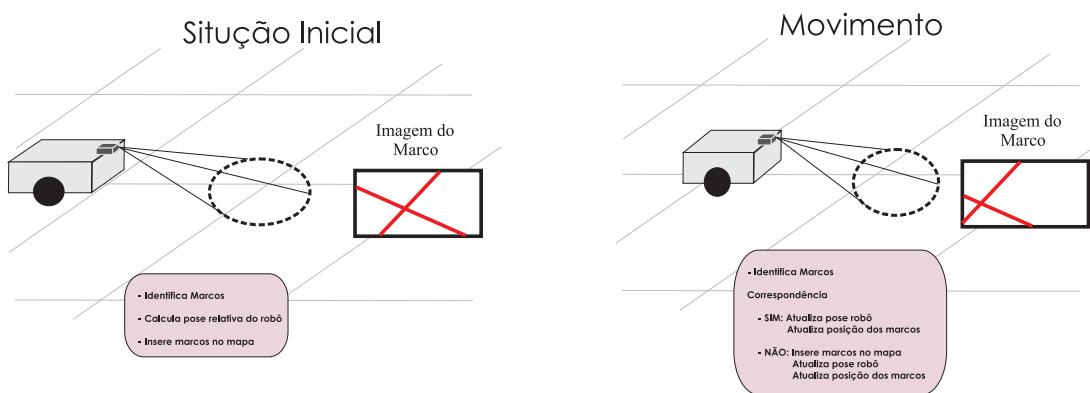


Figura 1.5: Esquema simplificado de SLAM.

As etapas de *Estimação de Estados*, *Atualização dos Estados* e *Atualização dos Marcos* normalmente são feitas utilizando estimadores estatísticos como o Filtro de Kalman e o Filtro de Partícula. Detalharemos estas fases nos Capítulos seguintes.

## 1.4 Motivação e Justificativa

Um atributo essencial para autonomia de robôs inteligentes é uma percepção confiável do mundo. Um robô capaz de perceber o ambiente usando seus sensores será capaz de realizar ações inteligentes para executar tarefas úteis e importantes, como a manipulação de objetos, limpeza hospitalar, entrega de mercadorias, iteração homem-robô, etc..

No tocante ao SLAM, acredita-se que uma solução para este problema permite o surgimento de robôs verdadeiramente autônomos, capazes de navegar de maneira segura por ambientes desconhecidos e cumprir objetivos sem a necessidade de auxílio externo.

Muitos são os sensores utilizados nos sistemas de percepção dos robôs para resolver o problema de SLAM. Como exemplo têm-se os sonares e os scanner laser. Porém, além da confiabilidade da percepção, para a aceitação generalizada de aplicações, é necessário que as tecnologias utilizadas proporcionem uma solução com preço acessível, ou seja, que os componentes sejam de baixo custo.

Sensores de distância conseguem fornecer apenas informações relativas à posição dos objetos em relação ao robô, geralmente em um único plano, o que dificulta a caracterização única dos objetos que o robô pode utilizar como marcos.

Para contornar estes problemas, uma solução é utilizar câmeras nos robôs para resolver os problemas de percepção do ambiente. Devido à utilização maciça de câmeras digitais pessoais, câmeras em computadores e celulares, o preço destes dispositivos diminuiu significativamente e os fizeram muito atraentes. Além disso, as câmeras podem ser usadas para resolver uma série de problemas-chave na robótica e em outras operações automatizadas pois fornecem uma variedade muito grande de informação do ambiente, consomem pouca energia e são facilmente integradas ao hardware do robô. Os principais desafios são tirar proveito deste poderoso e barato sensor e criar algoritmos confiáveis e eficazes que possam extrair as informações necessárias para a resolução de problemas.

No tocante à autonomia, sabe-se que a execução eficiente de tarefas por robôs é estreitamente relacionada ao conhecimento do seu espaço de trabalho. Existem basicamente três formas de representar os ambientes: representação métrica, representação topológica e representação de características. Cada uma dela tem suas vantagens.

Os mapas de características armazenam informações geométricas de formas encontradas nos ambientes, como círculos, retângulos, retas, entre outras. Tais características são encontradas principalmente em ambientes internos e estruturados, por exemplo, casas, corredores, escritórios e laboratórios. Esta representação é vantajosa por fornecer um melhor detalhamento do ambiente e facilitar a iteração entre robôs e humanos.

O mapeamento métrico produz uma definição geométrica do ambiente no qual o robô está inserido. Nesta abordagem o ambiente é apresentado de maneira detalhada incluindo os objetos que o compõe como paredes, obstáculos, passagens, entre outros, mantendo uma boa relação com o mundo real. Além disso, os mapas métricos são fáceis de serem construídos e mantidos, e são mais adequados para tratar o problema da correspondência.

Em face ao comentado até então, a motivação para este trabalho foi a de fornecer uma solução de SLAM com dupla representação do ambiente (métrica e de características) com o intuito de proporcionar ao robô um grau maior de autonomia. Além disso, o desafio foi o de utilizar apenas uma câmera como sensor principal do robô.

A justificativa está no fato da utilização de câmera como principal sensor do robô ser bastante difundida na comunidade nacional e internacional e, do ponto de vista prático, produzir soluções com representações mais ricas do ambiente com um preço mais acessível.

## 1.5 Objetivo

O objetivo desta tese é apresentar uma técnica de SLAM (Localização e Mapeamento Simultâneos) adequada para ambientes planos com linhas presentes no chão, de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de visão monocular.

Inicialmente, é feita uma etapa de segmentação para classificar as partes da imagem em “chão” e “não-chão”. Em seguida, o processamento de imagem identifica linhas na parte “chão” e os parâmetros dessas linhas são mapeados para o mundo, usando uma matriz de homografia. Finalmente, as linhas identificadas são usadas como marcos no SLAM, para construir um mapa de características.

Em paralelo, a pose corrigida do robô, a incerteza em relação à pose e a parte “não-chão” da imagem são usadas para construir uma grade de ocupação, gerando um mapa métrico com descrição dos obstáculos.

A utilização simultânea dos dois tipos de mapa obtidos (métrico em grade e de características) dá maior autonomia ao robô, permitindo acrescentar tarefas de planejamento em simultâneo com a localização e mapeamento. O esquema resumido do sistema proposto nesta tese é apresentado na figura 1.6.

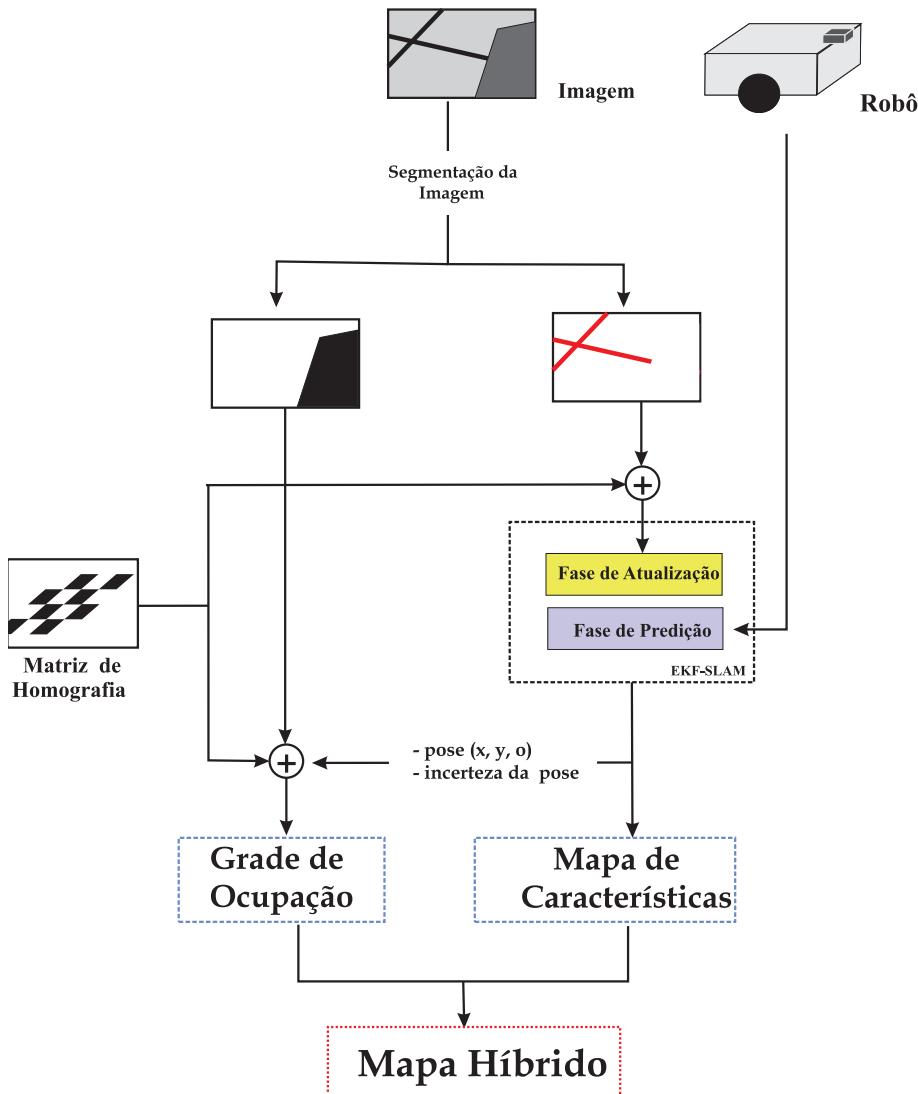


Figura 1.6: Esquema simplificado do sistema proposto nesta Tese.

## Contribuições

- utilização de técnicas consolidadas para a construção de um sistema de SLAM visual monocular em tempo real sem inicialização atrasada;
- modelagem do sensor ótico de modo a permitir a utilização de parâmetros de retas extraído de imagens diretamente nas equações de Kalman sem etapas intermediárias de cálculo de pose e distância;
- construção de grade de ocupação a partir de visão monocular;
- SLAM monocular com representação dual do ambiente em grade de ocupação e mapa de características;

## 1.6 Organização do trabalho

O restante desse documento encontra-se organizada como segue:

- O Capítulo 2 apresenta os fundamentos teóricos do problema de SLAM, a abordagem probabilística desse problema, os conceitos matemáticos sobre Filtro de Kalman e um conjunto de trabalhos que mostram a evolução da área;
- O Capítulo 3 descreve o contexto de soluções para SLAM usando câmera como sensor principal do robô, apresenta as três principais formas de representação dos ambientes comentando vantagens e desvantagens de cada uma e finda com um relato dos trabalhos em SLAM visual;
- O Capítulo 4 mostra o sistema proposto nesta tese, o ferramental teórico necessário para o seu entendimento, a modelagem matemática do sistema e um conjunto de trabalhos relacionados;
- O Capítulo 5 tráz os resultados do SLAM visual proposto que foi testado em quatro ambientes diferentes com configurações distintas. Nesse capítulo também apresentou-se um comparativo do sistema proposto contra uma abordagem clássica usada em SLAM visual;
- O Capítulo 6 descreve a construção de uma grade de ocupação probabilística 2D usando visão monocular. A modelagem pressupõe o conhecimento da pose do robô, das incertezas da pose e de uma matriz de homografia. A obtenção destes três requisitos é possível a partir do SLAM visual apresentado no Capítulo 4. Resultados são apresentados para validar a proposta;
- Por fim, no Capítulo 7 são apresentadas as principais conclusões sobre o sistema proposto e faz-se uma análise das potenciais perspectivas que podem ser adotadas para uma utilização deste sistema bem como possíveis melhorias.



---

# Capítulo 2

## Localização e Mapeamento Simultâneos

---

*O objetivo deste Capítulo é apresentar os fundamentos teóricos do problema de SLAM. O texto discorrerá sobre o assunto de forma clara e concisa com o intuito de prover condições mínimas necessárias para o entendimento do tema. O SLAM deve ser encarado como uma idéia e não como um simples algoritmo. Existem vários passos envolvidos no SLAM e estas diferentes etapas podem ser resolvidas usando um conjunto de diferentes técnicas. Será apresentado a visão probabilística do problema, os conceitos matemáticos sobre Filtro de Kalman e um conjunto de trabalhos que mostram a evolução da área.*

### 2.1 Correlações e incertezas

Como visto no Capítulo anterior, a localização e o mapeamento são fundamentais para que um robô possa navegar de forma autônoma. A maioria dos algoritmos de localização utiliza um mapa ou alguma informação do ambiente para determinar a pose (posição e orientação) do robô. Se o mapa é dado *a priori*, o robô pode se auto-localizar pela correspondência de algumas características observadas no ambiente.

A aquisição de um modelo preciso e completo do ambiente para a localização é uma tarefa tediosa, especialmente em grandes ambientes. Além disso, a maior parte dos algoritmos de mapeamento necessitam de uma boa localização para construir mapas que representam fielmente o ambiente. Um mapa produzido manualmente pode ser incompatível com a realidade, isto é, pode não refletir adequadamente todas as características que podem ser detectadas com o sistema de percepção do robô. Por isso, o robô deverá ser capaz de construir seu próprio modelo do ambiente.

Ocorre que existem situações nas quais não se tem nem informações prévias sobre o mapa e nem uma boa localização do robô. Uma solução para este problema consiste em o robô construir o mapa enquanto computa uma estimativa de sua pose. Isto é conhecido na literatura como uma solução para o problema de SLAM.

O termo SLAM é indicado como uma abreviatura para Localização e Mapeamento Simultâneo e foi originalmente desenvolvida por Leonard & Durrant-Whyte (1991) com base em trabalhos anteriores de Smith et al. (1987). Esta técnica está preocupada com o problema da construção de um mapa de um ambiente desconhecido por um robô móvel que ao mesmo tempo utiliza as informações do mapa para calcular sua pose.

No problema de SLAM, um robô móvel usa seus sensores para explorar o ambiente, ganha conhecimento sobre ele, interpreta o cenário, constrói um mapa adequado e, em seguida, calcula a sua posição relativa usando o mapa que está sendo criado. Uma vez que os mesmos dados sensoriais são usados para computar a pose do robô e para a construção do mapa é fácil perceber que a estimativa da pose do robô e as estimativas das características do ambiente são correlacionadas. A correlação existe porque um erro na localização terá efeito na percepção universal da localização de todas as características mapeadas [Smith & Cheesman 1987, Durrant-Whyte 1988].

Uma vez que o robô necessita de sensores para obter informações ao seu redor não se pode omitir que estas informações são imprecisas devido às restrições/imperfeições físicas dos sensores e, portanto, as estimativas de localização e mapeamento estabelecem diferenças com a realidade. As primeiras tentativas de se lidar com esses erros procuraram fazê-lo através da construção de sensores cada vez mais precisos, ou seja, que proporcionassem um resultado final mais confiável. Entretanto, o aumento na precisão fez com que os sensores utilizados ficassem cada vez mais caros. Vale ressaltar que SLAM é um problema bastante complexo devido não só às imperfeições das medidas sensoriais mas também devido às incertezas inerentes ao movimento do robô.

## Sensores em SLAM

Um dos sensores mais utilizados em SLAM são os scanners laser. Estes sensores são precisos, eficientes e não exigem muita computação para processar a informação por eles fornecida. As principais desvantagens na utilização destes sensores são que não funcionam bem em superfícies formadas por vidro e são muito caros. Soluções para SLAM usando laser podem ser vistas nos trabalhos de Welle et al. (2010), Burgard et al. (2009), Blanco et al. (2008) e Anousaki & kopoulous (2007).

Uma segunda opção de sensores também bastante utilizados em SLAM são os sonares. Eles são sensores baratos, quando comparados aos scanners laser, porém, não apresentam leituras tão precisas e enfrentam grandes problemas de reflexão das ondas sonoras. Exemplos da utilização de sonares em SLAM são apresentados por Huang et al. (2010), Yap & Shelton (2009) e Feng et al. (2008).

Há ainda um outra opção de sensores usados em SLAM, as câmeras. A utilização de sensores visuais como uma forma de obtenção de informações do ambiente parece ser mais intuitivo e pode ser mais atraente que laser ou sonar. O maior atrativo da utilização de imagens está na riqueza de informações que elas são capazes de fornecer, qualitativa e quantitativamente [Williams & Reid 2010, Kootstra & Schomaker 2009, Milford & Wyeth 2008]. A figura 2.1 resume o comentado até este instante.

No tocante ao uso de câmeras, o que costumava ser o grande gargalo da visão computacional (memória e capacidade de processamento) está sendo sensivelmente resolvido com a eficiência e o custo dos hardwares existentes no mercado. O rápido aumento no poder de processamento faz com que seja possível lidar com uma maior quantidade de informações, permitindo uma melhor compreensão do ambiente e aumentando a quantidade de dados que o robô dispõe para tomar decisões [Se et al. 2001].

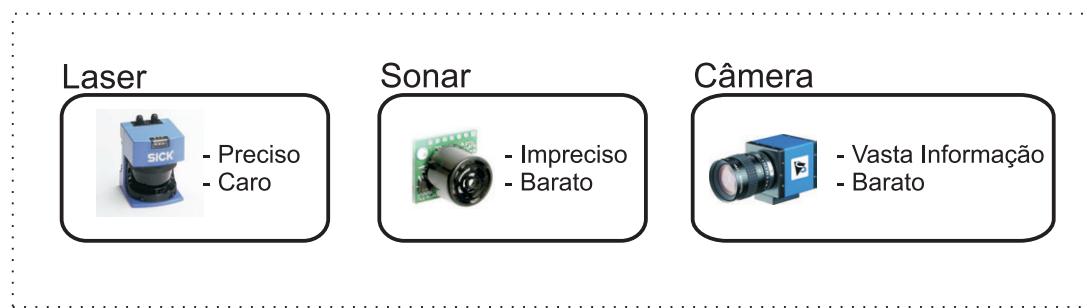


Figura 2.1: Sensores normalmente utilizados em SLAM.

É comum ser encontrado na literatura soluções para SLAM combinando ou fundindo informações de diferentes tipos de sensores. Um exemplo da utilização de laser e sonar é o trabalho de Diosi et al. (2005). Já usando sonar e câmera, Ahn & Lee (2007) apresentam uma possível solução enquanto que Fu et al. (2007b) usam laser e câmera para o resolver o problema de SLAM.

As características descobertas pelo sistema de percepção do robô seja usando laser, sonar ou câmera, são incertas devido ao ruído dos sensores. Além disso, a relação espacial entre as características detectadas e o robô também é incerta porque a base de dados sensorial é imprecisa. Por estas razões, qualquer solução matematicamente rigorosa para o problema de SLAM deve dispor de uma estrutura que permita a representação, manipulação e propagação da incerteza espacial [Skrzypczynski 2009].

Entre as várias abordagens matemáticas que podem suprir estes requisitos, a probabilística é adequada, porque tem forte fundamentação teórica que fornece uma maneira de provar propriedades de convergência e de otimização das soluções desenvolvidas. Individualmente os erros não podem ser previstos, mas quando considerados como um conjunto de fenômenos isolados percebeu-se que eles poderiam ser tratados estatisticamente, apresentando padrões que podem ser modelados. A idéia consiste em os estados de localização e mapeamento do robô deixarem de apresentar valores únicos e se tornam distribuições de probabilidade que indicam todos os estados possíveis dada a precisão das observações realizadas até aquele momento [Thrun et al. 2005].

A obtenção de estimativas precisas se torna, então, uma questão de minimizar as incertezas contidas em cada uma das variáveis de interesse visando eliminar os erros aleatórios gerados pelos ruídos oriundos da fonte sensorial [Elfes 1987].

## 2.2 Soluções para SLAM

Na década de 90, abordagens probabilísticas tornaram-se dominantes em soluções de SLAM, merecendo destaque o Filtro de Kalman (e suas variações) e o Filtro de Partículas. A principal razão para a popularidade destas técnicas reside no fato de que o mapeamento robótico é caracterizado por incertezas e ruídos dos sensores. Logo, algoritmos com base em probabilidade podem ser usados para modelar explicitamente as diferentes fontes de ruído e seus efeitos sobre as medidas [Thrun 2002].

O Filtro de Kalman (FK) é um conjunto de equações matemáticas que constitui um processo eficiente de estimação uma vez que o erro quadrático é minimizado. Através da observação da variável denominada variável de observação, outra variável não observável, denominada variável de estado pode ser estimada eficientemente [Kalman 1960].

Uma vez que o sistema está escrito na forma espaço de estados, o filtro de Kalman pode ser aplicado quando as equações de modelo são lineares e os ruídos sejam gaussianos brancos. A derivação do filtro apóia-se no fato de que tanto os ruídos das equações de medição e transição, como o vetor inicial de estado, são normalmente distribuídos. Isto significa que apenas os dois primeiros momentos são suficientes para descrever todos os estados em qualquer instante [Kalman 1960].

A perda da linearidade da equação de observação pode ser contornada através da utilização de uma variação do filtro de Kalman, o Filtro de Kalman Estendido (FKE). A extensão do filtro trata a não linearidade da equação de observação linearizando-a através da expansão em série de Taylor.

Uma variação estável, precisa e simples do filtro de Kalman que é implementada propagando a inversa da matriz de covariância (matriz de informação) é o Filtro de Informação (FI) que, assim como o filtro de Kalman, também possui sua versão estendida, denominada Filtro de Informação Estendido (FIE) [Thrun & Liu 2003].

A principal desvantagem desse filtro, quando aplicado a sistemas não-lineares, é que existe a necessidade de recuperar, através da inversão da matriz de informação, uma estimativa do estado na etapa de atualização [Walter et al. 2005]. Além disso, Walter et al. (2007) comentam que, para ambientes de grande dimensionalidade, há a necessidade de computar todas essas inversões da matriz de informação, o que faz o FI computacionalmente inferior ao FK e, na verdade, essa é uma das razões pelas quais o FKE é muito mais popular que o FIE.

O Filtro de Kalman Unscented (FKU) ataca as questões de linearização do FKE que podem introduzir grandes erros na matriz de covariância, e levá-lo, por vezes, a soluções divergentes [Wan & Merwe 2001]. O FKU utiliza uma técnica de amostragem determinística conhecida como Transformação Unscented para selecionar um conjunto mínimo de pontos de amostragem (pontos sigma) distribuídos ao redor da média [Julier et al. 1995].

No FKU a distribuição do estado também é representada por uma gaussiana, mas é especificada usando os pontos sigma que são propagados através dos modelos não-lineares. Apesar da propagação das amostras através dos modelos ser computacionalmente cara, esta abordagem elimina a necessidade de calcular os jacobianos do Filtro de Kalman Estendido. Thrun et al. (2005) comentam que tanto o FKU quanto o FKE possuem a mesma complexidade assintótica, e para sistemas não-lineares o FKU produz resultados tão bons quanto o FKE. Estes autores também afirmam que a principal vantagem do FKU é que esse não necessita da computação dos jacobianos, aumentando assim a facilidade de implementação; a desvantagem consiste em uma solução menos analítica.

Até agora comentamos o filtro de Kalman como solução para modelos lineares gaussianos e apresentamos as idéias do FKE e do FKU para modelos não-lineares gaussianos. Para solucionar o caso de sistemas não-lineares com modelos não-gaussianos utiliza-se uma técnica para implementação de um filtro Bayesiano por meio do método de Simulação Monte Carlo denominada Filtro de Partículas (FP).

Essa técnica tem como objetivo a representação da função de densidade de probabilidade por um conjunto aleatório de amostras (partículas) com pesos associados [Thrun et al. 2005]. Por período de amostragem, cada partícula é modificada de acordo com o modelo do processo, incluindo a adição de ruído aleatório para simular o efeito do ruído nas variáveis de estado e, então, o peso de cada partícula é reavaliado com base na última informação sensorial. Define-se um algoritmo para descartar aquelas partículas com pesos próximos de zero e recriam-se novas partículas com base naquelas que sobram. Quando o número efetivo de amostras está abaixo de um determinado limiar, geralmente calculado com base em uma percentagem das M partículas, então a população das M partículas é reamostrada (*resampling*), eliminando-se probabilisticamente aquelas cujos pesos são pequenos e duplicando aquelas com pesos elevados.

A principal vantagem no uso do Filtro de Partículas, está, sem dúvida, na possibilidade de trabalhar com distribuições multi-modais; por outro lado, a desvantagem está ligada à sua elevada complexidade computacional. A fim de minimizar a complexidade, este tipo de filtro tem sido aplicado de forma combinada com o filtro de Kalman na técnica chamada FastSLAM [Montemerlo et al. 2002, Montemerlo et al. 2003].

A figura 2.2 apresenta um resumo das soluções matemáticas normalmente utilizadas para resolver o problema da localização e mapeamento simultâneos.

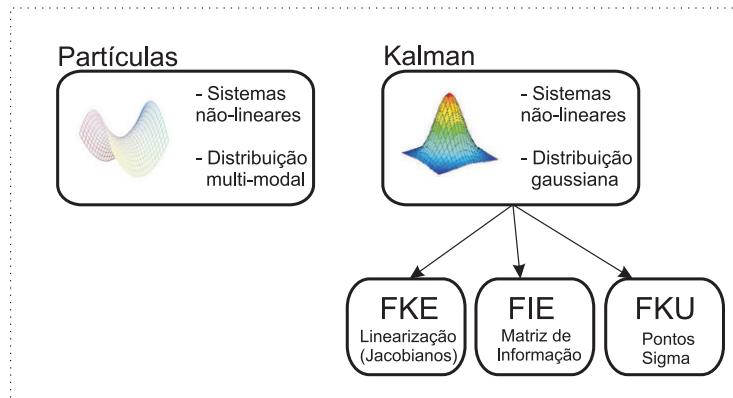


Figura 2.2: Soluções para SLAM.

A seguir apresentaremos, de forma resumida, os conceitos matemáticos da formulação de Kalman necessários ao entendimento do restante deste trabalho.

## 2.3 Filtro de Kalman

Em 1960, Rudolph Emil Kalman publicou um famoso artigo descrevendo um processo recursivo para solucionar problemas lineares relacionados à filtragem de dados discretos [Kalman 1960]. O filtro de Kalman é um procedimento aplicável quando os modelos estão escritos sob a forma espaço-estado e permite a estimativa dos parâmetros desconhecidos do modelo através da maximização da verossimilhança via decomposição do erro de previsão. Com o avanço computacional, o filtro de Kalman e suas extensões a problemas não lineares representam um produto largamente utilizado na engenharia.

### Filtro de Kalman Discreto - FKD

A modelagem do Filtro de Kalman Discreto pressupõe que o sistema seja linear e descrito pelo modelo de equações do sistema (2.1):

$$\begin{cases} \mathbf{s}_t &= \mathbf{A}_t \mathbf{s}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \gamma_t \\ \mathbf{z}_t &= \mathbf{C}_t \mathbf{s}_t + \delta_t \end{cases} \quad (2.1)$$

onde  $\mathbf{s} \in R^n$  é o vetor de estados;  $\mathbf{u} \in R^l$  é o vetor das entradas de controle;  $\mathbf{z} \in R^m$  é o vetor de medições; a matriz  $n \times n$ ,  $\mathbf{A}$ , é a matriz de transição de estados;  $\mathbf{B}$ ,  $n \times l$ , é a matriz de coeficientes de entrada; a matriz  $\mathbf{C}$ ,  $m \times n$ , é a matriz de observação;  $\gamma \in R^n$  representa o vetor de ruídos do processo e  $\delta \in R^m$  o vetor de erros de medição. Os índices  $t$  e  $t - 1$  representam os instantes de tempo atual e anterior respectivamente.

O filtro opera em modo de predição-atualização levando em consideração as propriedades estatísticas do ruído. Um modelo interno do sistema é usado para atualização e um esquema de realimentação realiza as medições. As etapas de predição e atualização para o FKD podem ser descritas pelos sistemas (2.2) e (2.3) respectivamente.

$$\begin{cases} \bar{\mu}_t &= \mathbf{A}_t \mu_{t-1} + \mathbf{B}_t \mathbf{u}_t \\ \bar{\Sigma}_t &= \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t \end{cases} \quad (2.2)$$

$$\begin{cases} \mathbf{K}_t &= \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t &= \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\mu}_t) \\ \Sigma_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t \end{cases} \quad (2.3)$$

O filtro representa o vetor de estados  $\mathbf{s}_t$  no tempo  $t$  por sua média  $\mu_t$  e covariância  $\Sigma_t$ . As matrizes  $\mathbf{R}$ ,  $n \times n$ , e  $\mathbf{Q}$ ,  $l \times l$ , são as matrizes de covariância dos ruídos de processo ( $\gamma$ ) e medição ( $\delta$ ) respectivamente e a matriz  $\mathbf{K}$ ,  $n \times m$ , representa o ganho do filtro.

### Filtro de Kalman Estendido - FKE

Uma das derivações do filtro de Kalman aplicado a sistemas não lineares é o Filtro de Kalman Estendido. A idéia do FKE é linearizar as funções em torno da estimativa corrente usando as derivadas parciais do processo e das funções de medição para calcular as estimativas, mesmo em face a relações não-lineares.

O modelo do sistema não linear para o FKE é dado pelo sistema (2.4):

$$\begin{cases} \mathbf{s}_t &= g(\mathbf{u}_t, \mathbf{s}_{t-1}) + \gamma_t \\ \mathbf{z}_t &= h(\mathbf{s}_t) + \delta_t \end{cases} \quad (2.4)$$

onde  $g(\mathbf{u}_t, \mathbf{s}_{t-1})$  é uma função não-linear que representa o modelo do sistema, e  $h(\mathbf{s}_t)$  é uma função não-linear que representa o modelo das medições.

As etapas de predição e atualização do FKE podem ser obtidas pelos sistemas de equações (2.5) e (2.6) respectivamente.

$$\begin{cases} \bar{\mu}_t &= g(\mathbf{u}_t, \mu_{t-1}) \\ \bar{\Sigma}_t &= \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t \end{cases} \quad (2.5)$$

$$\begin{cases} \mathbf{K}_t &= \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t &= \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \quad (2.6)$$

A matriz  $\mathbf{G}$ ,  $n \times n$ , é o jacobiano que lineariza o modelo, e  $\mathbf{H}$ ,  $l \times n$ , o jacobiano que lineariza o vetor de medições. Tais matrizes são definidas pelas equações (2.7) e (2.8).

$$\mathbf{G}_t = \frac{\partial g(\mathbf{u}_t, \mathbf{s}_{t-1})}{\partial \mathbf{s}_{t-1}} \quad (2.7)$$

$$\mathbf{H}_t = \frac{\partial h(\mathbf{s}_t)}{\partial \mathbf{s}_t} \quad (2.8)$$

### FKD vs FKE

Considere os seguintes algoritmos do Filtro de Kalman Discreto e do Filtro de Kalman Estendido apresentados nos quadros 1 e 2 respectivamente. Comparando-os percebe-se duas diferenças básicas: a linha 3 do quadro 1 com linha 4 do quadro 2 e a linha 4 do quadro 1 com linha 6 do quadro 2. Isso se deve ao fato de o FKE utilizar generalizações não-lineares das matrizes utilizadas no FKD. No filtro estendido o jacobiano  $\mathbf{G}_t$  corresponde às matrizes  $\mathbf{A}_t$  e  $\mathbf{B}_t$  e o jacobiano  $\mathbf{H}_t$  corresponde à matriz  $\mathbf{C}_t$  [Thrun et al. 2005].

---

#### **Algoritmo 1** Filtro de Kalman Discreto.

---

- 01:**  $\text{FKD}(\mu_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$
  - 02:**  $\bar{\mu}_t = \mathbf{A}_t \mu_{t-1} + \mathbf{B}_t \mathbf{u}_t$
  - 03:**  $\bar{\Sigma}_t = \mathbf{A}_t \Sigma_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$
  - 04:**  $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{C}_t^T (\mathbf{C}_t \bar{\Sigma}_t \mathbf{C}_t^T + \mathbf{Q}_t)^{-1}$
  - 05:**  $\mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \bar{\mu}_t)$
  - 06:**  $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \bar{\Sigma}_t$
  - 07:**  $\text{return}(\mu_t, \Sigma_t)$
-

**Algoritmo 2** Filtro de Kalman Estendido.

- 
- 01:**  $\text{FKE}(\mu_{t-1}, \Sigma_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$
  - 02:**  $\bar{\mu}_t = g(\mathbf{u}_t, \mu_{t-1})$
  - 03:**  $\mathbf{G}_t = \frac{\partial g(\mathbf{u}_{t-1}, \mathbf{s}_{t-1})}{\partial \mathbf{s}_{t-1}}$
  - 04:**  $\bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{R}_t$
  - 05:**  $\mathbf{H}_t = \frac{\partial h(\mathbf{s}_t)}{\partial \mathbf{s}_t}$
  - 06:**  $\mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1}$
  - 07:**  $\mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t))$
  - 08:**  $\Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t$
  - 09:**  $return(\mu_t, \Sigma_t)$
- 

## 2.4 Formulação Probabilística do SLAM

Considere um robô móvel que se desloca em um ambiente, fazendo observações relativas de um número desconhecido de marcos utilizando um sensor acoplado a sua estrutura. Sejam definidas em um tempo  $k$  as seguintes grandezas:

- $\mathbf{s}_k$ : o vetor que descreve o estado do robô (posição e orientação);
- $\mathbf{u}_k$ : o vetor de sinais de controle, aplicado no tempo  $k-1$  para conduzir o veículo para um estado  $\mathbf{s}_k$  no tempo  $k$ ;
- $\mathbf{m}_i$ : o vetor que descreve a posição do  $i$ -ésimo marco, assumindo que sua posição é invariante no tempo;
- $\mathbf{z}_{i(k)}$ : uma observação da localização do  $i$ -ésimo marco no tempo  $k$ , tomada do robô.
- $\mathbf{S}_{0:k} = \{\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_k\} = \{\mathbf{S}_{0:k-1}, \mathbf{s}_k\}$ : histórico das poses do robô;
- $\mathbf{U}_{1:k} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\} = \{\mathbf{U}_{1:k-1}, \mathbf{u}_k\}$ : histórico dos sinais de controle aplicados;
- $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_n\}$ : conjunto de todos os marcos;
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_0, \mathbf{z}_1, \dots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$ : conjunto de todas as observações dos marcos.

Na forma probabilística do problema de SLAM, presume-se que o vetor de estados, contendo a pose do robô  $\mathbf{s}_k$  e o mapa  $\mathbf{m}$ , é representado em cada passo de tempo  $k$  por uma Função de Densidade de Probabilidade. Para resolver o problema é necessário encontrar a distribuição de probabilidade *a posteriori* no tempo  $k$ , assumindo que todas as observações  $\mathbf{Z}_{0:k}$  e todos os sinais de entrada  $\mathbf{U}_{1:k}$  a cada tempo  $k$  são conhecidas.

A distribuição de probabilidade procurada deve descrever a localização dos marcos e o estado do robô (no tempo  $k$ ), com base nas observações registradas, nos sinais de controle e no estado inicial do robô. A equação (2.9) define esta distribuição.

$$P(\mathbf{s}_k, \mathbf{m} | \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{s}_0) \quad (2.9)$$

Iniciando com uma estimativa para a distribuição  $P(\mathbf{s}_{k-1}, \mathbf{m} | \mathbf{Z}_{0:k-1}, \mathbf{U}_{1:k-1})$  no tempo  $k-1$ , o conjunto posterior, seguindo um sinal de controle  $\mathbf{u}_k$  e uma observação  $\mathbf{z}_k$ , é calculado utilizando o teorema de Bayes. Este cálculo pressupõe que o modelo de transição de estados e o modelo de observação sejam definidos descrevendo os efeitos do sinal de controle.

O modelo de observação descreve a probabilidade de fazer uma observação  $\mathbf{z}_k$  quando a pose do robô e a localização do marco são conhecidas e geralmente é descrita pela equação (2.10).

$$P(\mathbf{z}_k | \mathbf{s}_k, \mathbf{m}) \quad (2.10)$$

O modelo de movimento do robô pode ser descrito em termos de uma distribuição de probabilidade apresentada na equação (2.11). Isto é, a transição de estado presume ser um processo Markoviano no qual o próximo estado  $\mathbf{s}_k$  depende apenas do estado anterior  $\mathbf{s}_{k-1}$  e do sinal de controle  $\mathbf{u}_k$  aplicado, independente das observações e do mapa.

$$P(\mathbf{s}_k | \mathbf{s}_{k-1}, \mathbf{u}_k) \quad (2.11)$$

É interessante notar que o problema da construção do mapa pode ser formulado como o cálculo da densidade condicional  $P(\mathbf{m} | \mathbf{S}_{0:k}, \mathbf{Z}_{0:k}, \mathbf{U}_{1:k})$ . Isto pressupõe que o estado  $\mathbf{s}_k$  seja conhecido em todos os momentos desde o instante inicial. Um mapa  $\mathbf{m}$  é então construído por fusão de várias observações diferentes. Inversamente, o problema da localização pode ser formulado como o cálculo da probabilidade de distribuição  $P(\mathbf{s}_k | \mathbf{Z}_{0:k}, \mathbf{U}_{1:k}, \mathbf{m})$ . Isto pressupõe que a localização dos marcos é conhecida e o objetivo é calcular uma estimativa da pose do veículo com relação a estes marcos.

O mais importante na visão de SLAM é perceber que as correlações entre as estimativas dos marcos aumentam monotonicamente à medida que mais e mais observações são feitas. Isto significa que o conhecimento da localização dos marcos sempre melhora e nunca diverge, independentemente do movimento do robô. Esta convergência ocorre porque as observações feitas pelo robô podem ser consideradas como medições “quase independentes” da localização relativa dos marcos.

As soluções para o problema de SLAM visam encontrar uma representação adequada, tanto para o modelo de observação quanto para o modelo de movimento do robô. A representação clássica e mais comum é a forma de um modelo de espaço de estado com o ruído gaussiano aditivo. Isto leva ao uso do filtro Kalman para resolver o problema SLAM utilizando um algoritmo conhecido por EKF-SLAM (EKF abreviatura de *Extended Kalman Filter*). Maiores detalhes sobre como utilizar Kalman para resolver o problema de SLAM são apresentados a seguir.

## Modelo do processo

Considere que o estado do sistema é composto pela pose do robô acrescido das características dos marcos. Admitindo que o estado do robô é dado por  $\mathbf{s}_k$ , seu movimento através do ambiente pode ser modelado pela equação (2.12):

$$\mathbf{s}_{k+1} = \mathbf{F}_k \mathbf{s}_k + \mathbf{u}_{k+1} + \mathbf{v}_{k+1} \quad (2.12)$$

onde  $\mathbf{F}_k$  é a matriz de transição de estados,  $\mathbf{u}_k$  o vetor de sinais de controle e  $\mathbf{v}_k$  um vetor de ruídos do processo com média zero e covariância  $\mathbf{R}_k$ .

A localização do i-ésimo marco é denotada  $\mathbf{m}_i$  e, considerando que todos os marcos são estáticos, a equação de estado para o i-ésimo marco é dada pela equação (2.13).

$$\mathbf{m}_{i(k+1)} = \mathbf{m}_{i(k)} = \mathbf{m}_i \quad (2.13)$$

Pode ser visto que o modelo para a evolução dos marcos não tem qualquer incerteza. Assumindo que  $N$  é válido, o vetor de todos os  $N$  marcos é denotado pela equação (2.14).

$$\mathbf{m} = [\mathbf{m}_1^T \quad \dots \quad \mathbf{m}_N^T]^T \quad (2.14)$$

Logo, o vetor de estados aumentado contendo tanto o estado do veículo como todas as características dos marcos é denotado pela equação (2.15)

$$\mathbf{s} = [\mathbf{s}_k^T \quad \mathbf{m}_1^T \quad \dots \quad \mathbf{m}_N^T]^T \quad (2.15)$$

Por fim, o modelo de transição de estados pode ser agora descrito por:

$$\begin{bmatrix} \mathbf{s}_{k+1} \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix} = \begin{bmatrix} \mathbf{F}_k & 0 & \cdots & 0 \\ 0 & \mathbf{I}_{m_1} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \mathbf{I}_{m_N} \end{bmatrix} \begin{bmatrix} \mathbf{s}_k \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_N \end{bmatrix} + \begin{bmatrix} \mathbf{u}_{k+1} \\ \mathbf{0}_{m_1} \\ \vdots \\ \mathbf{0}_{m_N} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{k+1} \\ \mathbf{0}_{m_1} \\ \vdots \\ \mathbf{0}_{m_N} \end{bmatrix} \quad (2.16)$$

onde  $\mathbf{I}_{m_i}$  é uma matriz identidade de dimensão  $\dim(\mathbf{m}_i) \times \dim(\mathbf{m}_i)$  e  $\mathbf{0}_{m_i}$  é um vetor nulo de dimensão  $\dim(\mathbf{m}_i)$ . Como pode ser visto a partir de equação (2.16), as dimensões das matrizes envolvidas foram aumentadas de  $(n * N)$ , sendo  $n$  o número de características necessárias para representar um marco e  $N$  o número de marcos incorporados no mapa.

## Modelo de Observação

Considere que o robô esteja equipado com um sensor que pode fornecer medidas de distância relativa em relação aos marcos. O modelo para a observação do  $i$ -ésimo marco é descrito pela equação (2.17):

$$\mathbf{z}_{i(k)} = \mathbf{H}_{i(k)} \mathbf{s}_k + \mathbf{w}_{i(k)} \quad (2.17)$$

onde  $\mathbf{w}_{i(k)}$  é um vetor de ruído de observação com média zero e variância  $\mathbf{Q}_{i(k)}$ .  $\mathbf{H}_{i(k)}$  é a matriz de observação que relaciona a saída do sensor  $\mathbf{z}_{i(k)}$  com o estado vetor  $\mathbf{s}_k$ , quando observado o  $i$ -ésimo marco. O modelo para essa observação é expresso pela equação (2.18) que reflete o fato das observações serem relativas.

$$\mathbf{H}_i = [-\mathbf{H}_k, \quad 0 \quad \dots \quad 0, \mathbf{H}_{m_i}, \quad 0 \quad \dots \quad 0] \quad (2.18)$$

## Processo de Estimação

Na formulação teórica do problema de SLAM, o filtro de Kalman é usado para fornecer estimativas da pose do robô e da localização dos marcos. O filtro calcula recursivamente as estimativas de um estado  $\mathbf{s}_k$  que evolui de acordo com o modelo do processo e com as observações realizadas sobre o processo.

Ele calcula uma estimativa, que equivale à média condicional  $\hat{s}(p|q) = E[s(p)|\mathbf{Z}^q]$  com  $p \geq q$ , onde  $\mathbf{Z}^q$  é a seqüência de observações tomadas no tempo  $q$  e o erro da estimativa é denotado por  $\tilde{s}(p|q) = \hat{s}(p|q) - s(p)$ .

O filtro opera em duas fases, predição e atualização, e fornece uma estimativa recursiva para a estimativa da covariância  $\Sigma(p|q) = E[\tilde{s}(p|q)\tilde{s}(p|q)^T|\mathbf{Z}^q]$  na estimativa  $\hat{s}(p|q)$ .

**Predição:** Conhecidos os modelos do processo e de observação descritos anteriormente, o algoritmo gera uma previsão  $\hat{\mathbf{s}}(k|k)$  para a estimativa do estado  $\mathbf{s}_k$  no tempo  $k$  juntamente com a estimativa da matriz de covariância  $\Sigma(k|k)$  para um tempo  $k+1$  como mostram as equações (2.19), (2.20) e (2.21) respectivamente.

$$\hat{\mathbf{s}}(k+1|k) = \mathbf{F}_k \hat{\mathbf{s}}(k|k) + \mathbf{u}_k \quad (2.19)$$

$$\hat{\mathbf{z}}_i(k+1|k) = \mathbf{H}_{i(k)} \hat{\mathbf{s}}(k+1|k) \quad (2.20)$$

$$\Sigma(k+1|k) = \mathbf{F}_k \Sigma(k|k) \mathbf{F}_k^T + \mathbf{R}_k \quad (2.21)$$

**Atualização:** Após a previsão e, a partir de uma re-observação  $\mathbf{z}_{i(k+1)}$  do  $i$ -ésimo marco no tempo  $k+1$ , supondo-a correta, pode-se calcular o erro entre a medida estimada e a medida do modelo, bem como uma estimativa da matriz de covariância  $\mathbf{S}_i$ , como mostram as equações (2.22), (2.23) respectivamente.

$$\mathbf{v}_{i(k+1)} = \mathbf{z}_{i(k+1)} - \hat{\mathbf{z}}_i(k+1|k) \quad (2.22)$$

$$\mathbf{S}_{i(k+1)} = \mathbf{H}_{i(k)} \Sigma(k+1|k) \mathbf{H}_{i(k)}^T + \mathbf{Q}_{i(k+1)} \quad (2.23)$$

Por fim, as estimativas do vetor de estados e da matriz de covariância são atualizadas de acordo com as equações (2.24) e (2.25):

$$\hat{\mathbf{s}}(k+1|k+1) = \hat{\mathbf{s}}(k+1|k) + \mathbf{K}_{i(k+1)} \mathbf{v}_{i(k+1)} \quad (2.24)$$

$$\Sigma(k+1|k+1) = \Sigma(k+1|k) - \mathbf{K}_{i(k+1)} \mathbf{S}_{i(k+1)} \mathbf{K}_{i(k+1)}^T \quad (2.25)$$

onde a matriz ganho de Kalman  $\mathbf{K}_{i(k+1)}$ , no tempo  $k+1$  é dada pela equação (2.26).

$$\mathbf{K}_{i(k+1)} = \Sigma(k+1|k) - \mathbf{H}_{i(k)}^T \mathbf{S}_{i(k+1)}^{-1} \quad (2.26)$$

Na solução de SLAM usando Kalman, o vetor de estados é uma das partes mais importantes do sistema, juntamente com a matriz de covariância. A covariância de duas variáveis prevê uma medida de quão fortemente estas duas variáveis são correlacionadas e a correlação é um conceito utilizado para medir o grau de dependência entre as variáveis.

No Kalman, a matriz ( $\Sigma$ ) contém a covariância sobre a pose do robô, a covariância entre a pose do robô e a posição dos marcos e, finalmente, contém a covariância entre os marcos. Esta matriz deve ser inicializada utilizando alguns valores padrões para a diagonal a fim de refletir a incerteza na posição inicial.

O cálculo da matriz  $\mathbf{K}$  serve para encontrar um compromisso entre as medidas da odometria e as medidas dos sensores. Isto é feito usando a incerteza da observação do marco juntamente com a qualidade da medida odométrica. Se os dados da odometria forem mais confiáveis que os dados do sensor no cálculo/utilização de  $\mathbf{K}$ , a prioridade será dada às informações odométricas, caso contrário, a prioridade será dada às informações do marco.

Como foi visto, o filtro de Kalman opera em duas fases: Predição e Atualização. Na primeira etapa deve-se calcular o estado atual do robô usando o seu modelo. Isto é, usam-se os sinais de controle aplicados para calcular uma estimativa da sua nova pose. Na segunda etapa, dado que foi obtida a estimativa para a pose do robô (fase de predição), o objetivo desta fase é compensar os erros oriundos da predição e então calcular sua nova pose de maneira mais precisa usando a informação dos marcos.

O custo computacional para esta solução é elevado pois a cada nova observação deve ser atualizado todo o vetor de estado bem como a matriz de covariância. Este problema se agrava quando a quantidade de marcos é muito grande. Porém, trabalhos mais antigos mostram que implementações funcionam em tempo real em ambiente com um número grande de marcos [Guivant & Nebot 2001, Leonard & Feder 2000].

Embora o algoritmo original de SLAM possua complexidade  $O(n^3)$ , onde  $n$  é o número de elementos do vetor de estados, alguns pesquisadores mostram que otimizações podem ser feitas no sentido de diminuir esta complexidade. Nebot (2002) apresenta, matemática e experimentalmente, uma série de otimizações nas fases de predição e atualização que reduzem a complexidade do algoritmo de SLAM de  $O(n^3)$  para  $O(n^2)$ . Outra otimização, também é apresentada por Nebot (2002), consiste em usar um filtro comprimido que reduz a complexidade do algoritmo para  $O(2 \cdot N_a^2)$ , onde  $N_a$  é o número de marcos na área local. Nesta última, a idéia é manter como fonte de referência para o robô não o mapa como um todo, mas simplesmente uma parte do mapa (chamado de mapa de área local). Com este conceito reduz-se o custo computacional e a complexidade passa a ser proporcional ao número de marcos existente no domínio da área local.

Ainda sobre otimizações, Paz et al. (2007) usaram a estratégia dividir para conquistar e reduziram a complexidade da fase de atualização do algoritmo para  $O(n)$  e, em um trabalho posterior Paz, Tardos & Neira (2008), também usando a estratégia dividir para conquistar, reduzem a complexidade total do algoritmo EKF-SLAM para  $O(n^2)$ .

## Correspondência ou Matching

Em geral, as soluções probabilísticas de SLAM assumem implicitamente que o pareamento entre as características do mapa e as observações reais é conhecido, ou seja, assume-se que o problema da associação de dados já está tratado. Porém, em qualquer implementação real do EKF-SLAM esse problema tem que ser resolvido para cada observação  $\mathbf{z}_k$ .

A correspondência entre as características detectadas está intimamente ligada ao problema de quando introduzir novos elementos no vetor de estados, ou seja, se não houver um pareamento da característica detectada com alguma outra existente no vetor de estados ela é considerada nova e, portanto, é inserida neste vetor.

Um problema ocorre se muitos marcos são inseridos no vetor de estados, pois o número de hipóteses da associação tende a crescer exponencialmente considerando que cada marco deve ser identificado de forma única e inequívoca. Uma técnica padrão usada para resolver este problema de correspondência é o teste de distância de Mahalanobis [Neira & Tardos 2001].

Em estatística, a distância de Mahalanobis é uma medida baseada nas correlações entre variáveis com as quais distintos padrões podem ser identificados e analisados. É uma estatística útil para determinar as similaridades em amostras e distingue-se da distância euclidiana por ser invariante à escala [Manly & Kelly 2005].

Skrzypczynski (2009) comenta que durante a operação de um sistema real de SLAM, uma ambiguidade na associação de dados pode ocorrer mesmo quando passar no teste de Mahalanobis pois podem existir pares estatisticamente viáveis. Assim, um teste adicional usado para contornar este problema consiste em usar a geometria das características detectadas (pontos, linhas, etc.) como informação adicional na etapa de correspondência.

## 2.5 Evolução dos trabalhos em SLAM

A gênese do problema de SLAM ocorreu no *IEEE Robotics and Automation Conference* (1986) realizado em San Francisco, USA. No decorrer desta conferência, enquanto os pesquisadores Peter Cheeseman, Jim Crowley, e Hugh Durrant-Whyte mostraram métodos teóricos de estimativa para os problemas de localização e mapeamento, outros estudiosos da área (Raja Chatila, Oliver Faugeras, Randal Smith) levantaram questionamentos sobre a construção consistente de mapas. Do resultado das discussões concluiu-se que mapeamento probabilístico era um problema fundamental em robótica e que questões conceituais e computacionais precisavam ser tratadas.

Nos anos seguintes, um número expressivo de documentos foram produzidos. Trabalhos como os de Smith & Cheeseman (1987) e Durrant-Whyte (1988), estabeleceram uma base estatística para descrever as relações entre os marcos e manipular geometricamente a região de incerteza. O elemento-chave destes trabalhos foi mostrar que existe um elevado grau de correlação entre as estimativas da localização dos diferentes marcos em um mapa e que essas correlações iriam crescer com as sucessivas observações.

Ao tempo em que Ayache & Faugeras (1988) trabalharam com navegação visual, Crowley (1989) publicou um trabalho de navegação de robôs móveis baseado em sonar e usando filtro de Kalman. Estas duas vertentes de investigação tinham muito em comum com os resultados de Smith et al. (1987). Crowley mostrou que, como um robô móvel se move através de um ambiente desconhecido, tendo observações relativas dos marcos, as estimativas de todos estes marcos são necessariamente correlacionadas umas com as outras por causa do erro comum na estimativa da localização do robô. A implicação deste trabalho foi profunda, pois uma solução completa e coerente para o problema de localização e mapeamento exigiria um conjunto composto pelo estado do robô e todas as posições dos marcos observados. Além disso, após cada marco observado, todo o sistema deveria ser atualizado. Isto exigiria empregar um estimador para um grande vetor de estado o que acarretaria um altíssimo custo computacional.

Essencialmente, o trabalho de Smith et al. (1987) não considerava as propriedades de convergência do algoritmo. Assim, dada a complexidade computacional do problema de mapeamento e sem conhecimento do seu comportamento de convergência, os pesquisadores iniciaram uma série de aproximações para este problema, assumindo, ou mesmo forçando, que as correlações entre os marcos deveriam ser minimizadas ou eliminadas (Leonard & Durrant-Whyte (1992) e Renken (1993), por exemplo). Por estas razões, trabalhos teóricos sobre a localização e mapeamento simultâneos muitas vezes foram tratados como problemas distintos de mapeamento e localização.

O avanço conceitual surgiu com a concretização da combinação do problema de mapeamento e localização pois, uma vez formulado como um único problema de estimação, ele era realmente convergente. Mais importante ainda, foi o reconhecimento que as correlações entre os marcos, que a maioria dos pesquisadores tinham tentado minimizar, eram, na realidade, a parte crítica do problema e que, pelo contrário, quanto mais essas correlações crescam, melhor era a solução.

A estrutura do problema de SLAM e o resultado da convergência foi apresentado pela primeira vez em um *Survey* no *International Symposium on Robotics Research* (1996) [Durrant-Whyte et al. 1996]. A teoria essencial sobre a convergência e muitos dos resultados iniciais foram desenvolvidos por Csorba & Durrant-Whyte (1996) e Csorba (1997).

Posteriormente, os trabalhos passaram a abordar a melhoria da eficiência computacional e a resolução de questões de associação de dados. O *International Symposium on Robotics Research* (1999) foi um importante ponto de encontro quando da realização da primeira sessão sobre SLAM. Neste evento foram apresentados os métodos probabilísticos de mapeamento e localização baseados em filtro de Kalman, introduzidos por Thrun et al. (1998). O *IEEE International Conference on Robotics and Automation* (2000) atraiu vários pesquisadores centrados na questão da complexidade algorítmica do problema de SLAM, bem como nos desafios dos problemas implementacionais. Neste evento, Leonard & Feder (2000) mostram uma forma eficiente para SLAM em grandes ambientes.

Um episódio que marcou a motivação dos pesquisadores para solucionar o problema de SLAM foi a realização do primeiro *1st SLAM-Summer School* em Estocolmo-Suécia no ano de 2002. Este evento foi realizado com muito sucesso e contou com a participação de cerca de 50 pesquisadores da área, merecendo destaque Hugh Durrant-Whyte, John Leonard, Juan Tardós, Raja Chatila, Sebastian Thrun e Wolfram Burgard.

Outras escolas de verão aconteceram posteriormente. Em 2004 Andrews Davison, Paul Newman e Jose Neira juntaram-se aos grandes nomes do *1st SLAM-Summer School* na segunda versão desse evento que foi realizado em Toulouse-França. O *3rd SLAM-Summer School* foi realizado em Oxford-Inglaterra (2006) e ingressaram no grupo dos pesquisadores Dieter Fox, Javier Civera e Simon Lacroix. A quarta e última edição do *SLAM-Summer School* aconteceu em janeiro de 2009 Sidney-Austrália.

Voltando a comentar a evolução dos trabalhos em SLAM, por volta dos anos 2000 Andrews Davison apresentou uma nova vertente para a solução de SLAM utilizando câmera. Ele mostrou resultados utilizando visão ativa [Davison & Murray 2002, Davison et al. 2002] e também utilizando uma única câmera [Davison 2003, Davison et al. 2004].

Em 2005 Thrun et al. comentaram que o atual estado da arte em SLAM concentra-se nas áreas de complexidade computacional, associação de dados e representação do ambiente e em 2006 foram publicadas dois bons tutoriais sobre o tema [Durrant-Whyte & Bailey 2006a, Durrant-Whyte & Bailey 2006b].

Muitas tentativas foram realizadas para reduzir a complexidade dos algoritmos de SLAM. Dentre elas inclui-se a utilização de métodos de sub-mapeamento. Estes métodos exploram a idéia de que um mapa pode ser dividido em regiões com sistemas de coordenadas locais e disposto de uma maneira hierárquica [Estrada et al. 2005].

A abordagem direta para reduzir a complexidade consiste em explorar a estrutura do problema SLAM na reformulação do tempo necessário para fazer as previsões e atualizações. A atualização pode ser reduzida utilizando métodos de aumento de estados e a observação pode ser reduzida usando uma forma de atualização particionada das equações.

Nieto et al. (2005) comentam que a utilização de sub-mapas tem uma série de vantagens. Em particular, ele produz mapas localmente ótimos com complexidade computacional independente do tamanho do mapa completo. Além disso, por tratar atualizações localmente, é numericamente estável, e permite a associação entre várias estruturas.

Estudos mais teóricos sobre a complexidade de algoritmos em SLAM podem ser vistos em Paz et al. (2007) e Paz, Tardos & Neira (2008), enquanto que estudos sobre consistência de algoritmos são apresentados por Huang & Dissanayake (2007), Huang, Mourikis & Roumeliotis (2008) e Huang, Wang & Dissanayake (2008).

Uma segunda barreira importante para ser superada na implementação de métodos de SLAM é a correta associação de marcos quando de uma observação pois uma associação incorreta pode conduzir a falhas catastróficas no algoritmo [Censi et al. 2005, Fang et al. 2006, Chli & Davison 2008, Pink 2008].

No tocante à representação do ambiente, este tema será detalhado no Capítulo seguinte e por isso nos deteremos apenas a comentar que há uma tendência muito forte em utilizar sensores visuais para realizar esta tarefa dentro do SLAM. Imagens fornecem maior riqueza de informações, quando comparadas às medidas dos lasers e sonares, e esta gama de informações pode ser melhor utilizada tanto para calcular e manter atualizada a pose do robô quanto para construir mapas mais detalhados.

Muitas outras informações sobre a evolução dos trabalhos em SLAM poderiam ser citadas, mas finalizaremos esta seção mencionando o trabalho de Ouellette & Hirasawa (2007) que apresenta um quadro comparativo entre diversas técnicas para SLAM usando como sensor principal do robô uma câmera mostrando as vantagens na utilização deste tipo de sensor para resolver o problema. A esta nova forma de solução dá-se o nome de SLAM visual.

## 2.6 Resumo do Capítulo

Este Capítulo objetivou apresentar de forma mais detalhada o problema de Localização e Mapeamento Simultâneos (SLAM) que consiste em um robô móvel usando seus sensores ser capaz de construir um mapa de um ambiente desconhecido e ao mesmo tempo calcular e manter atualizada a sua pose.

Iniciou-se comentando que a correlação entre as medidas dos marcos existentes no ambiente, juntamente com as incertezas inerentes ao movimento do robô e as medidas dos sensores, são a base para a solução probabilística de SLAM. Mencionou-se que os laser, os sonares, as câmeras, e ou as combinações destes, são os principais tipos de sensores que compõem os sistemas de percepção dos robôs utilizados para solucionar o SLAM.

Falou-se que dentre possíveis soluções estatísticas para o problema de SLAM dois grupos de estimadores merecem destaque: o Filtro de Kalman e o Filtro de Partículas. Foi visto também que variações de Kalman como o Kalman Estendido, o Filtro de Informação e o Kalman Unscented, juntamente com a combinação de Kalman e Partículas são também abordagens adotadas para solucionar o SLAM.

Apresentou-se a descrição matemática do Filtro de Kalman Discreto, que é aplicado a sistemas lineares gaussianos, e também detalhou-se sua extensão aplicado a sistemas não lineares gaussianos - Filtro de Kalman Estendido.

Mostrou-se, também, uma formulação probabilística de SLAM de modo que o filtro de Kalman pudesse ser usado para solucioná-lo e, por fim, relatou-se a evolução dos trabalhos relacionados a SLAM e foi mostrado o surgimento de uma nova vertente para resolver o problema utilizando sensores visuais chamada de visual SLAM.

A figura 2.3 ilustra o esquema apresentado neste Capítulo e “introduz” a idéia do SLAM visual. Este tema, juntamente com as questões de representação do ambiente serão detalhados no Capítulo seguinte.

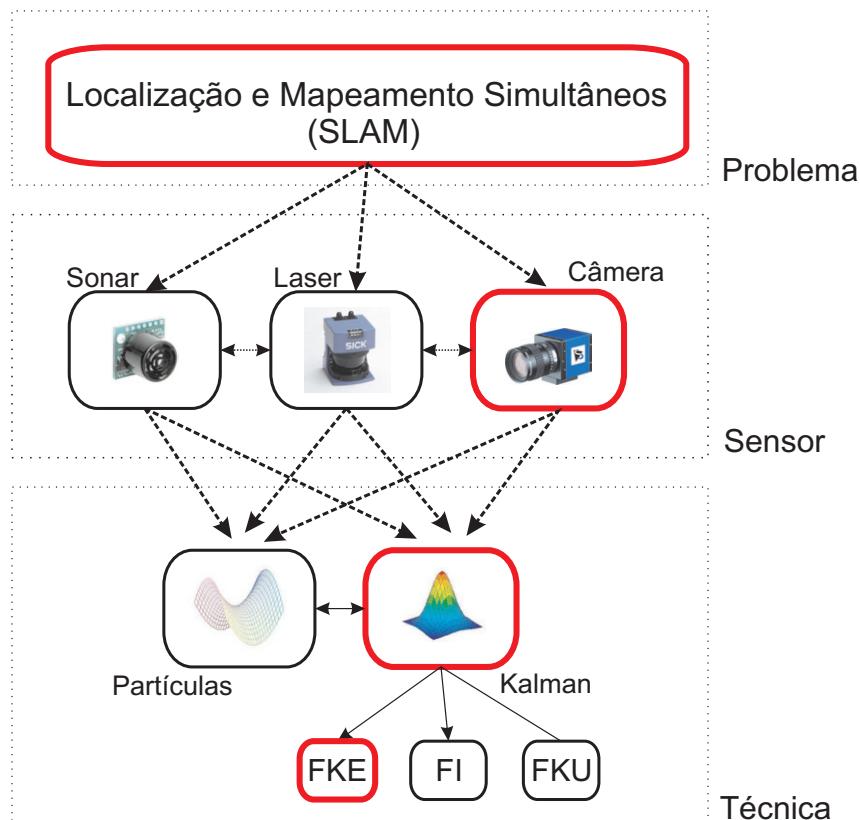


Figura 2.3: Resumo esquemático do Capítulo 2

---

# Capítulo 3

## SLAM Visual

---

*Este Capítulo apresentará soluções para o problema de SLAM utilizando como principal sensor as câmeras digitais. Esta idéia é encorajada pelo baixo custo dos sensores óticos associados com a gama de informações que eles fornecem sobre o ambiente. Apresentaremos o problema de navegação visual sem o conhecimento prévio do mapa para então chegarmos ao SLAM visual. Mostraremos as principais formas de representar os ambientes e, por fim, faremos um relato de vários trabalhos relacionados ao tema.*

### 3.1 Navegação Visual

A navegação, no contexto da robótica, pode ser basicamente descrita como o processo de determinação de um caminho adequado entre um ponto de partida e um ponto de destino para que um robô possa viajar entre eles de forma segura [Choset et al. 2005, Latombe 1991]. Diferentes sensores têm sido utilizados para esta finalidade e isto levou a um variado leque de soluções. Em particular, nas últimas três décadas, a navegação visual para robôs móveis tornou-se uma fonte de inúmeras contribuições para aumentar o âmbito de aplicação de veículos móveis autônomos, tais como: vigilância, patrulhamento, busca e salvamento, inspeção, missões aéreas, mapeamento, detecção de incêndio.

Os seres humanos confiam principalmente nos seus sensores externos, os olhos, para realizar tarefas de localização e navegação. Do mesmo modo em robótica, sensores externos (sensores que medem aspectos do mundo externo) podem render mais informações que os sensores internos. Um exemplo de sensor externo usado nos robôs são as câmeras.

Um atributo essencial para que um robô possa navegar de forma autônoma por um ambiente é que ele possua um sistema de percepção confiável. Além da confiabilidade da percepção, para a aceitação generalizada de aplicações, é necessário que as tecnologias utilizadas proporcionem uma solução com preço acessível, ou seja, que os componentes sejam de baixo custo. Uma solução interessante é a utilização dos sensores óticos.

Devido à utilização maciça de câmeras digitais pessoais, câmeras em computadores e celulares, o preço do sensor de imagem diminuiu significativamente e tornaram-se muito atraentes. Além disso, as câmeras podem ser usadas para resolver uma série de problemas-chave na robótica e em outras operações automatizadas pois fornecem uma variedade de informações do ambiente, consomem pouca energia e são facilmente integradas ao hardware do robô.

Os principais desafios são tirar proveito deste poderoso e barato sensor e criar algoritmos confiáveis e eficazes que possam extrair as informações necessárias para a resolução de problemas na robótica.

## Navegação Sem Mapa

No processo de navegação, um problema surge quando o mapa não está disponível (*mapless navigation*). A navegação sem mapa não necessita do conhecimento do ambiente a ser explorado e os movimentos do robô dependem dos elementos observados no ambiente (paredes, portas, mesas, etc.). Para a solução deste problema dois tipos principais de técnicas devem ser citadas: navegação baseada em aparência e navegação baseada em fluxo ótico [Bonin-Font et al. 2008].

Navegação baseada em aparência é uma técnica baseada no armazenamento de imagens que são usadas como modelos para o robô se auto-localizar e, além disso, são utilizadas para fazer correspondências de observações de marcos. Exemplos iniciais dessa abordagem são os trabalhos de Matsumoto et al. (1996) e Jones et al. (1997) onde uma seqüência de imagens é armazenada para serem usadas como uma memória.

Soluções baseadas em fluxo ótico estimam movimento de objetos ou características dentro de uma seqüência de imagens e os trabalhos iniciais nesta vertente são os de Horn & Schunck (1981) e Lucas & Kanade (1981). Uma abordagem interessante desenvolvida por Santos-Victor & Garibaldi (1993) emula o comportamento de abelhas voando. Neste trabalho um robô contendo duas câmeras apontadas para as paredes se desloca em um corredor e calcula as diferenças dos fluxos ópticos a partir das imagens com o intuito de se mover sempre na direção do fluxo ótico de menor amplitude. O principal problema desta técnica é que as paredes precisam ser texturizadas para apresentar bons fluxos.

Com o avanço na utilização de câmeras, foram dadas outras soluções para o problema da navegação, dentre elas a odometria visual. Esta abordagem está sendo muito utilizada em ambientes outdoor onde a odometria mecânica falha muito devido à derrapagem das rodas. Um exemplo é o trabalho apresentado por Campbell et al. (2004) que calcula a odometria visual utilizando fluxo ótico para diversos ambientes como areia, cascalho e gelo e mostra a eficiência da técnica.

Fernandez & Price (2004) implementam odometria visual da seguinte forma: estimam-se os vetores de fluxo ótico; filtra-os; separa-os em dois grupos, terreno e céu; estima-se as coordenadas x,y com os vetores do grupo terreno; estima-se a coordenada  $\theta$  com os vetores do grupo céu e, por fim, atualiza-se a pose do robô.

Moreira et al. (2007) fundem dois tipos de odometria (mecânica e visual). Eles utilizam filtro de partículas para estimar as velocidades lineares e angulares do robô usando na predição a odometria mecânica e na fase de atualização a odometria visual. A literatura mostra também soluções integrando odometria visual e IMUs (*Inertial Measurement Unit*) como nos trabalhos de Oskiper et al. (2007) e Nourani-Vatani et al. (2008).

## SLAM visual

Até agora comentamos soluções para a navegação visual sem o conhecimento *a priori* do ambiente. Porém, se no decorrer da navegação o robô fosse capaz de construir o mapa do ambiente, este mapa poderia ser utilizado posteriormente. Os mapas podem servir para realizar tarefas consideradas simples, tais como o desvio de obstáculos, planejamento de caminhos e localização, e também outras tarefas consideradas de maior complexidade, como o planejamento de tarefas. Davison & Kita (2001) afirmam que a construção do mapa e a localização simultânea é um grande problema para a navegação visual.

Uma solução para o problema de construir um mapa enquanto navega pelo ambiente, foi apresentada no Capítulo anterior e chamada de SLAM. Nesta técnica, para alcançar adaptabilidade e maior grau de autonomia o robô deve ser capaz de construir um mapa usando seus sensores e, em seguida, utilizar este mapa para saber onde ele está.

Uma abordagem para solucionar o problema de localização e mapeamento simultâneos usando câmera como o sensor principal do sistema de percepção do robô é chamada na literatura de SLAM visual. Esta nova técnica constrói um mapa visual, que consiste normalmente de um conjunto único de marcos identificados pelo robô no seu caminho. Em um ambiente desconhecido, o robô automaticamente identifica marcos, e quando revisita áreas mapeadas, ele utiliza os conhecidos marcos para estimar e melhorar sua pose e a localização dos marcos. Ao atualizar continuamente a posição de marcos e a pose do robô com base em novos dados, incrementalmente melhora-se o mapa calculado.

Os principais tipos de sistemas de percepção visual utilizados pelos robôs são: visão monocular, visão estéreo e visão omnidirecional (figura 3.1).



Figura 3.1: Sistemas visuais de percepção.

Os sistemas de visão monocular enfrentam o problema de estimar corretamente as informações 3D dos marcos a partir de uma imagem. Os sistemas de visão estéreo não possuem este problema mas possuem maior custo computacional. Já os sistemas de visão omnidirecional apresentam como vantagem a imagem panorâmica do ambiente e a desvantagem está associada à deformação que a imagem sofre quando o sistema utiliza um espelho ou o casamento das imagens quando são usados sistemas multi-câmera.

Vale ressaltar que o processamento visual possui dificuldades. Escolher os dados relevantes para o processamento e escolher a estrutura mais adequada para o armazenamento leva ao entendimento das principais formas usadas para representar o ambiente. Este tema será discutido na seção seguinte.

## 3.2 Representações de ambientes

No nível de Mapeamento do Ambiente, o robô deve coletar dados do seu entorno, utilizando sensores, visando gerar modelos computacionais com as principais características estruturais do ambiente. Em outras palavras, é necessário que os robôs sejam equipados com dispositivos de percepção capazes fornecer algum tipo de informação de seu entorno, de modo que, a partir de algum processamento, essas informações sejam usadas para construir um mapa do ambiente.

As primeiras investigações mais aprofundadas na linha de mapeamento robótico surgem em meados da década de 80, quando são publicados os primeiros resultados relevantes das pesquisas sobre o assunto [Chatila & Laumond 1985, Elfes 1987, Elfes 1989]. Surgem várias formas de mapeamento usando diversas estruturas geométricas e topológicas, com o intuito de se obter a melhor precisão possível na representação do ambiente.

Entre os diversos tipos iniciais de representação de mapas, pode-se citar: representação geométrica 3D por esferas [Goldstein et al. 1987], representação por meio de características como pontos, arestas e cantos [Merat & Wu 1987], representação poligonal [de Saint Vincent 1987], modelos geométricos hierárquicos [Kriegman et al. 1987], representação por grade de ocupação [Elfes 1987] e representação baseada na topologia do ambiente [Kuipers & Byun 1988].

Em um trabalho mais recente, em relação aos apresentados, Thrun (2002) propõe uma classificação para as pesquisas em mapeamento robótico, segundo duas abordagens principais: a abordagem métrica e a topológica. A diferença básica entre essas abordagens é que, enquanto os mapas métricos armazenam propriedades geométricas do ambiente, os mapas topológicos descrevem a conectividade entre diferentes lugares.

Alguns autores, porém, preferem detalhar essa classificação acrescentando mais um paradigma denominado de mapas de características [Choset & Fox 2004, Rocha 2006]. Esta representação consiste em armazenar informações métricas de certos objetos notáveis (características) do ambiente. A figura 3.2 ilustra as três formas clássicas apresentadas de representar um ambiente.

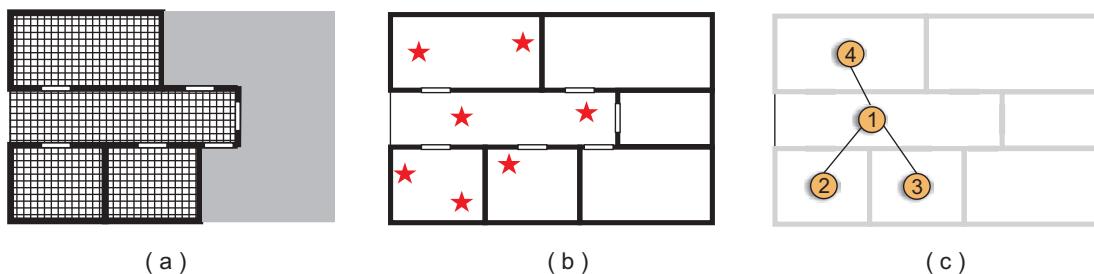


Figura 3.2: Representações do ambiente: (a) Grade de Ocupação; (b) Mapa de Característica; (c) Mapa Topológico.

## Mapas topológicos

Os mapas topológicos são aqueles representados computacionalmente através de um grafo, que, por sua vez, descreve um arranjo de nós (ou vértices) conectados entre si por uma série de arestas (elos ou arcos). Normalmente, os grafos descrevem os espaços livres para a execução de tarefas. Os nós correspondem a regiões que possuem informações sensoriais homogêneas e os arcos refletem a conexão entre estas regiões.

A utilização de um grafo para descrever um ambiente parece ser, intuitivamente, uma boa opção pois além de ser uma estrutura compacta para armazenamento na memória do sistema computacional do robô podem ser usados para resolução de problemas de alto nível como o planejamento de tarefas.

O grande problema desta representação consiste na ausência de um padrão de definição de quais estruturas serão associadas aos vértices e quais relações serão descritas para serem utilizadas como elos. Ainda, a localização do robô usando mapas topológicos é abstrata, ou seja, não há como definir explicitamente a posição e orientação do robô. No entanto, pode-se afirmar em qual nó do grafo ou em qual região do ambiente ele se encontra. A figura 3.3 ilustra intuitivamente o processo de construção de um mapa topológico de um determinado ambiente representado por sua planta baixa.

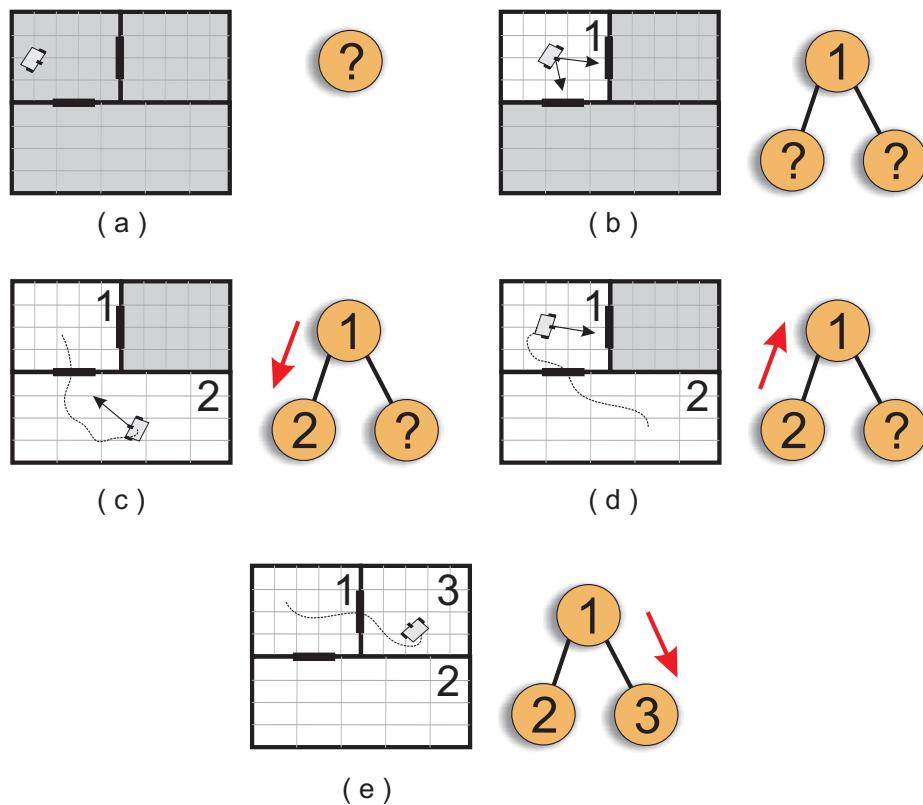


Figura 3.3: Construção de um mapa topológico.

Geralmente, esse tipo de representação produz mapas compactos para processamento e armazenamento e permitem definir de maneira mais simples as tarefas de mais alto nível como, por exemplo, planejamento de rotas por algoritmos tradicionais de busca em grafos.

Porém, a construção de um mapa topológico pode ser fortemente dificultada em função da dimensão do ambiente. Ambientes grandes e complexos podem apresentar informações sensoriais ambíguas, visto que, geralmente, são esparsas, o que pode acarretar em falhas na identificação de compartimentos já mapeados e o não mapeamento de compartimentos ainda não mapeados (associação de dados). Este aspecto dificulta significativamente a construção e manutenção desse tipo de mapa.

Outro problema encontrado nesta representação é a ausência de padronização de quais elementos devem ser considerados nós e arestas dos grafos. Kuipers & Byun (1991) usam os nós do mapa para representar lugares, caracterizados por dados sensoriais, e as arestas para representar caminhos entre os lugares, caracterizadas por estratégias de controle. Thrun (1998) usa um mapa topológico obtido a partir de uma grade de ocupação probabilística particionada em regiões (nós) separadas por passagens estreitas (arestas). Em um trabalho mais recente de Cheong et al. (2008), as definições de nós e arestas são feitas a partir de um método de extração da topologia baseada no Diagrama Generalizado de Voronoi que faz a esqueletização de imagens providas por um laser, para produzir informações topológicas adequadas.

## Mapas métricos

As representações métricas ou mapas métricos reproduzem, com um determinado grau de fidelidade, a geometria do ambiente no qual o robô está inserido. Objetos como paredes, obstáculos e passagens são facilmente identificados nesta abordagem, já que o mapa mantém uma boa relação topográfica com o mundo real. Os mapas métricos são representados por grades de ocupação e pelos mapas de características.

### - Grade de Ocupação

A representação usando grade de ocupação foi proposta inicialmente em 1987 e formalizada em 1989 por Elfes (1989). Com essa forma de representação, os espaços contínuos do ambiente são discretizados, de maneira que o ambiente passa a ser representado sob a configuração de uma grade ou matriz multi-dimensional (2D ou 3D). Cada elemento da matriz, também chamado de célula, representa um local do ambiente que pode estar, de acordo com uma formulação probabilística, ocupado, vazio ou pode ainda não ter sido explorado. A figura 3.4 mostra um exemplo de como seria a representação em uma grade de ocupação de uma parte de um ambiente com dados obtidos por um feixe de sonar.

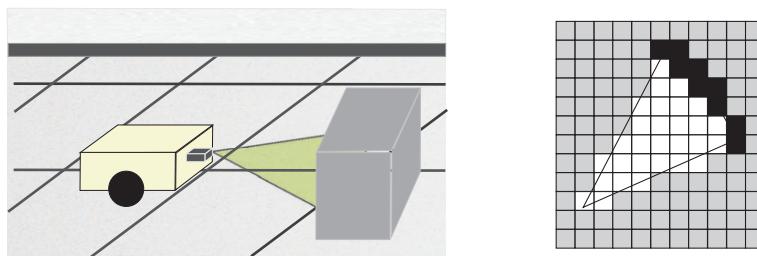


Figura 3.4: Representação de um feixe de sonar em uma grade de ocupação.

Na figura 3.4, as células em preto representam objetos detectados pelo sonar, as células em branco representam regiões livres ou não ocupadas por objetos e, por fim, as células em cinza representam regiões ainda não mapeadas pelo sonar. O modelo espacial probabilístico baseado em grade de ocupação apresenta-se como um mapa que pode ser usado diretamente para tarefas de navegação, tais como planejamento de caminho com desvio de obstáculo e estimativa de posição [Elfes 1989]. A representação em grade de ocupação foi proposta inicialmente para mapear ambientes em uma grade 2D, porém, em um trabalho posterior ao de Elfes, Moravec (1996) expandiu essa representação para uma configuração discreta 3D inovando no tipo de sensor utilizado para a construção do mapa.

Diferente de Elfes, que empregou sonares em suas experiências, Moravec utilizou um sistema de visão estéreo na construção de um mapa 3D. Além disso, Moravec também propôs uma nova abordagem para indicar a possibilidade de ocupação das células, baseada na teoria da evidência de Dempster-Shafer, o que também difere da abordagem probabilística Bayesiana proposta por Elfes. A grade 3D apresentada foi nomeada de grade de evidência (*evidence grid*) e cada célula que a compõe armazena um valor que indica a evidência de ocupação.

Outra variação no uso de grade de ocupação pode ser verificada no trabalho de Yguel, Aycard & Laugier (2006b) que aborda os problemas da representação e armazenamento de dados para mapas grandes e propõe uma forma de representação em grade de ocupação baseada em wavelets: *Wavelet Occupancy Grids*. Hata & Wolf (2010a) propõem a utilização de uma grade 2D construída a partir de núvens de pontos 3D visando identificar regiões navegáveis de um determinado terreno. A partir de informações dessa grade o robô pode inferir níveis de navegação: navegável, parcialmente navegável e não navegável.

Segundo Thrun (1998), os mapas usando representação em grade de ocupação são fáceis de serem construídos e mantidos, mesmo para ambientes complexos ou grandes. A grade de ocupação torna simples o reconhecimento de lugares diferentes com base na posição geométrica do robô dentro de um sistema de coordenadas global. Além disso, se este mapa é criado com resolução fina, permite representar melhor os detalhes do ambiente. Thrun também comenta que a desvantagem da representação em grade de ocupação está na necessidade de espaço de armazenamento maior e também de um maior tempo de processamento, isto é, o seu custo computacional.

### - Mapa de características

Os mapas de características armazenam informações de elementos importantes encontrados em localizações específicas do ambiente. Uma característica, ou *feature*, pode ser entendida como “algo” facilmente notável pelo robô. Além disso, as características devem ser distinguíveis entre si e devem poder ser reobservadas de diferentes posições do ambiente. Em outras palavras, se um robô detecta uma característica em um instante  $t$  e mais tarde, em um instante  $t + k$ , ele a reobserva de uma posição diferente, então esta característica tem que ser identificada de forma fácil e única.

Na construção de mapas de características usando imagens é comum identificar partes com propriedades especiais, por exemplo, pontos, linhas, planos e formas (figura 3.5).

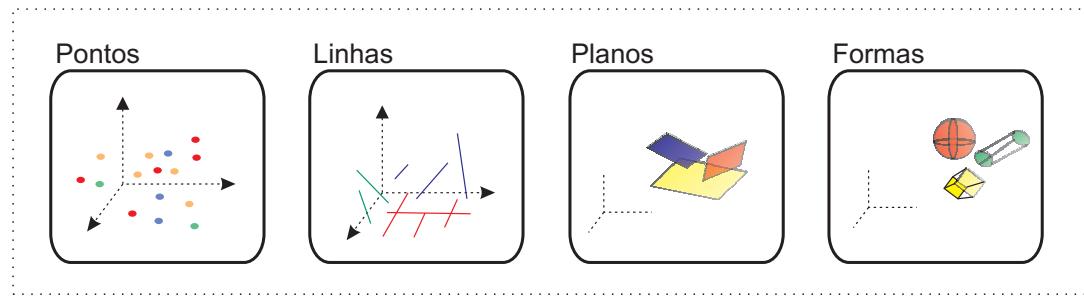


Figura 3.5: Principais tipos de características extraídas em imagens.

A maioria das soluções de SLAM visual utiliza pontos como características do ambiente tanto para abordagens baseadas em visão estéreo [Han et al. 2007, Paz, Pinies, Tardos & Neira 2008] quanto para abordagens com visão monociliar [Civera et al. 2008, Fu & Yang 2009]. Este fato ocorre devido aos vários algoritmos existentes para detectar pontos notáveis em imagem, associado ao fato de pontos serem objetos geométricos simples.

Uma das técnicas mais adotadas para a obtenção de pontos de interesse é o detector Harris (*Harris Corner Detector*) devido à sua forte invariância à rotação, escala, iluminação e ruído de imagem [Harris & Stephens 1998]. Outra técnica bastante utilizada é o algoritmo SIFT (*Scale-Invariant Feature Transform*) que detecta pontos-chave distintivos na imagens e calcula um descritor para eles [Lowe 1999]. Os pontos de interesse extraídos por este algoritmo são ditos invariante a escala, rotação e parcialmente às mudanças na iluminação. Uma variação do algoritmo SIFT, que também é bem utilizado, é chamado de SURF (*Speeded Up Robust Features*) e se baseia na matriz Hessiana devido à sua precisão e baixo custo computacional [Bay et al. 2006]. Para maiores detalhes pode-se citar o trabalho de Gil et al. (2009) que apresenta uma comparação entre os detectores e descritores de pontos mais utilizados no SLAM.

Comentou-se que a construção de mapas de características baseados em pontos é bem aceita, porém estes mapas normalmente não fornecem informações tratáveis e úteis para os humanos. Assim, faz-se necessário a construção de mapas de características utilizando outras formas geométricas, como linhas, por exemplo.

A literatura fornece muitas abordagens para extrair linhas a partir de imagens merecendo destaque a Transformada de Hough. O propósito dessa técnica é encontrar instâncias imperfeitas de objetos dentro de certa classe de formas por processo de votação. Esse processo de votação é efetuado no espaço de parâmetros, no qual objetos candidatos são obtidos como máximo local em uma matriz de acumulação que é construída pelo algoritmo para a computação desta transformada [Forsyth & Ponce 2002].

Se um modelo de mapa usando linhas contém mais informações úteis para os humanos que os que usam pontos, uma extensão natural é adicionar informações planares a estes mapas. Várias contribuições têm sido feitas para esse fim, porém a definição de planos requer mais informações a serem estimadas. Uma transformação clássica para obter mapeamento entre planos é chamada de Homografia. Esta matriz pode ser obtida a partir de dois pontos de vista próximos graças às técnicas de alinhamento de imagem [Baker & Matthews 2001, Aires & Medeiros 2010].

Planos são apenas o primeiro passo para a construção de um mapa com maior nível de representação do ambiente. Em ambientes estruturados muitos objetos podem ser descritos graças a esta primitiva. Porém, os próximos passos de representação do ambiente vão certamente se concentrar no desenvolvimento de mapas mais ricos e com informação semântica. Um trabalho recente que mostra esta tendência é o de Lacroix et al. (2008).

Expressamos até agora a idéia de resolver o problema de localização e mapeamento simultâneos utilizando sensores óticos e também abordamos as formas clássicas de representar ambientes. Tendo em vista esses conceitos juntamente com os outros apresentados no Capítulo anterior passaremos para a seção seguinte mostrando o que a literatura apresenta sobre SLAM visual.

### 3.3 Trabalhos em SLAM visual

Certamente os nomes mais citados, direta ou indiretamente, e que norteiam as pesquisas em SLAM e SLAM visual são:

- ***nos Estados Unidos***: Sebastian Thrun, Kurt Konolige e Dirk Haehnel da Universidade de Stanford; Dieter Fox da Universidade de Washington; John Leonard e Mathew Walter do Instituto Tecnológico de Massachusetts;
- ***na Alemanha***: Wolfram Burgard da Universidade Bonn, Cyrill Stachniss da Universidade Freiburg e Udo Frese da Universidade de Bremen;
- ***na Espanha***: Javier Civera, José Castellanos, Juan Tardós, José Neira, Carlos Estrada e José Martinez, todos da Universidade de Zaragoza;
- ***na Inglaterra***: Andrew Davison, David Murray e Paul Newman, todos da Universidade de Oxford;
- ***outros***: Hugh Durrant-Whyte da Universidade de Sidney-Austrália, Simon Lacroix do LAAS-França e Young-Ho Choi da Korea.

Nesta seção analisaremos inicialmente algumas soluções robóticas usando mais de duas câmeras, como o sistema de visão do robô AQUA que funciona em terra e flutua na água. Este robô usa um sistema de visão trinocular capaz de criar mapas 3D do ambiente e se localizar neste mapa [Georgiades et al. 2004, Georgiades et al. 2005].

Um recente trabalho de Rituerto et al. (2010) mostra uma solução para SLAM usando filtro de Kalman Estendido e um modelo esférico de uma câmera omnidirecional. Scaramuzza & Siegwart (2008) apresentam um algoritmo para a navegação de um veículo em estrada usando informações oriundas de um sistema omnidirecional. Os autores consideram apenas o movimento 2D do veículo e, na primeira fase do algoritmo, detectam pontos usando SIFT, usam RANSAC para eliminar os falsos positivos e para manter apenas pontos presentes no plano de movimento do robô e analisam quais deles obedecem à mesma transformação homográfica. Um experimento de 400m em ciclo fechado é apresentado e mostra um erro acumulado de 6.5m e 5°.

Com a idéia de um sistema multi-câmera, Kaess & Dellaert (2006) montaram um equipamento com oito câmeras defasadas de 45° ao longo de um círculo e utilizaram este artefato em SLAM visual. Os autores relatam que o uso deste tipo de sistema engloba as vantagens dos sistemas de visão monocular e estéreo além de fornecer informação

panorâmica do ambiente de forma mais detalhada quando comparada aos sistema omnidirecionais. Como pontos fracos, eles comentam que o sistema possui elevado custo computacional e não funcionou para o caso de fechamento de ciclo.

Joan Sola et al. (2008) mostram a possibilidade de usar sistemas com qualquer número de câmeras em qualquer configuração fundindo informação monoculares em um único filtro. A vantagem está na possibilidade de trabalhar com câmeras diferentes e não sincronizadas e a desvantagem está na forma centralizada de tratar as informações.

Há na literatura outras soluções para SLAM usando sistemas de visão estéreo [Davison et al. 2002, Jung 2004]. Mansinghka (2004) apresenta um SLAM visual para ambientes dinâmicos baseado na transformada SIFT e fluxo ótico. Nieto et al. (2005) propõem um método de mapeamento hierárquico que permite obter mapas métricos precisos de grandes ambientes em tempo real. O nível inferior do mapa é composto por um conjunto de mapas locais que são estatisticamente independentes. A parte superior do mapa é um grafo de adjacência cujos arcos são rotulados com a relação entre a localização dos mapas locais e uma estimativa relativa destes mapas locais é mantida em uma relação estocástica.

Herath et al. (2006) apresentam um estudo sobre a modelagem dos erros de um sistemas de visão estéreo aplicado a SLAM. Sola et al. (2007) utilizam um sistema com duas câmeras de uma forma diferente. Eles tratam as imagens de forma individual como se fossem dois sistemas monoculares. Embora este trabalho seja conceitual, a idéia é juntar as vantagens do sistema monocular com o sistema estéreo. Dentre os benefícios os autores destacam que as atualizações dos marcos podem ser feita mesmo quando este não é visto pelas duas câmeras, a possibilidade de estimar objetos próximos ao robô por triangulação e a não necessidade da calibração dos parâmetros extrínsecos da visão estéreo.

Outro trabalho, de Agrawal et al. (2007), concentra-se em mostrar como é possível construir um mapa, consistente, globalmente correto, e em tempo real, utilizando algoritmos estéreo para fazer mapas e odometria visual para a localização. Eles esclarecem que a maior contribuição do trabalho está na integração das técnicas utilizadas.

Enquanto Berger & Lacroix (2008) utilizam visão estéreo para mapear o mundo usando áreas planares ao redor de um ponto de interesse (facetas), Marks et al. (2008) propõem o Gama-SLAM, um sistema que funciona em ambientes não estruturados ao ar livre. Ao contrário de outros algoritmos baseados em grade de ocupação probabilística, esta técnica mantém uma distribuição *a posteriori* sobre a variação de altura em cada célula e para obter esta distribuição eles utilizam filtro de partículas. Eles comentam que este tipo de mapa é bem adequado para ambientes externos onde as grades de ocupação muitas vezes fornecem informações suficientes sobre conteúdo de uma célula.

Finalizaremos o relato de trabalhos usando visão estéreo, comentando um trabalho mais recente proposto por Schleicher et al. (2009). O resultado desse trabalho mostra uma solução com mapeamento hierárquico para SLAM visual. Inicialmente são construídos sub mapas locais igualmente espaçados e em seguida um mapa topológico, de mais alto nível, é criado. Os autores comentam que usando esta estratégia hierárquica em dois níveis, pode-se manter a consistência local dos sub mapas usando um mapa de características e a coerência global pode ser obtida pela representação topológica.

Seguiremos agora enfocando os trabalhos mais recentes que utilizam apenas uma câmera para resolver o problema de SLAM, embora soluções mais antigas também existam [Davison 2002, Davison 2003].

Chen & Samarabandu (2006) propõem a utilização de informações geométricas do ambiente e comentam que a redundância em SLAM pode reforçar a confiabilidade e a precisão das características observadas. Assim, eles mostram que primitivas geométricas comuns em ambientes internos, linhas e quadrados, por exemplo, podem ser incorporadas em um Filtro de Kalman para elevar o nível de conhecimento da característica observada.

Vidal-Calleja et al. (2006) propõem um controle ativo para uma câmera em SLAM e Frintrop et al. (2006) introduziram um novo método para detectar marcos. Este método consiste de um sistema de atenção biologicamente inspirado para encontrar regiões contrastantes na imagem. Essa abordagem faz com que regiões sejam facilmente redetectadas e por conseguinte apresentem mais facilidade de correspondência.

Um trabalho mais teórico de Jing Wu & Zhang (2007) apresenta uma modelagem de câmera para visual SLAM. O foco deste trabalho é sobre como modelar a incerteza do sensor ótico e como construir os componentes probabilísticos do modelo da câmera. Tendo o componente determinístico do processo de calibração da câmera (parâmetros intrínsecos), eles são utilizados para reprojeção do erro. Os erros são então encontrados de acordo com uma distribuição Gaussiana bi-variada, a medição da covariância pode ser calculada quando as características estão a diferentes distâncias da câmera.

Choi & Oh (2007) apresentam uma abordagem para a construção de um mapa de grade baseado em visão para ambiente doméstico utilizando uma única câmera. Uma parte interessante deste trabalho é a utilização da derivada de primeira ordem da diagonal de cor de Mahalanobis para a extração de candidatos a pontos marcantes.

Ethan Eade & Drummond (2007) utilizam um grafo para representar o ambiente e minimizam os erros de correspondência a partir de informações locais deste grafo. Clemente et al. (2007) demonstraram pela primeira vez que SLAM com uma única câmera fornecendo dados de entrada pode atingir larga escala ao ar livre funcionando em tempo real. O método utiliza inversão de profundidade e mapas hierárquicos no experimento.

Lemaire et al. (2007) propõem a utilização de linhas 3D como marcos. Eles falam que existem algumas vantagens em utilizar este tipo de representação porque essas primitivas são muito numerosas em ambientes internos; e também porque ao contrário do mapa de pontos esparsos, que só são úteis para fins de localização, um mapa de segmentos relevantes dá informações sobre a estrutura do ambiente. Também utilizando linhas, no caso verticais, Fu et al. (2007a) apresentam um trabalho de fusão de informações de laser e câmera em um filtro de Kalman estendido para SLAM. Neste trabalho as linhas são extraídas da imagem utilizando o detector de bordas Canny.

Baseado na variedade de detectores/descritores utilizados em SLAM visual, Klippenstein & Zhang (2007) apresentam uma comparação entre várias técnicas. As conclusões são que fluxo ótico funciona muito bem em situações em que o deslocamento entre imagens é pequeno e, além disso, apresenta boa imunidade a sintonização de parâmetros. Harris e SIFT podem ser utilizadas quando aumenta a distância entre as imagens. Harris é mais rápido que SIFT, e pode ser utilizado quando a invariância rotacional não é exigida. No que diz respeito à correspondência, Harris e SIFT favorecem a correlação.

Três trabalhos também merecem ser citados. O primeiro é o de Hafez et al. (2008) que apresenta uma otimização na técnica de triangulação para estimar profundidade e funcionar em tempo real. O segundo, o de Calway & Cuevas (2008), mostra mecanismos para descobrir e, em seguida, incorporar alto nível de estrutura tais como linhas e superfícies, no mapa do SLAM visual. O terceiro, de Civera et al. (2008), apresenta uma nova formulação para descrever pontos em SLAM que permite a representação eficiente e precisa das incertezas durante a inicialização. Este trabalho mostra que existe uma estrutura unificada e parametrizada para inicializar características próximas e muito distantes, no âmbito do FKE padrão. A estimativa da incerteza é modelada por uma gaussiana multivariada e, devido à inicialização sem atraso e a utilização plena e imediata de infinitos pontos, as estimativas de orientação da câmera são significativamente melhoradas.

Frintrop & Jensfelt (2008) mostram um sistema visual biologicamente inspirado para detectar características, que privilegia as regiões mais salientes das imagens. Um estudo da repetibilidade destas regiões é apresentado e a conclusão é que as regiões obtidas a partir do sistema proposto se repetem mais quando comparada ao uso de detectores padrão.

Sobre o problema de fechamento de ciclo, Williams et al. (2008) explicam que há duas abordagens clássicas para sistemas que usam uma câmera de mão: imagem-imagem, que consiste em procurar a correspondência entre a última imagem da câmera e as outras já vistas; e imagem-mapa, que procura a correspondência entre a última imagem da câmera e características mapeadas. Eles compararam as duas abordagens e propõem um método imagem-mapa que usa RANSAC para realizar a correspondência.

Chen et al. (2008) apresentam uma solução baseada no algoritmo *Branch and Bound* para corrigir a orientação de um robô fazendo SLAM e Sola et al. (2009) mostram uma modelagem usando coordenadas de Plucket para inicializar de forma direta segmentos de retas obtidos a partir de imagem.

É possível encontrar na literatura trabalhos em SLAM cooperativo. Wu et al. (2009) mostram um algoritmo para uma frota de robôs onde cada um possui uma câmera em sua estrutura. Linhas verticais e placas das portas de um ambiente interno são os marcos visuais deste experimento e uma transformada global é apresentada para juntar os mapas.

Flint et al. (2010) comentam que embora os modernos sistemas de SLAM visual sejam capazes de localizar um robô de forma robusta e eficiente, mesmo no caso de visão monocular, os mapas produzidos por estes algoritmos são geralmente escassos, difíceis de interpretar e de pouca utilidade para os níveis superiores de raciocínio como a compreensão da cena. Assim, os autores propõem uma expansão na competência dos sistemas de SLAM visual para construir mapas mais ricos. Para tal, eles apresentam um rico conjunto de regularidades de ambiente internos que auxiliam na construção de um modelo semântico. Os fragmentos do ambiente são rotulados de “piso”, “parede” e “teto” e esta restrição do espaço diminui as possíveis interpretações da cena e corroboram com a hipótese de mundo de Manhattan. Além desta representação semântica, a última etapa do sistema une os segmentos de linha das partes mapeadas em estruturas de Manhattan inspiradas no trabalho de reconstrução de Lee et al. (2009).

Uma proposta para melhorar a associação de dados no visual SLAM é apresentada por Wang et al. (2010) que usam SURF para detectar características. Os autores afirmam que para combinar os dados de alta dimensão criados pela técnica SURF o método

convencional do vizinho mais próximo não é o mais adequado. Eles propõem uma modificação no algoritmo de Shi-Tomasi para resolver este problema. Outro trabalho baseado em detectores de características é o de Wu et al. (2010). Os marcos utilizados nesta abordagem são pontos e estes são identificados com SIFT. A idéia é selecionar, dentre os identificados, pontos-chave distribuídos uniformemente para utilizá-los no algoritmo de SLAM. A vantagem é diminuir a dimensão do mapa.

Para finalizar esta seção, comentaremos três outros trabalhos que usam linhas como características. Fu & Yang (2009) extraem linhas do teto de um típico ambiente interno e as usam como marcos em um algoritmo de SLAM. Nesse trabalho os segmentos de linhas mapeiam as luzes do teto, RANSAC é usado para agrupá-los e a pose do robô é calculada através da informação do cruzamento de linhas perpendiculares.

Nos trabalho de Zhang & Song (2010) e anterior Zhang & Song (2009) há relatos do uso de linhas verticais (bordas de edifício e postes) para localizar um robô em um ambiente urbano. Os pesquisadores afirmam que a extração deste tipo de marco é robusta às condições de iluminação/sombra além de ser bastante sensível ao movimento de um robô em terreno plano. A abordagem proposta usa múltiplas linhas verticais para localizar o robô através de odometria visual. Uma análise da propagação do erro odométrico é feita e é formulada em termos de uma matriz de covariância. Como pesquisas futuras os autores mostram a intenção de expandir a abordagem usando linhas horizontais, planos verticais e pontos de intercessão entre planos e linhas.

## O uso de sensores visuais no Brasil

Na comunidade robótica brasileira é comum o uso de sensores visuais para a solução de inúmeros problemas de robótica. Alguns nomes consolidados de pesquisadores que utilizam este tipo de sensor nos seus trabalhos são: Adelardo Medeiros, Anna Reali, Denis Wolf, Fernando Osório, Geovany Borges, Geraldo Silveira, Guilherme Pereira, Jun Okamoto, Luiz Marcos, Mário Campos, Mário Sarcineli, Pablo Alsina, Raquel Vassallo, Roseli Romero, Silvia Botelho, Theodiano Bastso, Walter Lages, dentre outros.

Anna Reali, por exemplo, possui trabalhos relacionados a mapeamento visual monocular, reconstrução de planos para ambientes internos, mapas baseados em objetos, correspondência de linhas em imagens, dentre outros [Werneck & Costa 2010a, Werneck & Costa 2010b, Selvatici et al. 2009, Barra et al. 2009].

Denis Wolf e Fernando Osório usam redes neurais para classificação de terrenos a partir de imagens [Hata et al. 2009] enquanto que Geovany Borges identifica linhas em imagens e as utiliza em servo-visão [Bernardes & Borges 2010]. Geraldo Silveira faz SLAM visual usando registro de imagem [Silveira & Rives 2009] e Jun Okamoto resolve este problema usando visão omnidirecional [Guizilini & Okamoto Junior 2009].

Silvia Botelho, por sua vez, publicou diversos trabalhos usando visão computacional para ambientes subaquáticos e também para monitoramento de linhas subterrâneas de distribuição de energia [Drews & Botelho 2009, Botelho et al. 2009, Estrada et al. 2010, Gonçalves et al. 2009].

## 3.4 Resumo do Capítulo

Este Capítulo objetivou apresentar o problema de SLAM usando câmera como sensor principal do robô. Iniciou-se mostrando os aspectos gerais de navegação visual e foi comentado o desafio da navegação sem mapa de modo que pudéssemos mostrar que uma solução para este problema pode ser obtida com SLAM visual.

Mencionou-se que os sistemas de percepção visual dos robôs são normalmente formados por uma câmera (visão monocular), por um par de câmeras (visão estéreo) ou por sistemas que fornecem informações panorâmicas do ambiente (visão omnidirecional).

Proseguiu-se comentando sobre as principais formas de representação dos ambientes enfocando vantagens e desvantagens de cada uma. Comentou-se também que na classificação dos mapas métricos há uma sub-divisão em mapas de grade de ocupação e mapas de características, onde as características normalmente mapeadas são pontos, linhas, planos ou formas.

Por fim, apresentou-se uma série de trabalhos relacionados a SLAM visual e neste contexto foi mostrado que o uso de linhas como características é comumente encontrado. Para “introduzir” a idéia do SLAM visual que será apresentado no capítulo seguinte, apresentaremos na figura 3.6 um resumo esquemático do que foi abordado nesse Capítulo.

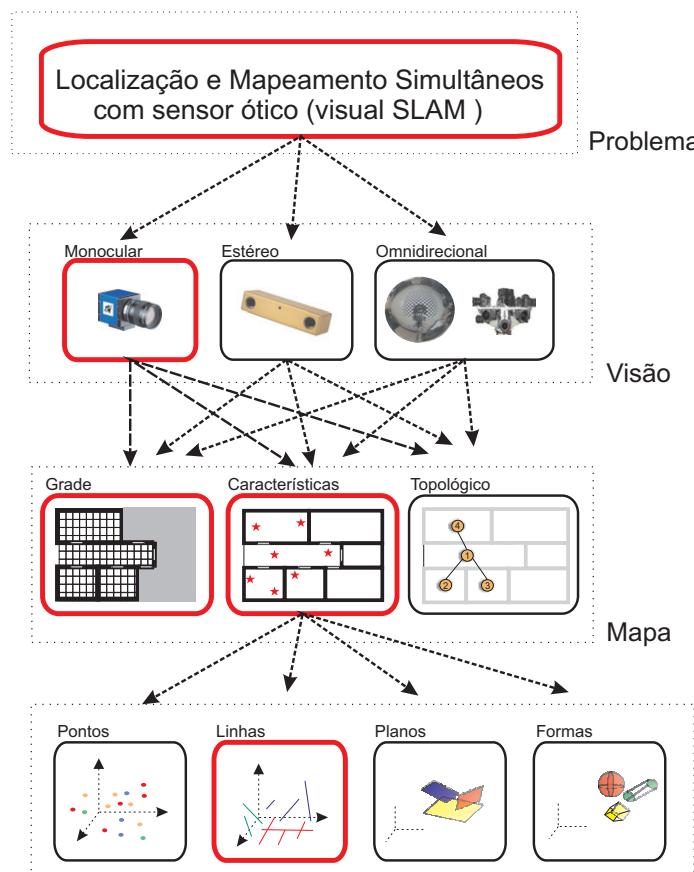


Figura 3.6: Resumo esquemático do Capítulo 3

---

# Capítulo 4

## Sistema Proposto

---

### 4.1 Visão Geral

A execução eficiente de tarefas por robôs está estritamente relacionada ao conhecimento do seu espaço de trabalho. Comentou-se que duas formas clássicas de descrever os ambientes são os Mapas de Características e os Mapas em Grade de Ocupação. A primeira representação possui algumas vantagens como fácil construção e manutenção além de fornecer boas informações para o cômputo da pose do robô. A desvantagem está associada às informações que são armazenadas neste tipo de mapa não serem muito úteis para a execução de tarefas de mais alto nível, como planejamento de caminhos, que normalmente utilizam a segunda forma de representação. Assim, acredita-se que uma solução de SLAM usando um mapa de característica e que simultaneamente constrói um mapa em grade de ocupação pode ser muito útil em aplicações de robôs autônomos.

Apesar do fato de alguns pesquisadores comentarem que o problema de SLAM é considerado resolvido do ponto de vista teórico, muitos trabalhos são publicados todos os anos visando a obtenção de soluções simples e acessíveis de modo a superar muitas das questões práticas que constituem este processo.

O sistema proposto neste trabalho apresenta uma técnica de SLAM visual adequada para ambientes planos com linhas presentes no chão, de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de visão monocular.

Inicialmente é feita uma etapa de segmentação objetivando classificar as partes da imagem em “chão” e “não-chão”. Em seguida, o processador de imagem identifica linhas no chão e os parâmetros destas linhas são usados como marcos no SLAM. Por fim, a pose do robô corrigida juntamente com a parte “não-chão” da imagem são usadas para construir uma grade de ocupação gerando um mapa final híbrido de grade e características.

O algoritmo utilizado para a obtenção do mapa de características é baseado no Filtro de Kalman Estendido (FKE), cuja fase da predição é implementada usando o modelo de odometria do robô. Já na fase de atualização, os parâmetros da representação normal da reta retornados pela Transformada de Hough são mapeados para medidas no mundo usando uma matriz de homografia. Ressalta-se que estes valores são usados diretamente nas equações do filtro de Kalman sem nenhuma etapa intermediária de cálculo de pose ou distância. A figura 4.1 mostra uma visão geral do sistema proposto.

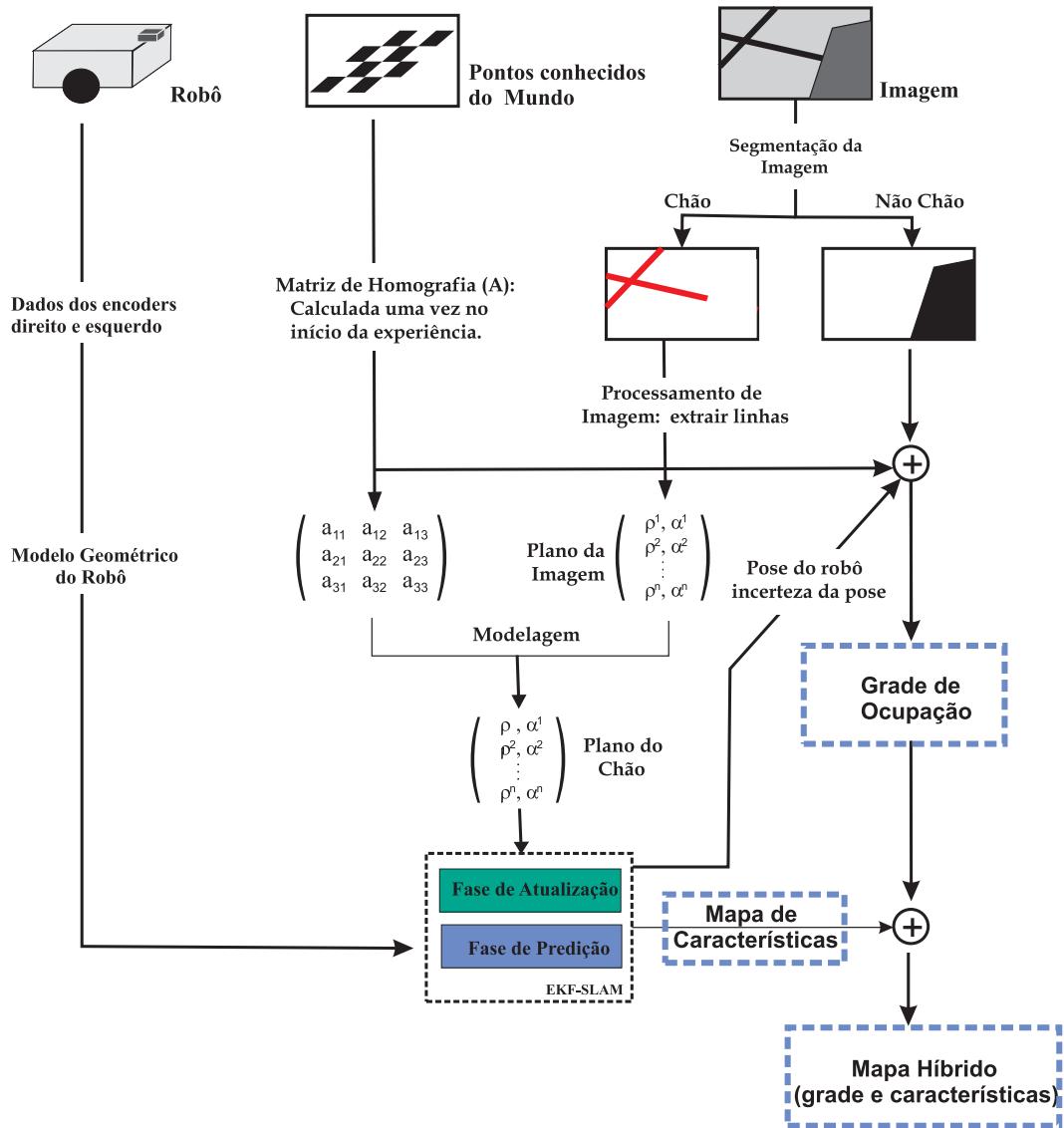


Figura 4.1: Esquema do SLAM visual proposto.

No sentido de obter uma solução simples e eficiente, primamos pelo uso de técnicas clássicas existentes na literatura para vencer individualmente as etapas práticas de um sistema de localização e mapeamento simultâneos. A segmentação usou o algoritmo *Mean Shift* [Yizong 1995], a identificação das linhas é feita usando a *Transformada de Hough* [Hough 1962], o mapeamento imagem-mundo é formulado por uma matriz de *Homografia* [Hartley & Zisserman 2004], a implementação é de um *EKF-SLAM* clássico [Thrun et al. 2005] e a grade de ocupação é construída com as idéias originais de Alberto Elfes [Elfes 1987]. Detalharemos neste capítulo apenas a modelagem do SLAM visual. A construção da grade de ocupação visual será apresentada no Capítulo 6.

### O uso de linhas

Mencionou-se que, do ponto de vista do sistema proposto, as informações que serão mapeadas são as das linhas do chão. Veja que esta não é uma condição muito limitante, pois muitos ambientes, como universidades, shoppings, museus, hospitais, casas e aeroportos, por exemplo, muitas vezes possuem linhas como componentes do piso.

Jeong & Lee (2006) explicam que devido as linhas serem formadas por um conjunto de pontos elas são menos sensíveis a ruídos quando comparadas a pontos individuais, ou seja, o ruído em um ponto pertencente a uma linha geralmente não afeta de forma substancial sua posição e orientação. Além disso, as linhas fornecem boa informação sobre a geometria do ambiente e são facilmente observadas de posições distintas.

Tem-se em mente que os vários desafios em SLAM visual são [Santana & Medeiros 2010]: a) como detectar características em imagens; b) como reconhecer que uma característica detectada é ou não a mesma que uma detectada previamente; c) como decidir se uma nova característica detectada será ou não adotada como um novo marco; d) como calcular a posição 3D de marcos a partir de imagens 2D; e) como estimar a incerteza associada com os valores calculados.

Usando linhas existentes como marcos, a complexidade total do SLAM é reduzida, pois: a) linhas podem ser facilmente detectadas em imagens; b) linhas no chão são geralmente igualmente bem espaçadas, então a possibilidade de confusão é reduzida; c) como o número de linhas na imagem não é tão grande, cada nova linha detectada pode ser definida como um novo marco; d) um chão plano é uma superfície 2D e assim existe uma matriz de conversão constante e fácil de calcular (uma homografia) entre o plano da imagem e o plano do chão, sem incertezas a respeito da informação 3D de profundidade dos pontos; e e) depois do processamento, o número de pixels na imagem que pertencem à linha é uma boa medida de confiança do marco detectado.

## 4.2 Trabalhos Relacionados

Embora tenha sido comentado que nossa proposta é baseada na extração de linhas usando visão monocular, muitos outros trabalhos utilizando diversos tipos de sensores também usam esta primitiva.

Um trabalho mais antigo usando sonar é apresentado por Tardos et al. (2002). Eles coletam as informações do ambiente e posteriormente usam a Transformada de Hough para construir um mapa 2D baseado em linhas. Navarro et al. (2007) fundem dados de um sonar rotativo e informações odométricas em um Filtro de Kalman Estendido. As características do ambiente são classificadas como cantos ou paredes e estas últimas são modeladas por retas cuja representação é feita em coordenadas polares.

Ainda com sonares, Choi et al. (2008) usam as restrições de que as linhas formadas pelas paredes e pelas mobílias que normalmente compõem um ambiente interno são ortogonais ou paralelas. Eles apresentam um algoritmo eficiente de extração de linha que reduz o espaço de busca. O algoritmo é combinando com a Transformada de Hough original e as medidas são usadas em um filtro de Kalman Estendido. Os autores mencionam a possibilidade de mapear o ambiente interno a partir de combinações de características

de linhas, mesmo usando sensores ruidosos. Yap & Shelton (2009) mostram uma solução de SLAM usando sonares esparsos. Neste trabalho é suposto a ortogonalidade entre as paredes de um ambiente interno e para extrair os segmentos de linha de um grupo de  $m$  varreduras de sonar os autores usaram *Randomized Hough Transform*.

Usando laser, Brunskill & Roy (2005) descrevem um fastSLAM que constrói um mapa composto por primitivas geométricas de baixa dimensionalidade usando diretamente os dados do sensor. A redução da dimensionalidade é feita com base em uma abordagem probabilística da Análise de Componentes Principais. Nguyen et al. (2006) mostram um algoritmo de SLAM-2D apropriado para ambientes internos estruturados, o orthoSLAM. Os autores querem explorar o fato da maioria das principais linhas (de paredes, de armários, de portas) são paralelas ou perpendiculares umas com as outras. As linhas são identificadas usando o algoritmo *Split-and-Merge* e são classificadas como horizontais e verticais. A partir desta classificação, apenas uma variável é utilizada para descrever as retas e isto reduz o custo computacional do filtro de Kalman Estendido que é utilizado.

Connette et al. (2007) comentam que a maioria dos ambientes podem ser eficientemente descritos usando segmentos de linha. Porém, eles ressaltam que a incorporação destas medidas muitas vezes é dificultada devido aos segmentos não serem detectados de forma completa, ou seja, os segmentos podem crescer com o passar do tempo. Assim, eles propõem uma descrição estatística do crescimento dos segmentos de linha. Os parâmetros utilizados são decompostos em linha e canto. Eles são monitorados separadamente através de um Filtro de Kalman Estendido e a matriz de covariância cresce à medida que novas regiões de uma mesma linha são descobertos.

Para finalizar o relato de trabalhos usando laser e linhas, comentaremos dois: o de Armesto et al. (2008) e um mais novo de Cheein et al. (2010). O primeiro apresenta uma forma de filtragem multi-taxa assíncrona e descreve um método baseado em mínimos quadrados para detectar linhas em ambiente internos. Já o segundo, mostra um algoritmo de SLAM aplicados à auxílio robótico para a navegação em ambientes desconhecidos. Neste, um robô semi-autônomo controlado por sinais eletromiográficos navega e constrói um mapa de características formado por linhas e cantos. As linhas representam as paredes do ambiente e os cantos são os pontos de intercessão das paredes.

A literatura também mostra trabalhos usando linhas extraídas a partir de imagem. Exemplificando, Dailey & Parnichkun (2005) e Ahn, Lee, Chung & Sang-Rok Oh (2007) usam visão estéreo enquanto Bosse et al. (2005) e Kim & Oh (2008) usam visão omnidirecional. Serão apresentados a seguir apenas trabalhos onde o sistema de visão do robô é monocular.

Dao et al. (2003) apresentam a idéia de usar grupos de três ou mais linhas para auto-localização de um robô considerando que o piso é plano e que a câmera é perpendicular ao piso. Eles comentam que usaram linhas por serem primitivas de rápida e fácil detecção, abundantes em ambiente internos e fáceis de serem rastreadas. As linhas são identificadas com detector de bordas Canny e o rastreamento é feito utilizando fluxo ótico. Folkesson et al. (2005) usam linhas obtidas por uma câmera olhando para o teto. Nesse trabalho, na primeira observação de uma linha é calculada uma estimativa da sua orientação, mas, a profundidade é desconhecida e posteriormente calculada por triangulação.

Dois trabalhos que mostram como selecionar com eficiência, observar e avaliar as características locais em um sistema de SLAM monocular em tempo real são apresentados por Eade & Drummond (2006) e Eade & Drummond (2009). Além disso, eles descrevem como inicializar os novos marcos, como observar marcos mapeados em imagens posteriores e como lidar com os desafios de dados de associação de linhas.

Smith et al. (2006) consideram que pontos e retas são complementares em um sistema de localização de câmera. Eles comentam que pontos são bem caracterizados, porém, são dependentes dos pontos de vista/observação, enquanto que as linhas não sofrem este problema. Foi apresentado um algoritmo rápido para a detecção de retas que é baseado em cantos para formar um conjunto de retas candidatas. Os autores ainda descrevem como as linhas podem ser facilmente incluídas em um filtro de Kalman Estendido de modo que sejam fundidas as informações dos pontos e das retas detectadas.

Um SLAM baseado em filtro de Kalman Estendido para localizar e reconstruir linhas 3D e cantos ao mesmo tempo e em tempo real é a proposta de Jeong & Lee (2006). O algoritmo apresentado distingue primeiramente linhas horizontais e verticais em uma imagem de entrada e constrói um mapa de características 3D de acordo com uma estimativa adequada da condição inicial. Para inicializar a reta, dois pontos são identificados e rastreados no quadro seguinte. Kitanov et al. (2007) apresentam um trabalho focado na construção de um modelo de ambientes internos complexos. Os autores usam dados de odometria para uma estimativa da pose do robô e as linhas são extraída usando *Random Window Randomized Hough Transform*.

Lemaire & Lacroix (2007) mostram um método para incorporar segmentos de linha 3D em SLAM visual. Coordenadas de Plücker são utilizadas para representar as linhas e um método baseado em uma aproximação gaussiana é usado para inicializar os segmentos. Uma vez inicializado, o marco é estimado ao longo do tempo em um filtro de Kalman Estendido que incorpora, na fase de atualização, as limitações associadas à representação. Ainda trabalhando com segmentos de reta, Gee & Mayol-Cuevas (2007) usam filtro de Kalman Unscented para resolver o problema da inicialização e Neubert et al. (2008) apresentam um algoritmo rápido para rastreamento de segmentos de retas em imagens.

Gee et al. (2008) descrevem um método para descobrir e incorporar estruturas de nível superior em um mapa de SLAM visual. Os autores explicam que as abordagens clássicas que usam pontos ou linhas facilitam muito a localização, porém, produzem uma representação da cena limitada e ignoram as redundâncias inerentes às características da estrutura física da cena. Assim, eles mostram uma forma de descobrir uma estrutura de nível superior, na forma de linhas e planos, que são descobertos simultaneamente com o SLAM em operação e, em seguida, incorporam-nas ao mapa tentando manter as informações de covariância cruzada para permitir a atualização consistente dos parâmetros.

Uma solução de SLAM visual para ambiente interno baseada no Filtro de Kalman Estendido usando retas verticais e inversão de profundidade é proposto por Li et al. (2008). As linhas verticais são perpendiculares ao plano terreno e são representadas na forma de profundidade inversa de modo a permitir uma eficiente e precisa representação das incertezas. Além disso, esta formulação aumenta a velocidade de computação do algoritmo de SLAM em comparação com o de linha de direção arbitrária.

Stuckler & Behnke (2008) abordam o problema da correção da orientação de um robô usando as paredes de um ambiente. Os autores explicitam que em ambientes internos o piso pode ser facilmente distinguido das paredes por características simples como cor ou textura. Eles separam o piso e utilizam apenas as paredes considerando que elas são perpendiculares ao plano do chão e modeladas como retas.

Trabalhos mais recentes também mostram o uso de linhas. Wu et al. (2009) apresentam um algoritmo para uma frota de robôs onde cada um possui uma câmera montada em sua estrutura. Linhas verticais e placas das portas de um ambiente interno são os marcos visuais deste experimento e uma transformada global é apresentada para juntar os mapas. Fu & Yang (2009) extraem linhas do teto de um típico ambiente interno e as usam como marcos em um algoritmo de SLAM. Nesse trabalho os segmentos de linhas mapeiam as luzes do teto, RANSAC é usado para agrupá-los e a pose do robô é calculada através da informação do cruzamento de linhas perpendiculares.

No trabalho de Zhang & Song (2010) e em um anterior, Zhang & Song (2009), há relatos do uso de linhas verticais (bordas de edifício e postes) para localizar um robô em um ambiente urbano. Os autores comentam que a extração deste tipo de marca é robusta às condições de iluminação/sombra além de ser bastante sensível ao movimento de um robô em terreno plano. A abordagem proposta usa múltiplas linhas verticais para localizar o robô através de odometria visual. Uma análise da propagação do erro odométrico é feita e é formulada em termos de uma matriz de covariância. Como trabalhos futuros, os autores mostram a intenção de expandir a abordagem usando linhas horizontais, planos verticais e pontos de intercessão entre planos e linhas.

Até agora mostramos uma série de trabalhos baseados nos seguintes aspectos: uso de linhas como características para descrever ambientes, uso do Filtro de Kalman Estendido como base para o SLAM e uso da Transformada de Hough e suas variações para extrair retas. Para finalizar esta seção mostraremos outras pesquisas onde o uso de uma matriz homográfica é aplicada a robótica.

Guerrero et al. (2005) mostram a navegação sem mapa de uma cadeira de rodas onde imagens são usadas para corrigir o movimento do robô, usando linhas como principais características. As linhas são automaticamente extraídas e combinadas e uma homografia é usada como base para corrigir a pose do robô. Kim & Eusticey (2009) mostram uma aplicação de SLAM para o problema de inspeção automática de cascos de navios. Eles utilizam uma combinação de SIFT e Harris para detectar pontos e como a superfície do casco do navio pode ser considerada localmente planar os pontos detectados são usados para o cálculo de uma matriz de homografia durante o registro da imagem.

Por fim, no trabalho mais recente, Servant et al. (2010) discorrem sobre a fusão de dados de IMU e visão monociliar em um Filtro de Kalman Estendido a fim de obter localização e mapeamento simultâneos. Os marcos utilizados são planos e parâmetros da matriz de homografia são usados na modelagem desta primitiva.

Vito uma série de trabalhos que influenciaram, direta ou indiretamente, a elaboração desta tese; mostraremos, a seguir, as formulações matemáticas utilizadas nessa proposta.

## 4.3 Fundamentação Teórica

Nossa proposta é fundamentada no uso do três técnicas clássicas: o EKF-SLAM para estimar a pose do robô e as características do mapa de características; a Transformada de Hough para detectar linhas em imagens; e uma transformação homográfica para fazer o mapeamento imagem-mundo. Apresentaremos a seguir cada uma dessas técnicas.

### EKF-SLAM

O Filtro de Kalman Estendido trabalha com um modelo segundo o sistema (4.1), cujas variáveis são descritas na Tabela 4.1. Os sinais  $\epsilon_t$  e  $\delta_t$  são ruídos gaussianos de média zero e as fases de predição e atualização do filtro são regidas pelos sistemas (4.2) e (4.3).

$$\begin{cases} \mathbf{s}_t = p(\mathbf{s}_{t-1}, \mathbf{u}_t, \boldsymbol{\epsilon}_{t-1}) \\ \mathbf{z}_t = h(\mathbf{s}_t) + \boldsymbol{\delta}_t \end{cases} \quad (4.1)$$

$$\begin{cases} \bar{\mu}_t = p(\mu_{t-1}, \mathbf{u}_t, 0) \\ \bar{\Sigma}_t = \mathbf{G}_t \Sigma_{t-1} \mathbf{G}_t^T + \mathbf{V}_t \mathbf{M}_t \mathbf{V}_t^T \end{cases} \quad (4.2)$$

$$\begin{cases} \mathbf{K}_t = \bar{\Sigma}_t \mathbf{H}_t^T (\mathbf{H}_t \bar{\Sigma}_t \mathbf{H}_t^T + \mathbf{Q}_t)^{-1} \\ \mu_t = \bar{\mu}_t + \mathbf{K}_t (\mathbf{z}_t - h(\bar{\mu}_t)) \\ \Sigma_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \bar{\Sigma}_t \end{cases} \quad (4.3)$$

Tabela 4.1: Símbolos das equações (4.1), (4.2) and (4.3)

$\mathbf{s}_t$	vetor de estados (ordem $n$ ) no instante $t$
$p(\cdot)$	modelo não linear do sistema
$\mathbf{u}_t$	sinais de controle (ordem $l$ ), no instante $t$
$\boldsymbol{\epsilon}_{t-1}$	ruído do processo (ordem $q$ ), no instante $t - 1$
$\mathbf{z}_t$	vetor de observações (ordem $m$ ) retornadas pelo sensor
$h(\cdot)$	modelo não linear do sensor
$\boldsymbol{\delta}_t$	ruído de medição
$\bar{\mu}_t, \mu_t$	média (ordem $n$ ) do vetor de estados $\mathbf{s}_t$ , antes e depois da fase de atualização
$\bar{\Sigma}_t, \Sigma_t$	covariância ( $n \times n$ ) do vetor de estados $\mathbf{s}_t$
$\mathbf{G}_t$	matriz Jacobiana ( $n \times n$ ) que lineariza o modelo do processo $p(\cdot)$
$\mathbf{V}_t$	matriz Jacobiana ( $n \times q$ ) que lineariza o ruído do processo $\boldsymbol{\epsilon}_t$
$\mathbf{M}_t$	covariância ( $q \times q$ ) do ruído de processo $\boldsymbol{\epsilon}_t$
$\mathbf{K}_t$	matriz ganho de Kalman ( $n \times m$ )
$\mathbf{H}_t$	matriz Jacobiana ( $m \times n$ ) que lineariza o modelo do sensor $h(\cdot)$
$\mathbf{Q}_t$	covariância ( $m \times m$ ) do ruído de medição $\boldsymbol{\delta}_t$

No EKF-SLAM, além de estimar a pose do robô, também são estimadas as características de todos os marcos encontrados pelo caminho. Assim, torna-se necessário incluir as características dos marcos no vetor de estado. Se  $^i c$  é o vetor de características do i-ésimo marco e existem  $k$  marcos, então o vetor de estados é dado pela equação (4.4).

$$\mathbf{s}_t = \left[ x_t \ y_t \ \theta_t \ {}^1 c_t^T \ \dots \ {}^k c_t^T \right]^T \quad (4.4)$$

### Transformada de Hough (TH)

A Transformada de Hough foi desenvolvida em 1962 por Paul Hough [Hough 1962]. De uma forma geral, o propósito dessa técnica é encontrar instâncias imperfeitas de objetos dentro de certa classe de formas por processo de votação [Gonzalez & Woods 2007].

A idéia da TH, quando aplicada a imagens digitais, consiste em fazer um mapeamento do espaço  $(u, v)$  para o espaço de representação paramétrico de uma curva (retas, círculos, elipses). Para isso, o espaço de parâmetros é discretizado e representado na forma de uma matriz de acumuladores onde cada posição desta matriz corresponde a um intervalo no espaço de parâmetros. Um processo de votação é efetuado no espaço de parâmetros e os objetos candidatos são obtidos como o máximo local na matriz de acumuladores.

Duda e Hart (1972) adaptaram a idéia de Hough para a obtenção de linhas em imagens digitais usando coordenadas polares para descrevê-las (equação 4.5).

$$\rho = u \cos(\alpha) + v \sin(\alpha) \quad (4.5)$$

Nesta abordagem, cada pixel  $(u, v)$  da imagem representa uma curva no espaço de parâmetros  $(\rho, \alpha)$  e quando pontos pertencem à mesma reta, as curvas por eles representadas se interceptam no domínio de parâmetros (ver figura 4.2). Computacionalmente, quando uma função Transformada de Hough é usada para detectar linhas, seu retorno são os parâmetros  $(\rho, \alpha)$  para cada reta obtida na imagem [Gonzalez & Woods 2007].

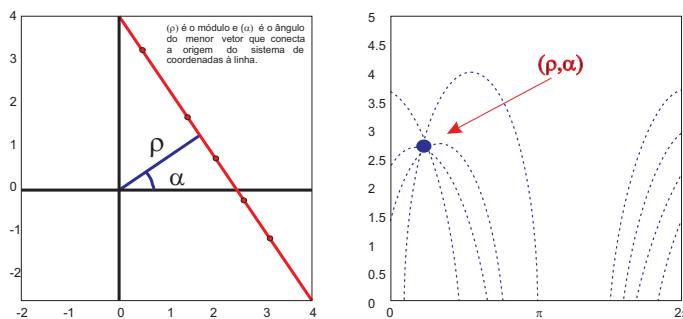


Figura 4.2: Transformada de Hough.

Do ponto de vista prático, Forsyth & Ponce (2002) elencam como problemas da Transformada de Hough a quantização da matriz de acumuladores e a influência do ruído da imagem. Eles propõem como soluções para esses problemas a utilização de um detector de bordas para suavizar a textura e aumentar o contraste das bordas e escolher o tamanho da matriz de acumuladores com cuidado e de acordo a sensibilidade da aplicação.

### Homografia

Em visão computacional, define-se homografia planar como um mapeamento projetivo de um plano para outro [Bradski & Kaehler 2008]. Assim, o mapeamento de pontos em uma superfície plana de duas dimensões para o plano da imagem de uma câmera é um exemplo de homografia planar.

Matematicamente é possível expressar esse mapeamento em termos de multiplicação de matrizes se usarmos coordenadas homogêneas para expressar tanto o ponto no mundo  $Q=[x, y, 1]^T$  quanto o ponto na imagem  $q=[u, v, 1]^T$  que é mapeado a partir de  $Q$ . Então, pode-se expressar a ação da matriz de homografia  $\mathbf{A}$  como mostra a equação (4.6) considerando  $s$  como um fator de escala arbitrário.

$$q \simeq s \cdot \mathbf{A} \cdot Q \quad (4.6)$$

## 4.4 Modelagem

### Fase de Predição: Modelo do Processo

Considere um robô com acionamento diferencial em que  $\Delta\theta_R$  e  $\Delta\theta_L$  são deslocamentos angulares direito e esquerdo das rodas respectivamente (figura 4.3).

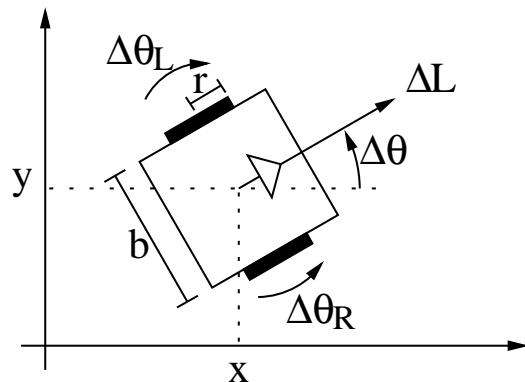


Figura 4.3: Variáveis do modelo geométrico do robô.

Assumindo que a velocidade pode ser considerada constante durante um período de amostragem, pode-se determinar o modelo geométrico cinemático do movimento do robô pelo sistema (4.7):

$$\begin{cases} x_t = x_{t-1} + \frac{\Delta L}{\Delta\theta} [\sin(\theta_{t-1} + \Delta\theta) - \sin(\theta_{t-1})] \\ y_t = y_{t-1} - \frac{\Delta L}{\Delta\theta} [\cos(\theta_{t-1} + \Delta\theta) - \cos(\theta_{t-1})] \\ \theta_t = \theta_{t-1} + \Delta\theta \end{cases} \quad (4.7)$$

no qual:

$$\begin{cases} \Delta L = (\Delta\theta_R r_R + \Delta\theta_L r_L)/2 \\ \Delta\theta = (\Delta\theta_R r_R - \Delta\theta_L r_L)/b \end{cases} \quad (4.8)$$

onde,  $\Delta L$  e  $\Delta\theta$  são os deslocamentos linear e angular do robô;  $b$  representa a distância entre as rodas e  $r_R$  e  $r_L$  são os raios da roda direita e esquerda, respectivamente. Porém, quando  $\Delta\theta \rightarrow 0$ , o modelo pode ser descrito pelo sistema (4.9), obtido do limite do sistema (4.7).

$$\begin{cases} x_t = x_{t-1} + \Delta L \cos(\theta_{t-1}) \\ y_t = y_{t-1} + \Delta L \sin(\theta_{t-1}) \\ \theta_t = \theta_{t-1} \end{cases} \quad (4.9)$$

Foi adotada a abordagem defendida por Thrun et al. (2005), que considera a informação de odometria como sinais de entrada a serem incorporados ao modelo do robô, ao invés de classificá-las como medições sensoriais.

As diferenças entre o deslocamento angular real das rodas ( $\Delta\theta_R$  e  $\Delta\theta_L$ ) e os deslocamentos medidos pelos encoders ( $\Delta\tilde{\theta}_R$  e  $\Delta\tilde{\theta}_L$ ) são modelados por um ruído branco gaussiano de média zero, de acordo com o sistema (4.10). As medidas  $\Delta\tilde{L}$  e  $\Delta\tilde{\theta}$  são definidas substituindo-se ( $\Delta\theta_R$  e  $\Delta\theta_L$ ) por ( $\Delta\tilde{\theta}_R$  e  $\Delta\tilde{\theta}_L$ ) no sistema (4.8).

$$\begin{cases} \Delta\theta_R = \Delta\tilde{\theta}_R + \varepsilon_R \\ \Delta\theta_L = \Delta\tilde{\theta}_L + \varepsilon_L \end{cases} \quad (4.10)$$

Se o vetor de estados  $\mathbf{s}$  é dado pela equação (4.4), o modelo do sistema  $p(\cdot)$  pode ser obtido do sistema (4.7) ou (4.9) pelo fato das coordenadas dos marcos  ${}^i c$  serem estáticas:

$$p(\cdot) = \begin{cases} x_t = \dots \\ y_t = \dots \\ \theta_t = \dots \\ {}^1 c_t = {}^1 c_{t-1} \\ \vdots \\ {}^k c_t = {}^k c_{t-1} \end{cases} \quad n = 3 + k \cdot \text{ordem}({}^i c) \quad (4.11)$$

$$\mathbf{u}_t = [\Delta\tilde{\theta}_R \Delta\tilde{\theta}_L]^T \quad l = 2 \quad (4.12)$$

$$\varepsilon_t = [\varepsilon_R \varepsilon_L]^T \quad q = 2 \quad (4.13)$$

assim, as matrizes  $\mathbf{G}$  e  $\mathbf{V}$  podem ser obtidas derivando o modelo  $p(\cdot)$  como segue:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & \frac{\Delta\tilde{L}}{\Delta\tilde{\theta}}[\cos(\theta_{t-1} + \Delta\tilde{\theta}) - \cos(\theta_{t-1})] & 0 & \cdots & 0 \\ 0 & 1 & \frac{\Delta\tilde{L}}{\Delta\tilde{\theta}}[\sin(\theta_{t-1} + \Delta\tilde{\theta}) - \sin(\theta_{t-1})] & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \quad (4.14)$$

$$\mathbf{V} = \begin{pmatrix} V_1 \cos(\beta) + V_2 [\sin(\beta) - \sin(\theta_{t-1})] & -V_1 \cos(\beta) + V_3 [\sin(\beta) - \sin(\theta_{t-1})] \\ V_1 \sin(\beta) - V_2 [\cos(\beta) - \cos(\theta_{t-1})] & -V_1 \sin(\beta) - V_3 [\cos(\beta) - \cos(\theta_{t-1})] \\ r_R/b & -r_L/b \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{pmatrix} \quad (4.15)$$

onde:

$$\beta = \theta_{t-1} + \frac{r(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)}{b}; V_1 = \frac{r(\Delta\tilde{\theta}_R + \Delta\tilde{\theta}_L)}{2(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)}; V_2 = \frac{-b\Delta\tilde{\theta}_L}{(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)^2}; V_3 = \frac{b\Delta\tilde{\theta}_R}{(\Delta\tilde{\theta}_R - \Delta\tilde{\theta}_L)^2}$$

Sabe-se que a odometria introduz erros acumulativos. Portanto, o desvio padrão dos ruídos  $\epsilon_R$  e  $\epsilon_L$  é assumido ser proporcional ao módulo do deslocamento angular medido para cada roda. Essas considerações levam à definição da matriz  $\mathbf{M}$  dada pela equação (4.16).

$$\mathbf{M} = \begin{pmatrix} (M_R|\Delta\tilde{\theta}_R|)^2 & 0 \\ 0 & (M_L|\Delta\tilde{\theta}_L|)^2 \end{pmatrix} \quad (4.16)$$

### Fase de Predição: Modelo do Sensor

Considere  $\mathbf{A}_{3 \times 3}$  uma matriz de Homografia e  $\mathbf{C}_{3 \times 3}$  a Matriz de Cofatores de  $\mathbf{A}$ , onde  $C_{ij} = (-1)^{i+j} \cdot \det(\mathbf{A})$  eliminando a linha  $i$  e a coluna  $j$ .

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad e \quad \mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \quad (4.17)$$

Sabe-se que um ponto  $(x^I, y^I, 1)$  pode ser mapeado em outro plano utilizando a relação expressa na equação (4.18). A matriz  $\mathbf{A}$  só pode ser determinada com um fator de escala e esse fator  $s$  pode ser obtido sabendo que o último elemento do vetor de coordenadas homogêneas é 1.

$$s \cdot \begin{pmatrix} x^I \\ y^I \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x^{II} \\ y^{II} \\ 1 \end{pmatrix} \quad (4.18)$$

Usando a equação paramétrica da reta, um ponto  $(x, y) \in \text{reta}(\rho, \alpha)$  pode ser definido pela equação (4.19):

$$\begin{cases} x = \rho \cdot \cos(\alpha) - \lambda \cdot \sin(\alpha) \\ y = \rho \cdot \sin(\alpha) + \lambda \cdot \cos(\alpha) \end{cases} \quad (4.19)$$

onde, o parâmetro ( $\rho$ ) representa o comprimento do vetor que passa pela origem e é perpendicular à reta e ( $\alpha$ ) é o ângulo que esse vetor forma com o eixo X (ver figura 4.2).

Expandindo a equação (4.18) e usando a equação (4.19), nota-se que:

$$x^I = \frac{K_x \cdot \lambda + C_x}{K_t \cdot C_t} \quad e \quad y^I = \frac{K_y \cdot \lambda + C_y}{K_t \cdot C_t} \quad (4.20)$$

onde,

- $K_x = a_{12} \cdot \cos(\alpha) - a_{11} \cdot \sin(\alpha)$  e  $C_x = a_{11} \cdot \rho \cdot \cos(\alpha) + a_{12} \cdot \rho \cdot \sin(\alpha) + a_{13}$
- $K_y = a_{22} \cdot \cos(\alpha) - a_{21} \cdot \sin(\alpha)$  e  $C_y = a_{21} \cdot \rho \cdot \cos(\alpha) + a_{22} \cdot \rho \cdot \sin(\alpha) + a_{23}$
- $K_t = a_{32} \cdot \cos(\alpha) - a_{31} \cdot \sin(\alpha)$  e  $C_t = a_{31} \cdot \rho \cdot \cos(\alpha) + a_{32} \cdot \rho \cdot \sin(\alpha) + a_{33}$

Faz-se uma série de mudanças de variáveis, manipulação de equações e chega-se no Sistema (4.21) que apresenta o mapeamento direto entre  $\rho^I$  e  $\alpha^I$  de um plano, em  $\rho^{II}$  e  $\alpha^{II}$  de outro plano, utilizando os componentes da matriz de homografia.

$$\begin{cases} \alpha^I = \tan^{-1}\left(\frac{K_x \cdot C_t - K_t \cdot C_x}{K_t \cdot C_y - K_y \cdot C_t}\right) \\ \rho^I = \frac{K_x \cdot C_y - K_y \cdot C_x}{\sqrt{(K_x \cdot C_t - K_t \cdot C_x)^2 + (K_t \cdot C_y - K_y \cdot C_t)^2}} \end{cases} \quad (4.21)$$

onde,

- $K_x \cdot C_t - K_t \cdot C_x = \rho^{II} \cdot C_{23} - \sin(\alpha^{II}) \cdot C_{22} - \cos(\alpha^{II}) \cdot C_{21}$
- $K_t \cdot C_y - K_y \cdot C_t = \rho^{II} \cdot C_{13} - \sin(\alpha^{II}) \cdot C_{12} - \cos(\alpha^{II}) \cdot C_{11}$
- $K_x \cdot C_y - K_y \cdot C_x = -\rho^{II} \cdot C_{33} + \sin(\alpha^{II}) \cdot C_{32} + \cos(\alpha^{II}) \cdot C_{31}$

Sabendo que a câmera encontra-se acoplada à estrutura do robô, faz-se necessário expressar os parâmetros da reta em função da pose do robô. Para tal, definiu-se um sistema de coordenadas fixo ( $F$ ) e sistema de coordenadas móvel ( $M$ ), anexado ao robô, ambos ilustrados na Figura 4.4.

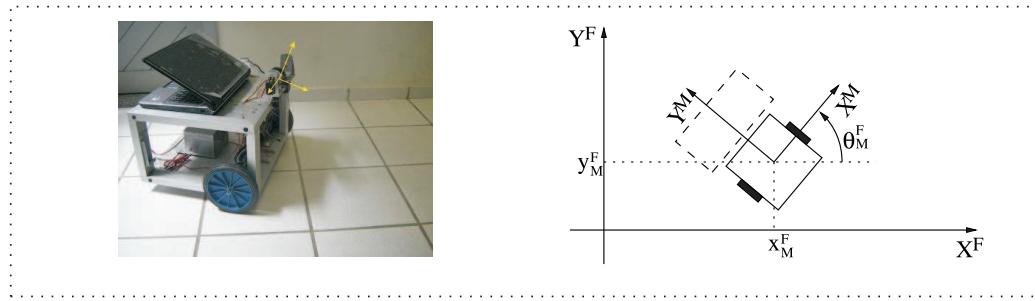


Figura 4.4: Sistemas de coordenadas Fixo ( $F$ ) e Móvel ( $M$ ).

A origem do sistema móvel tem coordenadas  $(x_M^F, y_M^F)$  no sistema fixo.  $\theta_M^F$  representa a rotação do sistema móvel com respeito ao sistema fixo. Deve-se notar que existe uma relação estreita entre essas variáveis  $(X_M^F, Y_M^F, \theta_M^F)$  e a pose do robô  $(x_t, y_t, \theta_t)$ , que é dada pela equação (4.22).

$$x_t = x_M^F \quad y_t = y_M^F \quad \theta_t = \theta_M^F + \pi/2 \quad (4.22)$$

Usou-se a relação entre as coordenadas dos sistemas ( $M$ ) e ( $F$ ) (sistema 4.23) e a equação (4.5) em ambos os sistemas de coordenada (equações 4.24 e 4.25).

$$\begin{cases} x^F = \cos(\theta_M^F)x^M - \sin(\theta_M^F)y^M + x_M^F \\ y^F = \sin(\theta_M^F)x^M + \cos(\theta_M^F)y^M + y_M^F \end{cases} \quad (4.23)$$

$$\rho^F = x^F \cos(\alpha^F) + y^F \sin(\alpha^F) \quad (4.24)$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \quad (4.25)$$

Substituindo o sistema (4.23) na equação (4.24), fazendo as equivalências com a equação (4.25) e substituindo algumas variáveis usando as equações (4.22), obtém-se os sistemas (4.26) e (4.27).

$$\begin{cases} \rho^M = \rho^F - x_t \cos(\alpha^F) - y_t \sin(\alpha^F) \\ \alpha^M = \alpha^F - \theta_t + \pi/2 \end{cases} \quad (4.26)$$

$$\begin{cases} \rho^M = -\rho^F + x_t \cos(\alpha^F) + y_t \sin(\alpha^F) \\ \alpha^M = \alpha^F - \theta_t - \pi/2 \end{cases} \quad (4.27)$$

Uma vez encontrada a relação entre  $\rho$  e  $\alpha$  e a pose do robô, o passo seguinte é substituir o sistema (4.26) ou o sistema (4.27) nas equações do sistema (4.21) a fim de mapear os parâmetros da reta diretamente da imagem para o mundo (sistema 4.28).

$$\begin{cases} \alpha^I = \tan^{-1}\left(\frac{K_x^* \cdot C_t^* - K_t^* \cdot C_x^*}{K_t^* \cdot C_y^* - K_y^* \cdot C_t^*}\right) \\ \rho^I = \frac{K_x^* \cdot C_y^* - K_y^* \cdot C_x^*}{\sqrt{(K_x^* \cdot C_t^* - K_t^* \cdot C_x^*)^2 + (K_t^* \cdot C_y^* - K_y^* \cdot C_t^*)^2}} \end{cases} \quad (4.28)$$

onde,

- $K_x^* \cdot C_t^* - K_t^* \cdot C_x^* = \rho^M \cdot C_{23} - \sin(\alpha^M) \cdot C_{22} - \cos(\alpha^M) \cdot C_{21}$
- $K_t^* \cdot C_y^* - K_y^* \cdot C_t^* = \rho^M \cdot C_{13} - \sin(\alpha^M) \cdot C_{12} - \cos(\alpha^M) \cdot C_{11}$
- $K_x^* \cdot C_y^* - K_y^* \cdot C_x^* = -\rho^M \cdot C_{33} + \sin(\alpha^M) \cdot C_{32} + \cos(\alpha^M) \cdot C_{31}$

Para decidir qual modelo usar, calculam-se ambos os valores de  $\alpha$  a partir do sistema (4.21) usando os sistemas (4.26) e (4.27) e usa-se o modelo que gerar o valor mais próximo ao valor medido de  $\alpha$ .

Usando o sistema (4.28) para obter o modelo do sensor  $h(\cdot)$ , é possível calcular, por derivação deste modelo, a matriz  $\mathbf{H}$  usada na fase de atualização do filtro de Kalman. Os elementos de  $\mathbf{H}$  juntamente com uma demonstração mais detalhada da formulação apresentada são descritos no Apêndice A.

### Correspondência ou Matching

Um aspecto crucial do algoritmo de SLAM é estabelecer uma correspondência entre a linha detectada na imagem e um dos marcos representado no vetor de estado. Para escolher o marco correto, primeiramente calculam-se os valores preditos de  $(\rho^F, \alpha^F)$  usando os valores medidos de  $(\tilde{\rho}^M, \tilde{\alpha}^M)$ . Usa-se o modelo do sistema (4.28) e a equação (4.26), se  $\tilde{\alpha} > 0$ , ou o sistema (4.28) e a equação (4.27), se  $\tilde{\alpha} < 0$ . Usamos  $(\sim)$  sobre a variável para indicar os valores medidos ao invés de calculados.

Os valores preditos são comparados com cada um dos valores  $(i\rho^F, i\alpha^F)$  no vetor de estado. Se a diferença entre o valor predito e o melhor  $(i\rho^F, i\alpha^F)$  é suficientemente pequena, a correspondência foi encontrada. Se não, considera-se que um novo marco foi detectada e ela é inserido no mapa, ou seja, o vetor de estados é aumentado.

## 4.5 Processamento de Imagem

### Detecção de Linhas

A técnica adotada neste trabalho para identificar os marcos, linhas pré-existentes no chão do ambiente, foi a Transformada de Hough. Esta técnica destina-se a encontrar instâncias imperfeitas de objetos em imagens digitais por um processo de votação.

Forsyth & Ponce (2002) comentam que do ponto de vista prático é interessante usar algum algoritmo detector de bordas para suavizar a textura e aumentar o contraste. Assim, optamos por utilizar junto com Hough o detector de bordas Canny [Canny 1986].

As imagens são capturadas em tons de cinza, convertidas para preto e branco usando detector Canny e só após esta etapa de segmentação é aplicada a Transformada de Hough. Ocorre que o uso deste detector de bordas é muito sensível e durante a experimentação observou-se que em imagens onde aparentemente deveriam ser detectadas algumas linhas o retorno do processamento sempre fornecia um número maior.

Constatou-se que este fato se dá devido ao detector Canny encontrar para cada linha na imagem suas duas fronteiras (bordas) e, então, quando da aplicação da Transformada de Hough (TH) o resultado para cada linha era na verdade as duas retas da sua fronteira (ver figuras 4.5 e 4.6).

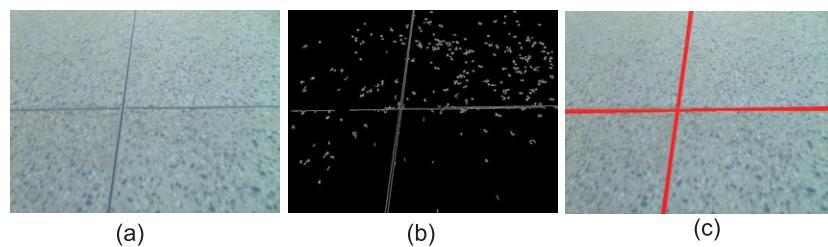


Figura 4.5: Imagem do chão de um ambiente interno, (a) Imagem Original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

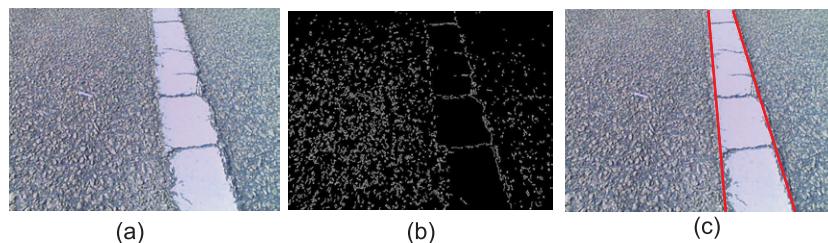


Figura 4.6: Imagem do chão de um ambiente externo: (a) Imagem Original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

No tocante aos SLAM, é fácil perceber que para ambientes internos o problema apresentado tem implicações maiores pois, além de aumentar em duas vezes o tamanho do vetor de estados o problema da associação de dados também fica mais crítico.

Assim, para superar este problema, implementou-se uma estratégia simples para filtrar a quantidade de linhas da imagem. A idéia é comparar todos os valores de  $\rho$  e  $\alpha$  obtidos na imagem e quando a diferença entre as estas medidas for menor que um determinado limiar assume-se que a média desses valores é a representação correta da linha (ver figura 4.7).

Outro problema enfrentado foi o de selecionar na imagem apenas as linhas existente no plano do chão. A solução encontrada foi utilizar como “padrão de chão” uma área imediatamente à frente do robô e esta informação é usada para segmentar a imagem. Detalharemos o algoritmo usado na segmentação no Capítulo 6. A figura 4.8 mostra o problema descrito e também como ele foi solucionado.

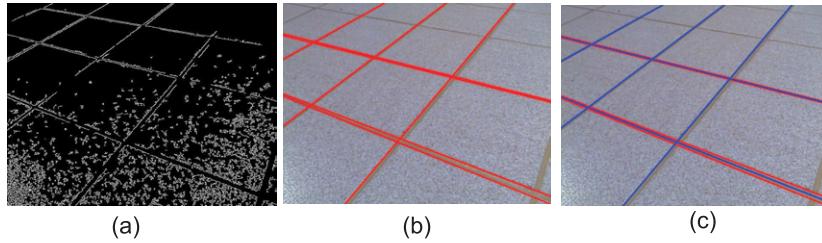


Figura 4.7: Filtragem de linhas.

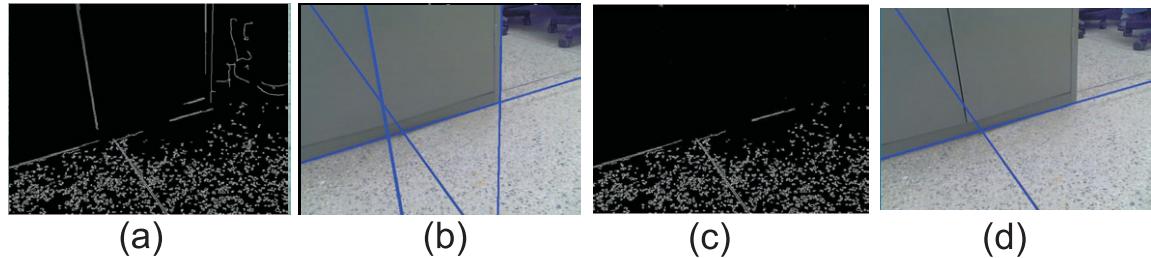


Figura 4.8: Imagem um ambiente interno: (a) Imagem processada usando detector de bordas Canny, (b) Imagem processada usando TH, (c) Imagem processada detector de bordas Canny e segmentação de chão, (d) Nova imagem processada usando TH.

### Conversão para o mundo

Assumindo que o chão é plano e que a câmera é fixa, então, existe uma relação constante (uma homografia  $\mathbf{A}$ ) entre os pontos no plano do chão  $(x, y)$  e os pontos no plano da imagem  $(u, v)$ :

$$s \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (4.29)$$

O fator de escala  $s$  é determinado para cada ponto de tal forma que o valor do terceiro elemento do vetor seja sempre 1. A homografia pode ser calculada *off-line* usando um padrão contendo quatro ou mais pontos notáveis com coordenadas conhecidas. Depois de detectar os pontos notáveis na imagem, têm-se diversas correspondências entre coordenadas de pontos no sistema de coordenadas móvel  $M$  e na imagem. Substituindo-se esses pontos na equação (4.29), obtém-se um sistema linear no qual se pode determinar os oito elementos da matriz de homografia  $\mathbf{A}$ .

No nosso caso, conforme apresentado no início deste Capítulo, a matriz  $\mathbf{A}$  é calculada uma única vez no início da experiência. Faz-se o robô identificar quatro ou mais pontos em uma imagem e após esta identificação os valores métricos dos pontos no mundo são passados para o robô através de um terminal de entrada.

Na prática, uma solução simples para calcular a matriz de homografia consiste em posicionar um padrão em frente à câmera e executar uma técnica de processamento de imagem de modo a obter automaticamente os pontos notáveis deste padrão que possuem coordenadas globais conhecidas. A figura (4.9) ilustra o comentado.

A identificação automática foi feita utilizando o detector Harris e o valor da homografia calculada é expresso na equação (4.30).

$$\mathbf{A} = \begin{pmatrix} 0.1389 & 0.0009 & -46.8035 \\ 0.0073 & -0.0711 & 98.6323 \\ 0.0001 & 0.0029 & 1 \end{pmatrix} \quad (4.30)$$

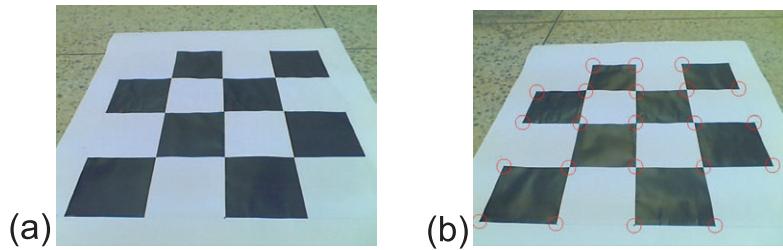


Figura 4.9: Cálculo da Homografia: (a) Padrão, (b) Detecção automática dos pontos.

## 4.6 Resumo do Capítulo

Este Capítulo objetivou apresentar o sistema de SLAM visual utilizado neste trabalho. Iniciou-se mostrando os aspectos gerais do sistema que tem como objetivo final a construção de um mapa híbrido com as representações em grade de ocupação e mapa de características. Comentou-se que neste Capítulo seria apresentado apenas a parte do SLAM visual e que a construção da grade será detalhada em Capítulos posteriores.

Foi mencionado que as bases do nosso sistema de SLAM são o Filtro de Kalman Estendido, a Transformada de Hough e o uso de uma matriz de Homografia. Apresentou-se a fundamentação teórica das principais técnicas envolvidas nesta solução e também foram mostrados alguns trabalhos relacionados a esta proposta. A parte final ficou com a modelagem matemática e também com comentários sobre alguns problemas intermediários que surgiram. O Capítulo seguinte trará os resultados práticos do SLAM visual proposto.



---

# Capítulo 5

## Resultados SLAM Visual

---

*Este Capítulo apresentará os resultados do SLAM visual que foi descrito matematicamente no Capítulo anterior. O sistema é baseado em visão monocular e usa como marcos do SLAM as linhas do chão que são extraídas das imagens pela Transformada de Hough. Os experimentos foram realizados utilizando um robô cujas rodas são movidas por motores DC com acionamento diferencial. Cada motor tem um encoder óptico e uma placa dedicada baseada em microcontrolador PIC que controla a velocidade local. Além disso, o robô possui, acoplado à sua estrutura, uma webcam colorida e um notebook cuja função é o processamento das informações. Para mostrar os resultados usou-se a idéia de cenários a fim de facilitar a discussão sobre cada experiência realizada.*

### 5.1 Cenário A: Linhas Paralelas

O local por onde o robô navegou durante esta experimentação foi o Laboratório de Automação em Petróleo - LAUT da Universidade Federal do Rio Grande do Norte - UFRN. Este prédio pode ser visto no sistema de mapeamento online *Google Earth* utilizando as coordenadas  $5^{\circ}84'31"S$   $35^{\circ}19'72"W$ . A figura 5.1 apresenta imagens deste prédio.



Figura 5.1: LAUT-UFRN: (a) Fachada do prédio, (b) Imagem interna do corredor.

Este prédio foi escolhido para a realização desta experiência por possuir linhas paralelas formadas pela junção entre as placas que compõem o piso. A idéia deste cenário foi verificar se o sistema proposto era adequado para este tipo de situação.

A figura 5.2 mostra a disposição das linhas no ambiente, a trajetória que foi comandada ao robô, o campo de visão do robô e o rótulo de cada linha. O rótulo das linhas é relacionado com a ordem temporal que cada uma é identificada pelo robô.

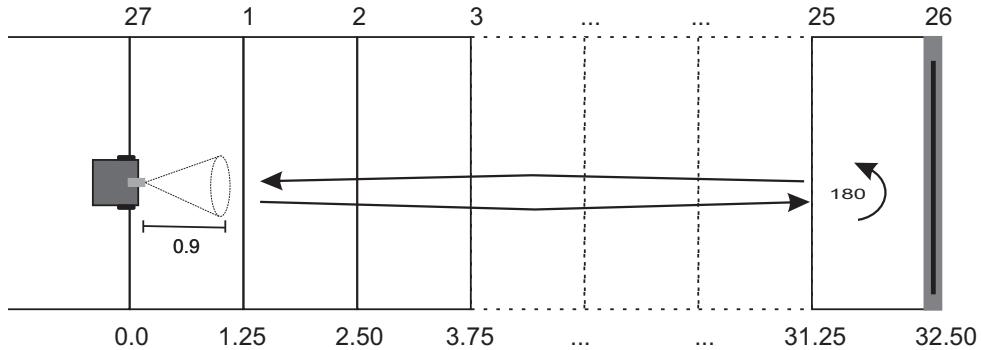


Figura 5.2: Cenário A: Descrição do ambiente.

No início do experimento foi posicionado à frente do robô um padrão para que a matriz de homografia pudesse ser calculada. O padrão foi posicionado de forma a preencher quase a totalidade do campo de visão do robô. A figura 5.3 ilustra o descrito acima e a equação (5.1) mostra o valor da matriz homográfica calculada.

$$\mathbf{A} = \begin{pmatrix} 0.1417 & 0.0009 & -49.2065 \\ 0.0073 & -0.0761 & 98.6323 \\ 0.0001 & 0.0029 & 1 \end{pmatrix} \quad (5.1)$$

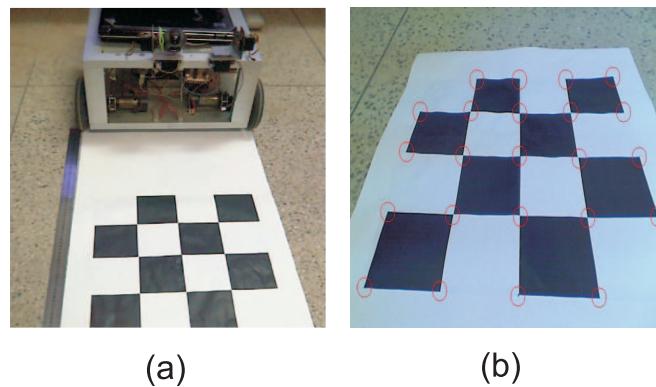


Figura 5.3: Cálculo da Homografia Cenário A: (a) Padrão, (b) Padrão processado.

Durante seu trajeto o robô processou 1712 imagens e o número médio de observação de cada reta foi de 23 vezes. O processamento de imagem foi satisfatório e falhou poucas vezes. A falha sempre se deu pela não identificação de uma reta na imagem porém, marcos espúrios não foram detectados. A figura 5.4 mostra uma imagem típica do ambiente por onde o robô navegou e a figura 5.5 mostra um caso de falha do processamento de imagem.

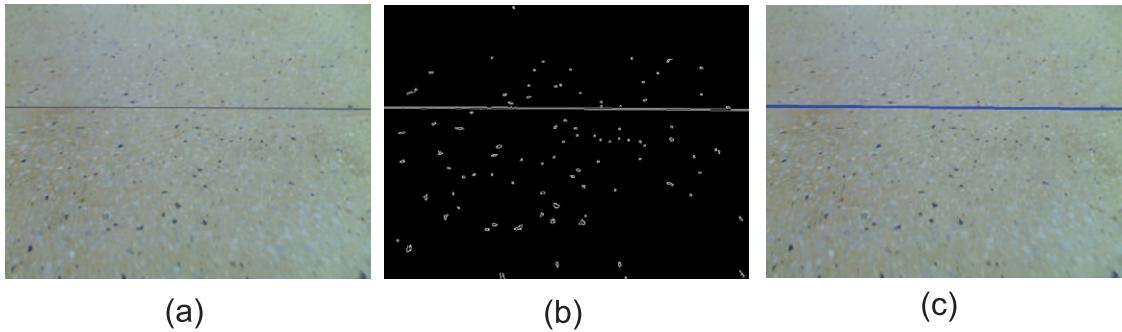


Figura 5.4: Imagem do piso LAUT: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

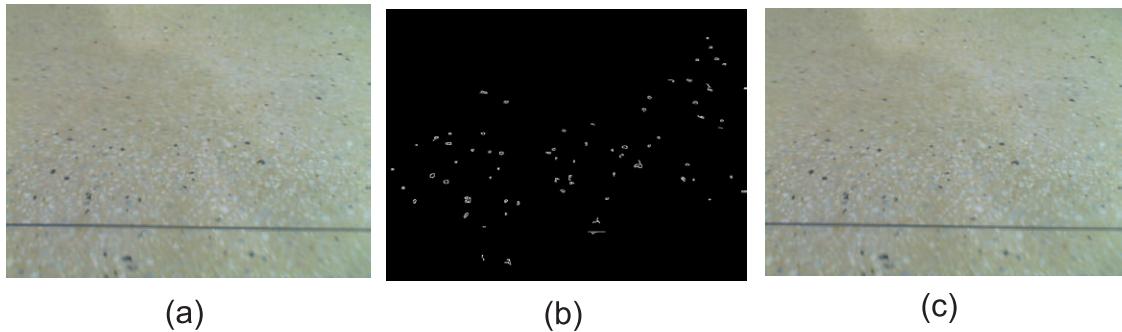


Figura 5.5: Falha no processamento de imagem: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

A sequencia de identificação das linhas pode ser vista na figura 5.6. O processo de correspondência funcionou corretamente em 93% dos casos e falhou no momento em que o robô executava o movimento de giro de 180° graus sobre seu próprio eixo.

O erro de correspondência pode ser visto na figura 5.6(b) uma vez que a linha 26 deveria ter sido detectada e na verdade o sistema considerou que a linha detectada foi a 25. Ainda nesta mesma figura, na parte (c), pode ser observado que a linha inicial (linha 27) sobre a qual o robô foi posicionado no início da experiência e não foi detectada neste momento só foi identificada no final da experiência.

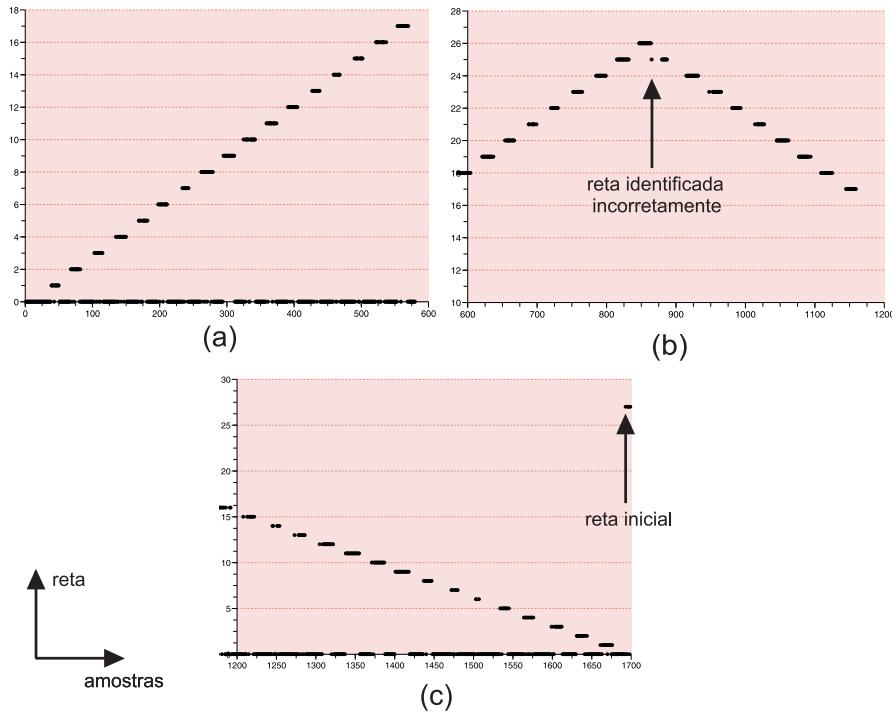
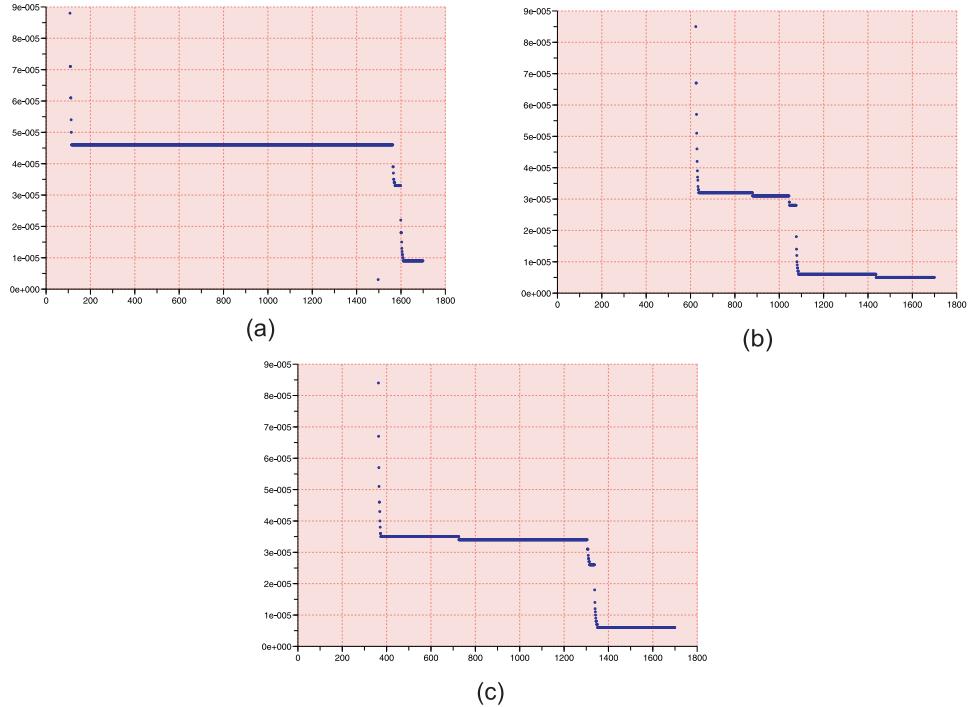
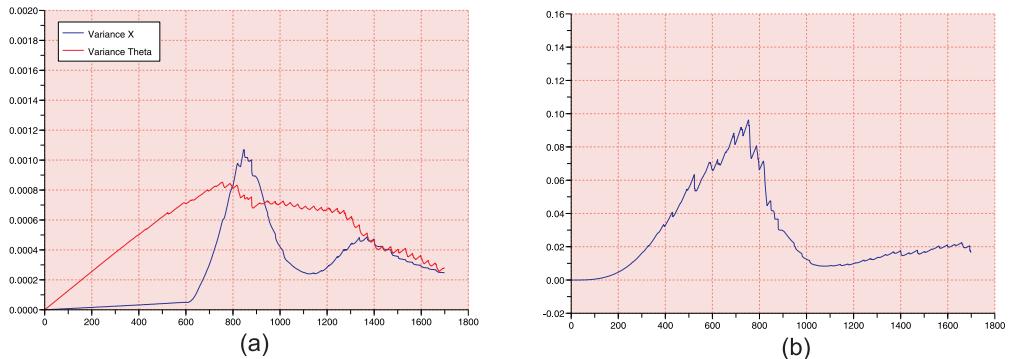


Figura 5.6: Sequencia de identificação das retas.

Para analisar o comportamento da variância foram escolhidas três retas posicionadas em espaços diferentes do ambiente. São elas: a reta 3 que foi observada no passo 103, e re-observada no passo 1599; a reta 11 que foi observada no passo 359, e re-observada no passo 1338, e a linha 19 que foi observada no passo 621, e re-observada no passo 1077. A figura 5.7 mostra a variância em  $\rho$  de cada uma da retas mencionadas. Nessa figura é possível verificar facilmente o instante da atualização mais brusca desta variância.

Ainda sobre variância, foi detectado que a magnitude das variâncias de  $x_R$  e  $\theta_R$  da pose do robô foi menor que a da coordenada  $y_R$ . Isso se deu porque no caminho percorrido pelo robô as retas são vistas de forma que a correção da coordenada  $y_R$  não pode ser realizada de forma eficiente, pois uma linha reta à frente do robô não fornece informação suficiente para a correção das três coordenadas de sua pose.

A figura 5.8(a) mostra as variâncias de  $x_R$  e  $\theta_R$  no decorrer deste experimento e a figura 5.8(b) mostra a variância da coordenada  $y_R$ . Observando esta segunda figura, próximo do passo 750, pode-se perceber que há uma significativa atualização da variância de  $y_R$ . Isto acontece porque é o instante em que o robô inicia o movimento de giro e uma reta que era vista de forma horizontal na imagem passou a ser vista de forma inclinada e desta forma ela contribui para a correção das três coordenadas do robô.

Figura 5.7: Comportamento da variância em  $\rho$ : (a) Linha 3, (b) Linha 11, (c) Linha 19.Figura 5.8: Comportamento da variância: (a) Coordenadas  $x$  e  $\theta$  da pose do robô, (b) Coordenada  $y$  da pose do robô.

Por fim, a Tabela 5.1 mostra os valores dos marcos calculados por SLAM depois de 1700 iterações. Os valores são expressos no sistema de coordenadas global assumindo que a pose inicial do robô é  $(0, 0, 0^\circ)$ . Os valores calculados indicaram um erro médio de  $0,043m$  para a característica  $\rho$  das retas e de  $2,33^\circ$  para a característica  $\alpha$ .

Tabela 5.1: Resultados experimentais: Cenário A.

Linha	Medido		Calculado		Linha	Medido		Calculado	
[.]	$\rho(m)$	$\alpha(^{\circ})$	$\rho(m)$	$\alpha(^{\circ})$	[.]	$\rho(m)$	$\alpha(^{\circ})$	$\rho(m)$	$\alpha(^{\circ})$
reta [1]	01.25	0	01.25	3.38	reta [15]	18.75	0	18.78	0.41
reta [2]	02.50	0	02.51	3.13	reta [16]	20.00	0	20.07	0.62
reta [3]	03.75	0	03.76	3.25	reta [17]	21.25	0	21.26	0.45
reta [4]	05.00	0	05.02	3.38	reta [18]	22.50	0	22.53	0.76
reta [5]	06.25	0	06.26	2.61	reta [19]	23.75	0	23.79	1.13
reta [6]	07.50	0	07.48	3.09	reta [20]	25.00	0	25.02	0.97
reta [7]	08.75	0	08.77	3.7	reta [21]	26.25	0	26.29	1.12
reta [8]	10.00	0	10.04	3.04	reta [22]	27.50	0	27.61	2.09
reta [9]	11.25	0	11.29	3.06	reta [23]	28.75	0	28.81	0.90
reta [10]	12.50	0	12.52	1.87	reta [24]	30.00	0	30.07	3.72
reta [11]	13.75	0	13.78	1.26	reta [25]	31.25	0	31.28	5.31
reta [12]	15.00	0	15.07	0.94	reta [26]	32.50	0	32.41	3.26
reta [13]	16.25	0	16.34	1.19	reta [27]	00.00	0	00.05	2.14
reta [14]	17.50	0	17.56	0.93					

## 5.2 Cenário B: Linhas Concorrentes

O segundo experimento foi realizado no Departamento de Engenharia de Computação e Automação - DCA da Universidade Federal do Rio Grande do Norte - UFRN. Este prédio pode ser visto no sistema de mapeamento online *Google Earth* utilizando as coordenadas  $5^{\circ}50'34''S$   $35^{\circ}11'51''W$ . A figura 5.9 apresenta imagens deste prédio.



Figura 5.9: DCA-UFRN: (a) Fachada do prédio, (b) Imagem interna do corredor.

A escolha do DCA-UFRN foi determinada devido o piso do ambiente possuir linhas concorrentes. A idéia deste segundo cenário era avaliar se com a existência de linhas concorrentes as variâncias da pose do robô ao final do trajeto teriam a mesma magnitude.

A figura 5.10 mostra a disposição das linhas no ambiente, a trajetória que foi comandada ao robô, o campo de visão do robô e o rótulo de cada linha. O rótulo das linhas é relacionada com a ordem temporal que cada uma é identificada pelo robô.

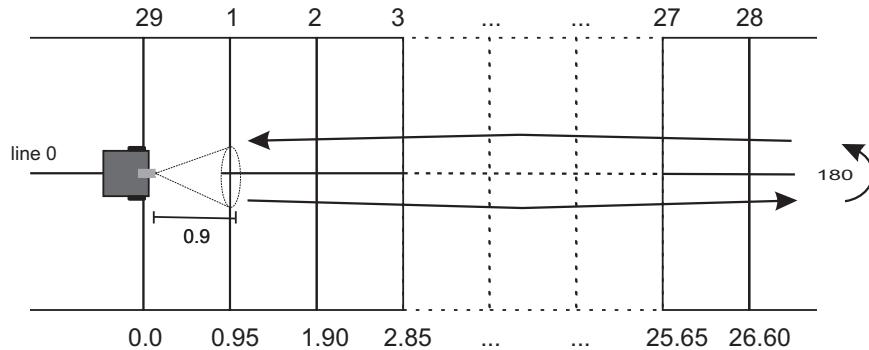


Figura 5.10: Cenário B: Descrição do ambiente.

Assim como no experimento descrito no Cenário A, no início deste teste também foi posicionado a frente do robô um padrão para que a matriz de homografia pudesse ser calculada. Porém, diferentemente da experiência anterior, os pontos do padrão foram identificados com a própria técnica de processamento de imagens adotada para detectar as linhas, ou seja, a Transformada de Hough. O propósito desta mudança dar-se em acreditar que se a homografia for calculada com base nos pontos identificados por Hough, o erro do processamento de imagem já será embutido na própria matriz. A figura 5.11 ilustra o comentado e a equação (5.2) mostra o valor da matriz homográfica calculada.

$$\mathbf{A} = \begin{pmatrix} 0.1378 & 0.0007 & -48.2710 \\ 0.0073 & -0.0691 & 98.8632 \\ 0.0002 & 0.0019 & 1 \end{pmatrix} \quad (5.2)$$

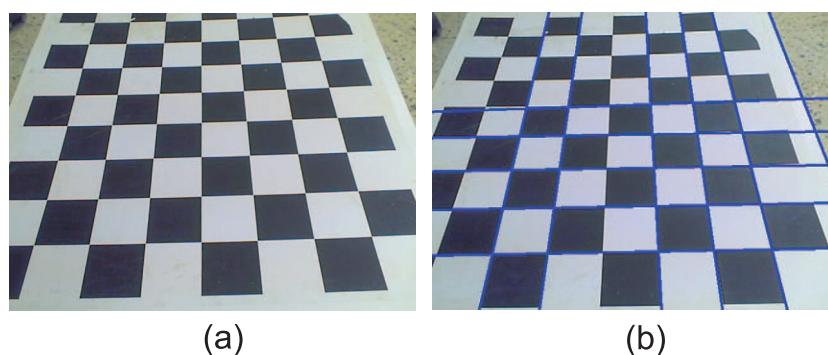


Figura 5.11: Cálculo da Homografia Cenário B: (a) Padrão, (b) Padrão processado.

Nesta experiência o robô navegou por aproximadamente 60m e capturou 2214 imagens. O processo de correspondência respondeu corretamente em 95% dos casos e, considerando o movimento total do robô, foi verificado que em 87% das amostras o robô identificou retas sendo que uma reta foi detectada em 61% das imagens e duas retas foram identificadas em 26% das imagens.

O processamento de imagem foi satisfatório e também falhou poucas vezes. A figura 5.12 mostra uma imagem típica do ambiente por onde o robô navegou e a figura 5.13 mostra uma situação com sombra e mudança de iluminação.

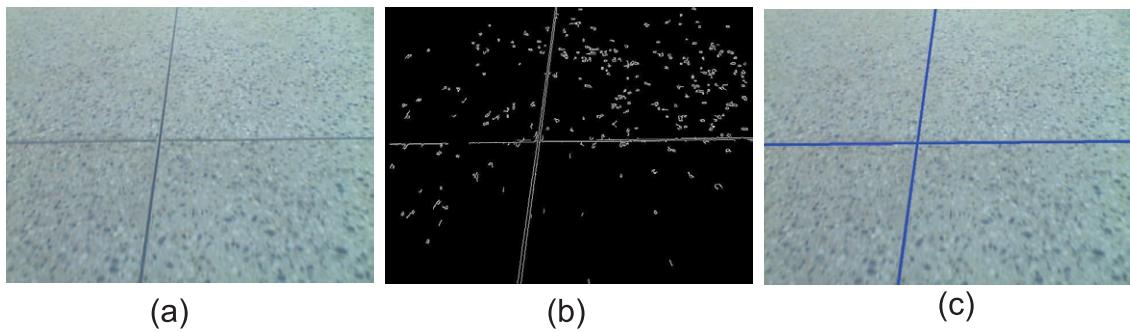


Figura 5.12: Imagem do piso DCA: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

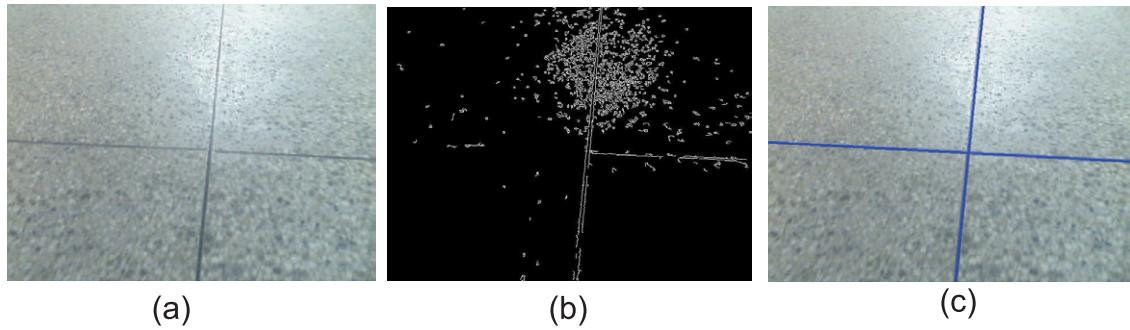


Figura 5.13: Influência da iluminação: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

As variâncias da pose do robô são exibidas na figura 5.14. Diferentemente do experimento do Cenário A, as variâncias de  $x_R$ ,  $y_R$  e  $\theta_R$  desta vez tiveram a mesma magnitude. Isto ocorre porque no trajeto do robô foram identificadas retas concorrentes e as informações existentes são suficientes para atualizar as três coordenadas da sua pose.

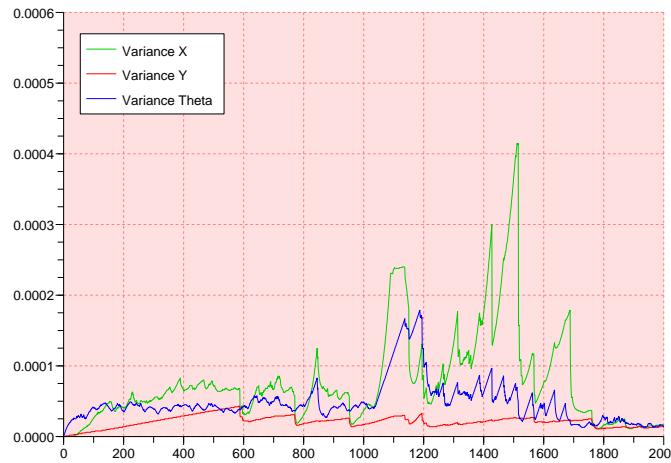


Figura 5.14: Comportamento das variâncias da pose do robô.

A Tabela 5.2 mostra os valores calculados dos marcos depois de 2214 iterações. Os valores são expressos no sistema de coordenadas global assumindo que a pose inicial do robô é  $(0, 0, 0^\circ)$  e os valores calculados indicaram um erro médio de  $0,029m$  para a característica  $\rho$  das retas e de  $2,13^\circ$  para a característica  $\alpha$ .

Tabela 5.2: Resultados experimentais: Cenário B.

Linha	Medido		Calculado		Linha	Medido		Calculado	
[.]	$\rho(m)$	$\alpha(^{\circ})$	$\rho(m)$	$\alpha(^{\circ})$	[.]	$\rho(m)$	$\alpha(^{\circ})$	$\rho(m)$	$\alpha(^{\circ})$
reta [0]	00.00	90	0.01	89.25	reta [15]	14.25	0	14.24	1.05
reta [1]	00.95	0	00.92	3.21	reta [16]	15.20	0	15.24	5.06
reta [2]	01.90	0	01.94	3.83	reta [17]	16.15	0	16.19	5.10
reta [3]	02.85	0	02.86	0.23	reta [18]	17.10	0	17.12	3.05
reta [4]	03.80	0	03.81	1.64	reta [19]	18.05	0	18.06	3.71
reta [5]	04.75	0	04.72	2.85	reta [20]	19.00	0	19.03	2.49
reta [6]	05.70	0	05.68	1.53	reta [21]	19.95	0	19.92	1.26
reta [7]	06.65	0	06.64	0.47	reta [22]	20.90	0	20.81	2.45
reta [8]	07.60	0	07.61	0.35	reta [23]	21.85	0	21.90	2.57
reta [9]	08.55	0	08.58	1.29	reta [24]	22.80	0	22.76	5.21
reta [10]	09.50	0	09.53	0.49	reta [25]	23.75	0	23.72	2.46
reta [11]	10.45	0	10.49	1.52	reta [26]	24.70	0	24.73	3.74
reta [12]	11.40	0	11.46	1.49	reta [27]	25.65	0	25.67	5.02
reta [13]	12.35	0	12.31	2.43	reta [28]	26.60	0	26.62	6.00
reta [14]	13.30	0	13.31	1.54	reta [29]	00.00	0	00.03	0.44

### 5.3 Cenário C: Fechamento de laço

O terceiro experimento foi realizado no Departamento de Física Teórica e Experimental-DFTE da Universidade Federal do Rio Grande do Norte - UFRN. Este prédio pode ser visto no sistema de mapeamento online *Google Earth* utilizando as coordenadas  $5^{\circ}50'29''S$   $35^{\circ}11'49''W$ . A figura 5.15 mostra quatro vistas do interior deste prédio.

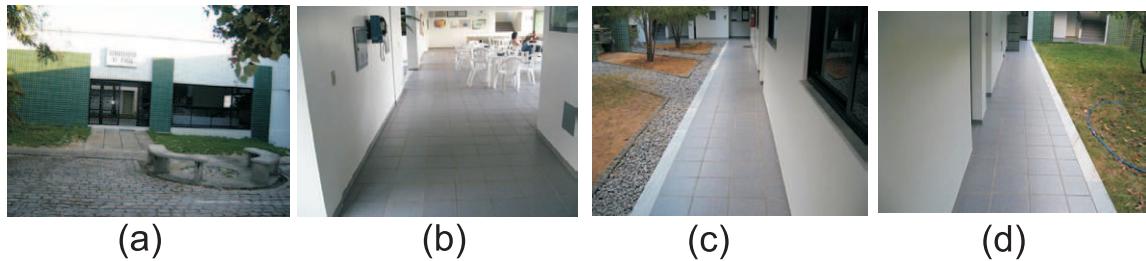


Figura 5.15: DF-UFRN: (a) Fachada do prédio, (b) Imagem interna do corredor vista 1, (c) Imagem interna do corredor vista 2, (d) Imagem interna do corredor vista 3.

O DFTE-UFRN foi escolhido para este terceiro experimento (Cenário C) por possuir uma área livre em formato quadrado permitindo testar a nossa proposta no problema de fechamento de laço. O ambiente possui piso formado por cerâmicas quadradas com  $25cm$  de lado e as interseções destas cerâmicas formam linhas horizontais e verticais que são identificadas no trajeto do robô. A figura 5.16 ilustra as características deste local.

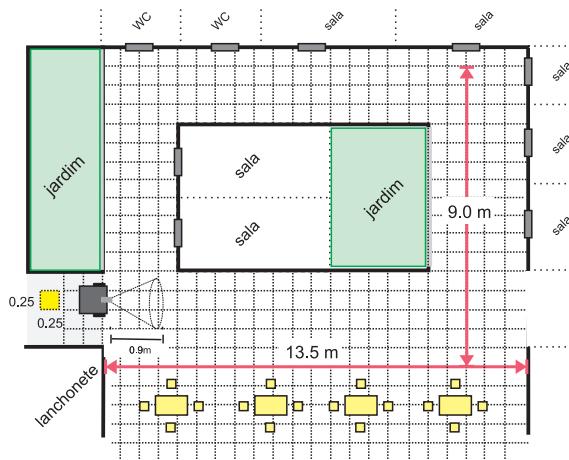


Figura 5.16: Cenário C: Descrição do ambiente.

O robô executou uma trajetória quase retangular de circuito fechado dentro deste edifício e durante seu movimento 1962 imagens foram processadas. A câmera utilizada captura imagens  $640 \times 480$  e cada imagem é processada, na média, em  $180ms$ .

A figura 5.17 mostra os gráficos do tempo de aquisição da imagem, tempo de processamento e o tempo total do sistema incluindo aquisição, processamento e cálculos do algoritmo de SLAM. Nesse gráfico é possível observar nitidamente dois picos. Estes picos se deram devido à mudança de iluminação existente no ambiente quando o robô se aproximou das áreas de jardim. O tempo médio de aquisição foi de  $50ms$ , o tempo médio de processamento foi de  $125ms$  e o tempo médio total do sistema foi de  $180ms$ .

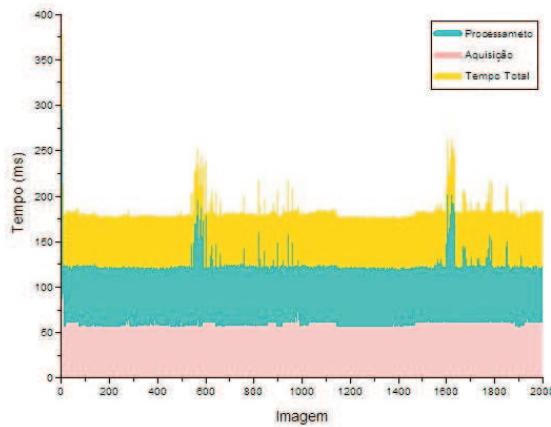


Figura 5.17: Tempos do Sistema de Visão.

No tocante à homografia, foram utilizados os próprios pontos do plano de movimento do robô como mostra a figura 5.18. A câmera foi posicionada de modo que foi possível ter um ângulo de visão da ordem de duas vezes o tamanho do robô. A equação (5.3) mostra a matriz de homografia utilizada.

$$\mathbf{A} = \begin{pmatrix} 0.1317 & 0.0008 & -47.3183 \\ 0.0073 & -0.0861 & 98.1458 \\ 0.0001 & 0.0017 & 1 \end{pmatrix} \quad (5.3)$$

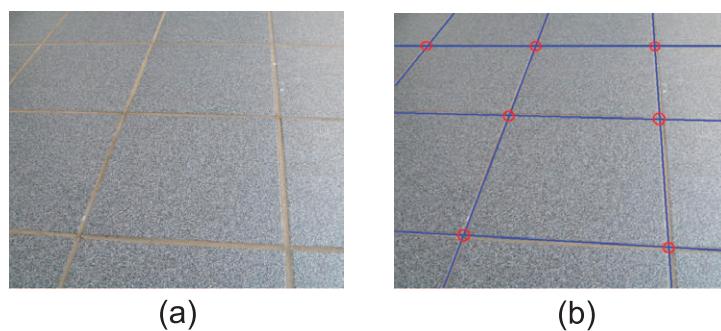


Figura 5.18: Cálculo da Homografia Cenário C: (a) Padrão, (b) Padrão processado.

A fim de verificar a corretude da homografia encontrada, calculou-se o erro de re-projeção utilizando os pontos detectados na imagem e seus correspondentes no mundo. O erro médio calculado foi de  $e = 1.9cm$ .

Após o movimento total realizado pelo robô (cerca de 45m), a correspondência de linhas detectadas foi correta em 95% dos casos e cada linha foi observada em 15 imagens consecutivas, em média. Em 98% das imagens o sistema detectou linhas: três linhas em 61%, quatro linhas em 26% e cinco linhas em 11% das imagens. Os 2% restantes foram erros do algoritmo de processamento de imagem.

Mesmo com 2% de erro, o sistema de visão se comportou bem no tocante a problemas de iluminação e também na presença de sujeira no chão. A figura 5.19 mostra uma imagem típica do ambiente a o aspecto da iluminação pode ser visto na figuras 5.20.

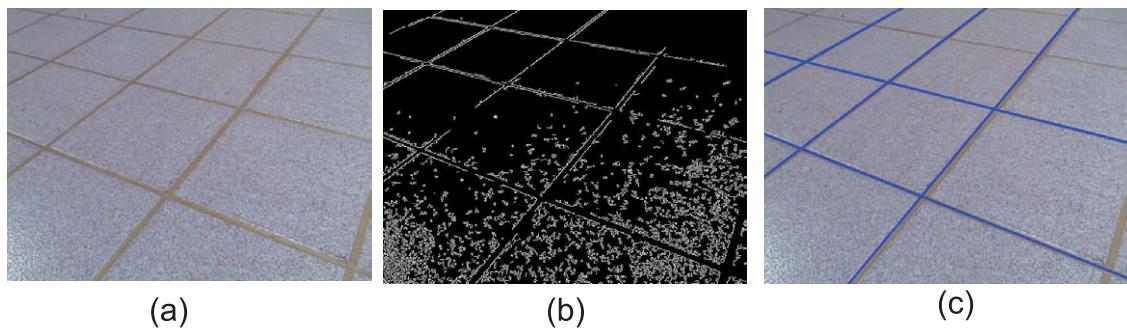


Figura 5.19: Imagem do piso DFTE: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

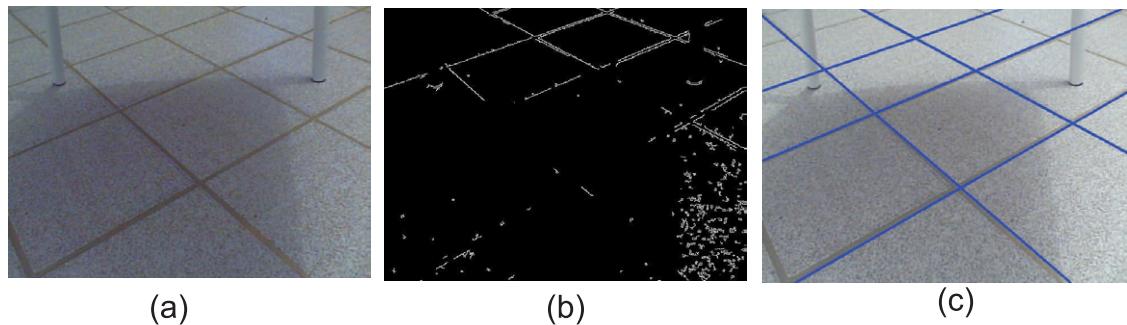
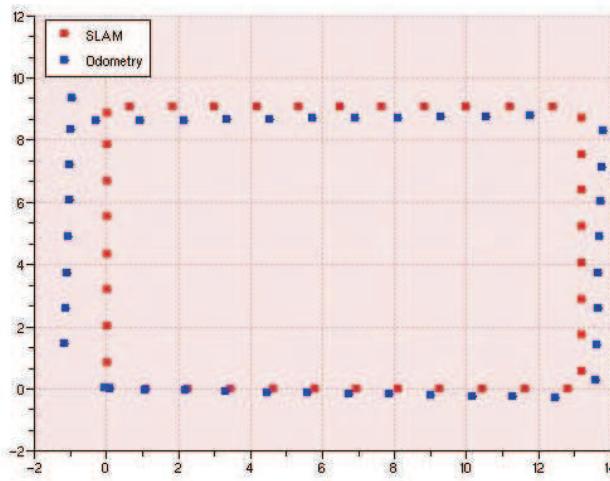


Figura 5.20: Influência da iluminação: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

A figura 5.21 mostra as trajetórias calculadas. Os pontos em azul correspondem à trajetória calculada apenas por odometria e os pontos em vermelho à trajetória calculada por SLAM. A distância entre a posição inicial e final, assumindo uma pose inicial  $(0, 0, 0^\circ)$ , foi de  $0.89m$ , usando somente odometria, e  $0.03m$ , usando SLAM. Vale ressaltar que a distância final real, medida *in loco*, foi muito próxima da calculada por SLAM.



A figura 5.23 mostra a variância ( $\sigma^2$ ) em  $\rho$  quando da primeira re-observação da linha que caracteriza o fechamento do circuito. Esta linha foi observada pela primeira vez no passo 23, e re-observada pela primeira vez no passo 1591. É possível verificar a atualização da variância apenas quando a linha é observada e re-observada. Antes da primeira re-observação (circuito fechado), o valor da variância de  $\rho$  era de  $4.6 \times 10^{-5}$  ( $\sigma = 0.68\text{cm}$ ) e após a re-observação este valor foi reduzido para  $9.2 \times 10^{-6}$  ( $\sigma = 0.30\text{cm}$ ).

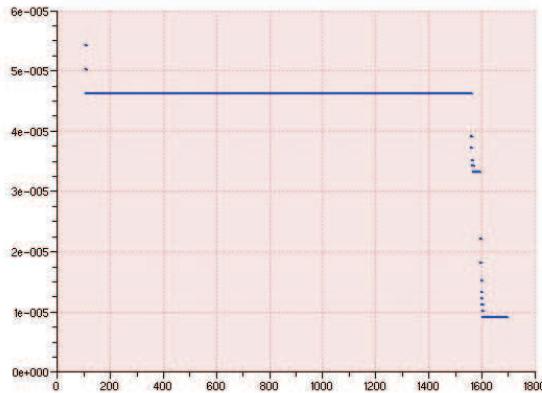


Figura 5.23: Variância em  $\rho$  da linha que caracteriza o fechamento de circuito.

## 5.4 Cenário D: Ambiente externo

A quarta experiência, Cenário D, foi realizada no estacionamento de um supermercado situado em Natal-RN-Brasil. Este ambiente possui linhas pintadas no chão: linhas brancas para delimitar a região de estacionamento e linhas amarelas para indicar a região central da pista por onde os automóveis se deslocam. A figura 5.24(a) mostra a imagem do supermercado que pode ser vista no *Google Earth* utilizando as coordenadas  $5^{\circ}45'31''S$   $35^{\circ}14'50''W$  e a figura 5.24(b) resume a descrição do ambiente.

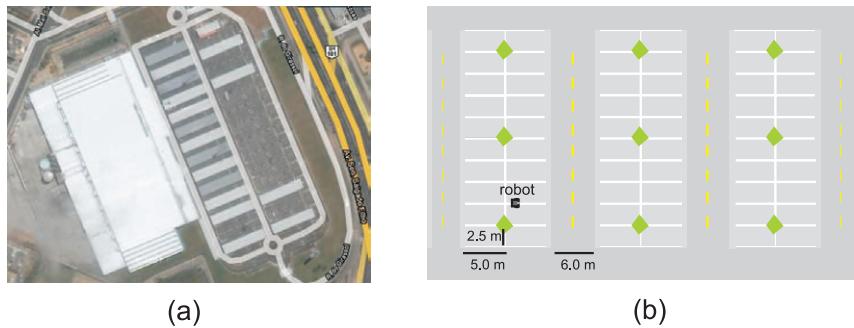


Figura 5.24: Supermercado Nata-RN: (a) Imagem satélite, (b) Descrição do ambiente.

Nesta experiência, assim como na anterior, também foi comandado ao robô a execução de uma trajetória retangular de circuito fechado. A tarefa consistiu de o robô se mover reto

para frente  $35m$ , girar sobre o próprio eixo  $90^\circ$ , mover-se reto para frente  $16m$ , girar sobre o próprio eixo  $90^\circ$ , mover-se novamente para frente  $35m$  e depois girar  $90^\circ$  sobre seu eixo e, por fim, mover-se  $16m$  para frente. A figura 5.25 apresenta a trajetória realizada pelo robô utilizando o algoritmo de SLAM visual proposto.

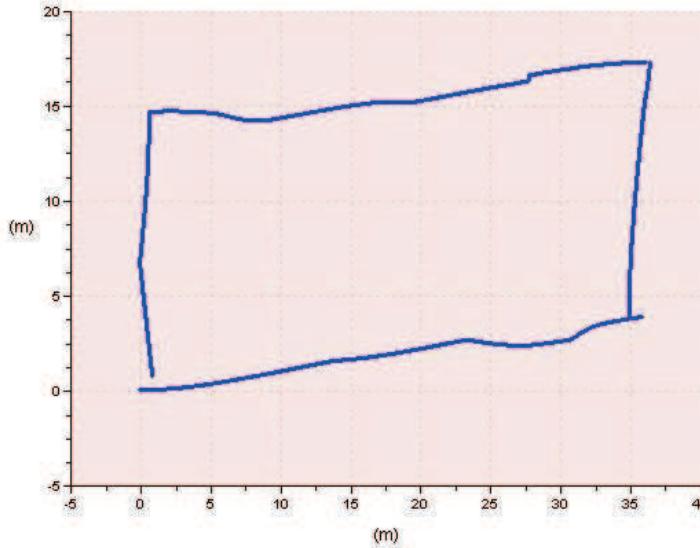


Figura 5.25: Trajetória percorrida pelo robô no Supermercado-Natal/RN.

Durante o movimento total do robô 2566 imagens foram processadas em um percurso aproximado de  $100m$ . O tempo médio de processamento de cada imagem foi de  $100ms$  e foi bem próximo do tempo médio do sistema de visão utilizado no experimento do Cenário B e menor do que o tempo de processamento do Cenário C. A redução dá-se pelo fato de menos retas serem processadas.

A câmera existente no robô também foi posicionada de modo que o ângulo de visão do robô fosse da ordem de duas vezes o seu tamanho e a matriz de homografia utilizada é apresentada na equação 5.4. Foi usado o mesmo padrão apresentado no Cenário B e é possível observar que a ordem de grandeza dos elementos é semelhante.

$$\mathbf{A} = \begin{pmatrix} 0.1118 & 0.0012 & -56.3155 \\ 0.0068 & -0.0651 & 102.1874 \\ 0.0001 & 0.0032 & 1 \end{pmatrix} \quad (5.4)$$

No percurso, retas foram identificadas em 78% das imagens sendo que: uma linha em 42%, duas linhas em 22%, três linhas em 9% das imagens e quatro linhas em 5%.

A figura 5.26 mostra uma imagem típica do ambiente externo por onde o robô navegou e a figura 5.27 apresenta uma situação em que o processador de imagem deveria ter identificado duas retas e o mesmo identificou apenas uma.

Sobre sujeira no chão, os resultados desta experiência não foram tão satisfatórios. Como exemplo tem-se a figura 5.28 que mostra uma marca que não foi identificada pelo robô. O sistema de visão também falhou por não identificar sempre todas as retas na imagem como mostra a figura 5.26.

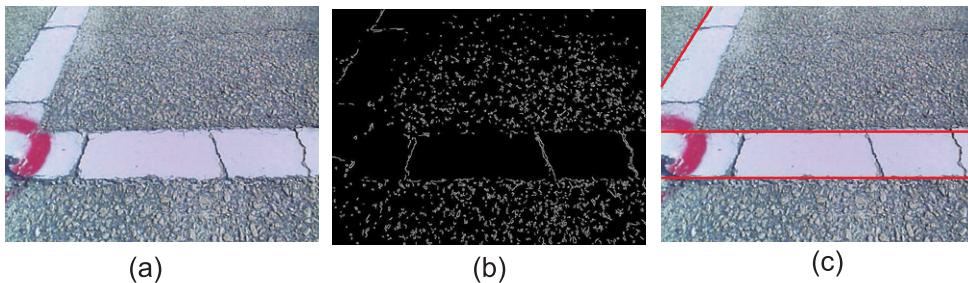


Figura 5.26: Imagem do piso do ambiente externo: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

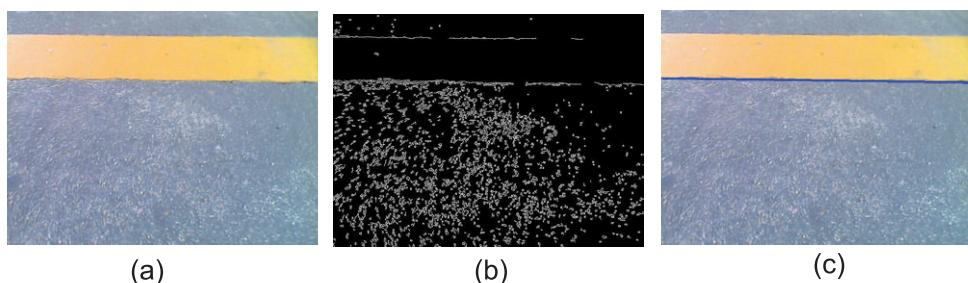


Figura 5.27: Falha no processamento de imagem: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

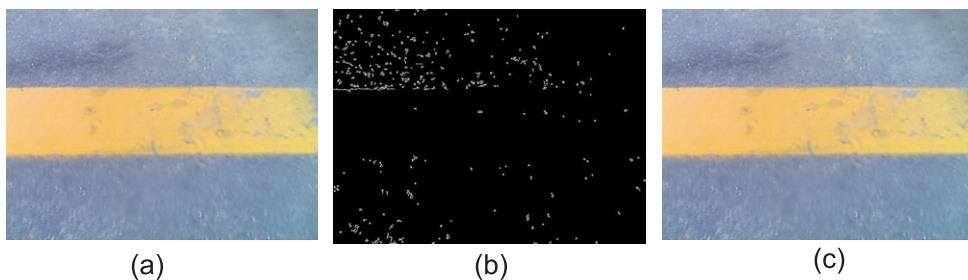


Figura 5.28: Falha no processamento de imagem: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

A correspondência de linhas detectadas foi correta em 94% dos casos e cada linha foi observada em 7 imagens consecutivas, na média. Mesmo como menos observação de cada reta, quando comparada à experiência indoor, a taxa de correspondência foi maior. Isto se deve ao fato de que as retas no ambiente outdoor são mais espaçadas.

No tocante ao fechamento do circuito, o algoritmo comportou-se adequadamente mesmo depois do robô ter navegado por mais de 100m. O erro entre a posição final do robô calculada por SLAM e sua posição inicial foi de 0.88m. Este valor foi medido no ambiente considerando a posição de partida do robô e a posição final.

## 5.5 Resultados Comparativos

A título de comparação foi usada a técnica clássica de SLAM visual baseada em pontos. Sabe-se que um grande problema relacionado a esta técnica é a inicialização correta das três coordenadas do marco detectado. Porém, existem trabalhos que propõem soluções para suprir este problema como o apresentado por Civera et al. (2008).

O algoritmo implementado para confrontar nossa proposta utiliza como características as coordenadas ( $x, y, z$ ) dos pontos de interseção entre as retas que formam o piso do ambiente. Para que a comparação fosse feita da forma mais justa possível a identificação das retas utilizadas para a extração dos pontos foi feita com a Transformada de Hough e no instante da detecção de um novo marco foi atribuído à coordenada relacionada à altura o seu valor correto. Somente a partir daí este valor passa a ser atualizado pelo algoritmo.

A figura 5.29 mostra as trajetórias calculadas pelo robô utilizando o algoritmo baseado em pontos (azul) e utilizando o algoritmo baseado em retas (vermelho). Percebe-se que a abordagem utilizando retas comportou-se de forma mais satisfatória que a abordagem utilizando pontos.

O desvio padrão da coordenada da altura do sistema baseado em pontos foi de  $\sigma = 0.02m$ . O erro final, medido *in-loco*, usando linhas, foi de  $0.03m$  e usando pontos foi de  $0.18m$ . O número médio de marcas observadas por imagem foi de 8 para o método de pontos e de 5 para o método de linhas. Porém, o número médio de observações de cada marca foi de 15 vezes para o método usando linhas e de 6 vezes para o usando pontos.

Outro resultado comparativo que merece ser destacado é a quantidade de características utilizadas no algoritmo de SLAM durante o movimento total do robô. Para a abordagem usando pontos foram inseridas no vetor de estados 1401 características referentes a 467 marcos enquanto que no algoritmo utilizando retas foram inseridas 388 características de 194 marcos. A Tabela 5.3 resume a comparação descrita.

Tabela 5.3: Resultados comparativos.

<i>Item Comparado</i>	<i>Abordagem com Pontos</i>	<i>Abordagem com Linhas</i>
Tempos de Processamento	180ms	180ms
Número médio de observações	8	5
Número de marcos	467	194
Número de características	1401	388
Erro final	0.18m	0.03m

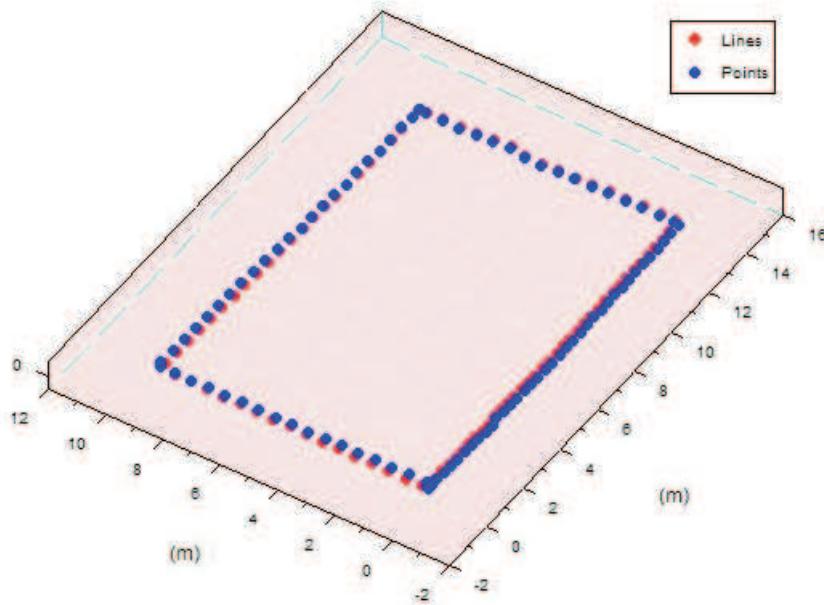


Figura 5.29: Trajetória usando SLAM com linhas e SLAM com pontos.

Por fim, as figuras 5.30, 5.31 e 5.32 apresentam respectivamente as variâncias das componentes  $x_R$ ,  $y_R$  e  $\theta_R$  da pose. Estas figuras mostram que o comportamento da variância da pose foi menor durante a grande maioria do trajeto e esclarecem de outra forma o porque do resultado do método proposto ser mais satisfatório que o método clássico.

## 5.6 Conclusões

Neste Capítulo foi apresentado uma série de resultados obtidos em diversas situações a fim de validar o SLAM visual proposto. O sistema foi testado considerando linhas no piso paralelas e perpendiculares. O sistema também foi posto em prova em um trajeto de laço fechado tanto para ambiente interno quanto para ambiente externo. Por fim, comparou-se a abordagem proposta com a solução clássica de SLAM visual usando pontos.

Em todos os testes o sistema se comportou de forma estável e os resultados foram bem precisos. Isto nos encorajou a usar a pose do robô, as incertezas da pose e o resto da imagem processada para construir uma grade de ocupação visual usando uma única câmera. Este tema será discutido no Capítulo seguinte.

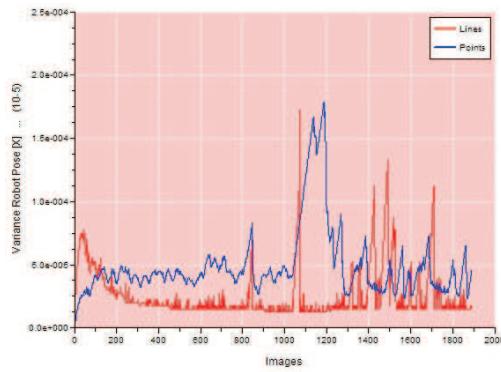


Figura 5.30: Comparação entre as variâncias da coordenada  $x_R$  da pose do robô.

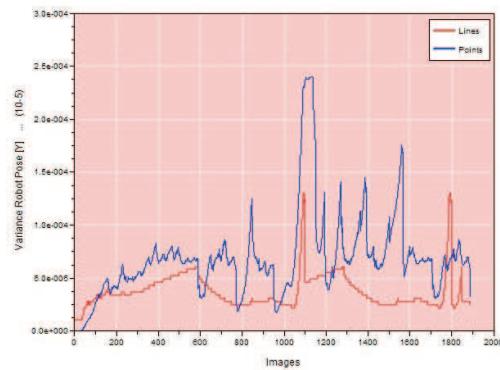


Figura 5.31: Comparação entre as variâncias da coordenada  $y_R$  da pose do robô.

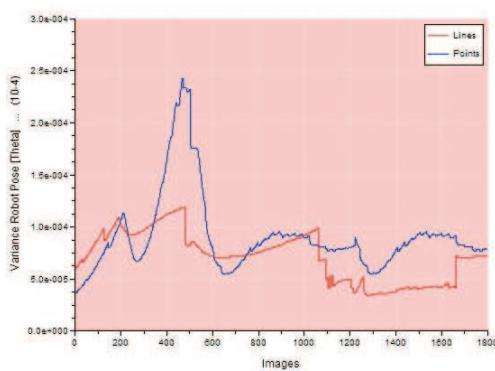


Figura 5.32: Comparação entre as variâncias da coordenada  $\theta_R$  da pose do robô.



---

# Capítulo 6

## Grade de Ocupação Visual

---

*O objetivo deste Capítulo é apresentar a construção de uma grade de ocupação probabilística 2D usando visão monocular. A modelagem pressupõe o conhecimento da pose do robô, das incertezas da pose e de uma matriz de homografia. A obtenção destes três requisitos é possível a partir do SLAM visual apresentado no Capítulo 4. Resultados serão apresentados a fim de validar a proposta.*

### 6.1 Grade de Ocupação

A investigação na construção automática de mapa usando robôs tem uma longa história. Um representante precoce da abordagem de mapeamento é Alberto Elfes, que apresentou um importante algoritmo para modelagem do ambiente utilizando grades de ocupação [Elfes 1987, Elfes 1989]. Com essa forma de representação, os espaços contínuos do ambiente são discretizados, de maneira que o ambiente passa a ser representado sob a configuração de uma grade ou matriz multi-dimensional (2D ou 3D). Cada elemento da matriz, também chamado de célula, representa um local do ambiente que pode estar, de acordo com uma formulação probabilística, ocupado, vazio ou pode ainda não ter sido explorado. A abordagem de Elfes impulsionou muitos outros trabalhos na década de 90 como os de Borenstein & Koren (1991), Buhmann et al. (1995) e Burgard et al. (1997).

A figura 6.1 mostra um exemplo de como seria a representação em uma grade de ocupação de uma parte de um ambiente com dados obtidos por um feixe de sonar. Na figura 6.1, as células em preto representam objetos detectados pelo sonar, as células em branco representam regiões livres ou não ocupadas por objetos e, por fim, as células em cinza representam regiões ainda não mapeadas pelo sonar. O modelo espacial probabilístico baseado em grade de ocupação apresenta-se como um mapa que pode ser usado diretamente para tarefas de navegação, tais como planejamento de caminho com desvio de obstáculo e estimativa de posição [Elfes 1989].

Um problema enfrentado por Elfes (1989) foi a existência de incertezas nas medidas retornadas pelos sonares. Embora tenham adotado métodos para filtrar informações não-desejadas, a informação do feixe sonoro fornecia apenas um dado indireto sobre uma região ocupada do ambiente (por exemplo, o sonar apenas fornece a informação de distância entre o robô e o obstáculo, mas não informa sobre a posição deste obstáculo). Como um feixe do sonar só informava se um volume do ambiente de trabalho estava

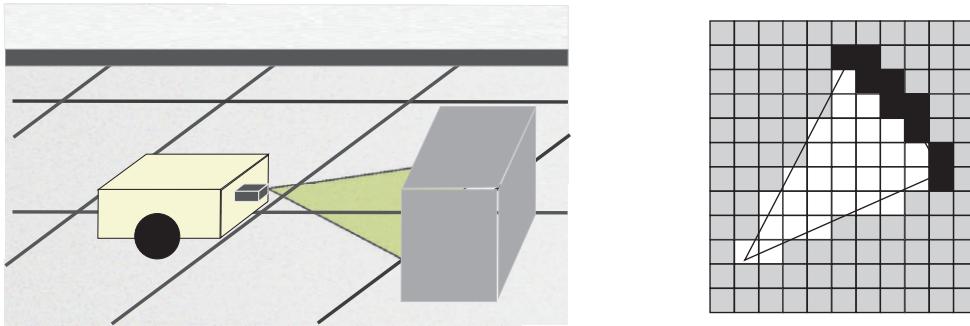


Figura 6.1: Representação de um feixe de sonar em uma grade de ocupação.

vazio ou ocupado, Elfes (1989) modelou o feixe usando funções de distribuição de probabilidade. Tais funções descreviam a confiança de que vários pontos dentro do cone (modelo matemático do feixe) representavam regiões livres de obstáculos, e a incerteza sobre a localização de um ponto (obstáculo real) sobre a superfície deste cone.

As informações retornadas pelas funções de distribuição de probabilidade sobre os volumes do espaço real eram projetadas em uma matriz, com células descrevendo uma área pré-definida do ambiente [Thrun et al. 2005]. Essas probabilidades eram combinadas com as probabilidades de outros feixes sonoros para a mesma região do espaço real. Portanto, o mapa resumia-se a uma matriz cujas células armazenavam valores entre  $[-1, 1]$ , correspondendo a probabilidade da região estar vazia (valores negativos) ou ocupada (valores positivos). Células com valor nulo representavam o desconhecimento sobre a ocupação de regiões não investigadas pelo robô.

A razão dessa popularidade das técnicas probabilísticas vem do fato de que o processo de mapear ambientes sofre uma forte influência das incertezas presentes nas informações sensoriais e, com as técnicas probabilísticas, é possível tratar esse problema modelando explicitamente as diversas fontes de ruídos e seus efeitos nas medições [Thrun et al. 2005]. Assim, os algoritmos probabilísticos têm se apresentado como uma boa alternativa para tratar o problema das incertezas e ruídos, produzindo resultados mais satisfatórios para o problema de mapeamento.

## Formulação Probabilística

A fórmula probabilística padrão para representar o mapeamento com grade de ocupação é dada pela expressão matemática mostrada na equação (6.1) [Elfes 1989, Thrun et al. 2005]. É oportuno lembrar que o algoritmo supõe o conhecimento da pose do robô.

$$P(\mathbf{M}|\mathbf{Z}_{0:t}) \quad (6.1)$$

Nesta equação,  $\mathbf{M}$  representa o mapa adquirido e  $\mathbf{Z}_{0:t}$  é o conjunto das medições sensoriais até o instante de tempo  $t$ . É importante deixar claro que o algoritmo supõe que as coordenadas cartesianas e a orientação do robô são conhecidas.

O espaço contínuo do ambiente no qual o robô se encontra é discretizado em células, as quais, juntas, representam de forma aproximada o ambiente mapeado. Essa discretiza-

ção pode corresponder a um corte longitudinal do ambiente 3D no plano dos sensores, no caso de ser uma grade 2D, ou pode ser uma discretização 3D do ambiente, no caso de uma grade 3D. Isso depende das características e/ou modelo dos sensores utilizados.

Considerando a discretização do ambiente em células, pode-se definir o mapa como um conjunto finito de células  $m_{x,y}$ , onde cada uma possui um valor agregado que corresponde à probabilidade de ela estar ocupada. As células podem possuir valores entre o intervalo de 0 a 1. O valor 0 significa que a célula está vazia e 1 que a célula está ocupada.

Sendo o mapa um conjunto finito de células, o problema de mapeamento pode ser decomposto em um conjunto de problemas de estimativa do valor de cada célula do mapa. A equação (6.2) representa uma instância da estimativa do valor de uma célula  $m_{x,y}$ , e é traduzida como a probabilidade da célula  $m_{x,y}$  estar ocupada, quando são dadas as medições sensoriais  $z_{0:t}$  até o instante  $t$ .

$$P(m_{x,y}|z_{0:t}) \quad (6.2)$$

Por razões de instabilidade numérica com probabilidades próximas de 0 ou 1, é comum calcular o *log-odds* (probabilidade logarítmica) de  $P(m_{x,y}|z_{0:t})$  ao invés de  $P(m_{x,y}|z_{0:t})$ . O *log-odds* é definido por:

$$l_{x,y}^t = \log \frac{P(m_{x,y}|z_{0:t})}{1 - P(m_{x,y}|z_{0:t})} \quad (6.3)$$

O valor da probabilidade de ocupação da célula pode ser recuperado através da equação (6.4).

$$P(m_{x,y}|z_{0:t}) = 1 - \frac{1}{e^{l_{x,y}^t}} \quad (6.4)$$

O valor da probabilidade pode ser estimado recursivamente em qualquer instante  $t$  pela regra de Bayes aplicada a  $P(m_{x,y}|z_{1:t})$  (equação 6.5).

$$P(m_{x,y}|z_{0:t}) = \frac{P(z_t|z_{0:t-1}, m_{x,y})P(m_{x,y}|z_{0:t-1})}{P(z_t|z_{0:t-1})} \quad (6.5)$$

$P(z_t|z_{0:t-1}, m_{x,y})$  representa o modelo probabilístico do sensor de alcance,  $P(m_{x,y}|z_{0:t-1})$  é o valor de ocupação da célula  $m_{x,y}$  no instante  $t-1$  e  $P(z_t|z_{0:t-1})$  é o valor real medido pelo sensor.

Supondo que o mapeamento seja realizado em ambientes estáticos, logo, vale a assertiva de que a medição atual do sensor é independente das medições passadas, dado o mapa  $m$ , em qualquer instante, isso implica nas equações (6.6) e (6.7).

$$P(z_t|z_{0:t-1}, m) = P(z_t|m) \quad (6.6)$$

$$P(z_t|z_{0:t-1}) = P(z_t) \quad (6.7)$$

Visto que o mapa é decomposto em células, essa suposição pode ser estendida como mostra a equação (6.8).

$$P(z_t|z_{0:t-1}, m_{x,y}) = P(z_t|m_{x,y}) \quad (6.8)$$

Com base nas suposições feitas, a equação (6.5) pode ser simplificada resultando na equação (6.9).

$$P(m_{x,y}|z_{0:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|z_{0:t-1})}{P(z_t)} \quad (6.9)$$

A aplicação da regra da Probabilidade Total à equação (6.9) implica na equação (6.10). Nela é computada a probabilidade de a célula  $m_{x,y}$  estar ocupada, tendo como base o modelo probabilístico do sensor  $P(z_t|m_{x,y})$  e o valor de ocupação da célula disponível *a priori*  $P(m_{x,y}|z_{0:t-1})$ .

$$P(m_{x,y}|z_{1:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|z_{0:t-1})}{\sum_{m_{x,y}} P(z_t|m_{x,y})P(m_{x,y}|z_{0:t-1})} \quad (6.10)$$

## 6.2 Trabalhos Relacionados

Em um trabalho mais antigo de Oriolo et al. (1997), os autores mostram que é possível formular e resolver problemas de percepção e planejamento lidando com suas incertezas usando a teoria dos conjuntos (lógica fuzzy). O mapa construído por essa técnica é definido como um conjunto fuzzy, no qual cada ponto é associado a um número real que quantifica a possibilidade de ele pertencer a um obstáculo. A principal vantagem apresentada está na possibilidade de se utilizar vários tipos de operadores fuzzy para modelar as incertezas e agregar informações provenientes de múltiplas fontes.

Konolige (1997) propõe um método que trata de um refinamento matemático do método de mapeamento apresentado por Elfes [Elfes 1987], nomeado de MURIEL (*Multiple Representation Independent Evidence Log*), visando tratar os problemas intrínsecos aos sonares, como múltiplas reflexões e redundância de leituras. Borenstein & Koren (1997) criaram um método baseado em histogramas, cujo principal objetivo é diminuir o custo computacional intrínseco à representação baseada em grade de ocupação. Outra variação pode ser verificada no trabalho de Yguel, Aycard & C. (2006). Este último trabalho aborda os problemas da representação e armazenamento de dados para mapas grandes e, para isso, propõe uma forma de representação em grade de ocupação baseada em wavelets: Wavelet Occupancy Grids.

Sobre representação de ambiente é importante pensar que além de pesquisar dados para o mapeamento, é possível, por exemplo, associar informações auxiliares como deformação do solo, umidade e temperatura. Esse conceito de unir informações auxiliares é de difícil incorporação dentro do SLAM tradicional. Estrada et al. (2005) elaboraram um método chamado DenseSLAM para permitir esse tipo de incorporação. Como o robô se move através do ambiente, dados auxiliares são armazenados em uma estrutura de dados apropriada, tal como grade de ocupação, e a região representada por cada rede de células é determinada por um conjunto de marcos locais do mapa. O resultado é uma capacidade de manter constantemente a localidade espacial da informações através da grade de ocupação.

Outras vertentes de investigações são relacionadas à confecção de mapas híbridos. Esta representação enfoca na possibilidade de prover subsídios para as tarefas de mais alto nível do robô, por exemplo, planejamento de caminhos. Ahn, Chung & Oh (2007) constroem o mapa com características de pontos 3D e linhas para ambientes internos. Já Wurm et al. (2007) representam um mapa de grade e de características, onde a característica mapeada é o centro de um loop de exploração. Blanco et al. (2007) apresentam uma representação métrica e topológica para um grande experimento de SLAM.

Utilizando uma formulação de força que diminui ao longo do tempo, Andrade-Cetto & Sanfeliu (2002) mostram uma solução para mapeamento de ambientes dinâmicos utilizando grade de ocupação. A cada reobservação de um marco a força associada a ele é reforçada e um filtro de Kalman é utilizado para filtrar marcos com baixa força. Canas & Matellan (2006) fazem uma comparação entre várias técnicas de mapeamento de ambientes dinâmicos utilizando grade.

Ainda sobre grade para ambientes dinâmicos, Mitsou & Tzafestas (2007) propõem um algoritmo capaz de mapear ambientes onde a posição dos objetos podem mudar ao longo do tempo. Os autores classificam os objetos como sendo: estáticos, objetos de baixa dinâmica (cadeiras ou portas, por exemplo) e objetos de alta dinâmica (seres humanos, por exemplo). O algoritmo identifica corretamente os modelos da dinâmica dos objetos no ambiente e insere essas informações em um mapa de grade a fim de registrar uma dimensão temporal e poder utilizar estas informações nas tarefas do robô.

Yguel, Aycard & Laugier (2006a) propõe a utilização de GPUs para fundir informações e construir um mapa de grade, desocupando o processador para que ele possa ser utilizado em outras tarefas endereçadas ao robô.

Na vertente de processamento de imagens, o trabalho de Merhy et al. (2008) apresenta estratégias de refino de grades de ocupação baseados em segmentação probabilística. Os autores consideram que as probabilidades associadas a cada célula do mapa de grade podem ser codificadas como níveis de textura de uma imagem.

Sabo (2002) utiliza uma única câmera para a construção de uma grade de ocupação. Pontos notáveis são identificados na imagem através de técnicas de diferença de gradientes de cor e são marcados na grade junto com uma região de incerteza associada à odometria. Braillon et al. (2006) detectam obstáculos fundindo informações de visão estéreo e fluxo ótico para representar uma grade de ocupação e em um trabalho posterior Braillon et al. (2008) propõe uma estratégia de não calculá-lo a cada passo devido ao custo computacional elevado.

Blanco & Gonzalez (2007) apresentam uma técnica de matching para mapas construídos com grade de ocupação baseado em imagens armazenadas durante o processo. O diferencial deste trabalho é que os mapas podem ser comparados mesmo que tenham sido discretizados de forma diferente.

Propondo a utilização de uma câmera como um sonar visual, Choi & Oh (2006) montam um mapa de um ambiente doméstico utilizando como uma única fonte sensorial uma câmera. O objetivo era construir um mapa de grade, e não de características, baseado apenas em informações visuais e comparar o método com mapas produzidos por sonares. O sonar visual foi modelado como um conjunto de raios virtuais na imagem e a detecção dos pontos limites (pontos finais) eram encontrados utilizando a diagonal de cor de Ma-

halanobis. Os autores comentam que o problema desta abordagem é a detecção de pontos espúrios e sugerem técnicas de filtragem para eliminá-los.

Outros trabalhos, propõem a construção de mapas volumétricos a partir de grades de ocupação [Fleuret et al. 2008, Marks et al. 2008]. No primeiro, uma grade 3D é construída, utilizando múltiplas câmeras, para fazer o rastreamento de pessoas em um ambiente outdoor. O segundo, constrói um mapa de grande, utilizando visão estéreo, com a informação da variância da altura do terreno. Essa informação é útil devido a necessidade de calcular a probabilidade de travessia de um célula do mapa considerando que o ambiente mapeado é defeituoso.

Em um trabalho mais recente, Hata & Wolf (2010b) propõem a utilização de uma grade 2D construída a partir de nuvens de pontos 3D visando identificar regiões navegáveis de um determinado terreno. A partir de informações dessa grade o robô pode inferir níveis de navegação: navegável, parcialmente navegável e não navegável.

Conforme os trabalhos já apresentados, verifica-se na literatura que muitos deles envolvendo mapeamento estão utilizando câmeras como o principal sensor. A seguir apresentaremos de forma mais específicas trabalhos relacionados à mapeamento visual.

Mapeamento é uma área de investigação muito ativa em Robótica e na Inteligência Artificial. Há pelo menos duas décadas e, apesar dos progressos significativos nesta área, ainda existem grandes desafios. Um deles está relacionado ao custo das soluções existentes e o outro à qualidade dos mapas encontrados. Baseado nisto, soluções utilizando câmeras como principais sensores mostram-se muito interessantes.

Embora apresentem muitas vantagens, as soluções baseadas em sensores ópticos apresentam problemas como: a distorção da lente, o campo de visão restrito e baixas resoluções. Maior detalhamento sobre essas dificuldades, bem como soluções para eles podem ser encontrado em Taylor et al. (2007).

Horswill (1993) comenta que ambientes *indoor* possuem duas propriedades que devem ser exploradas: a) Terreno plano: ambientes internos raramente têm pisos irregulares e são normalmente constituídos por um ou mais planos bem definidos ligados por elevadores ou escadas. Esse pressuposto permite que o ambiente possa ser modelado por um conjunto de mapas em duas dimensões; b) Texturas: ambientes internos possuem normalmente texturas uniformes e são normalmente revestidos por azulejos, ladrilhos ou carpete. A informação de textura pode ser usada para classificar áreas como ocupadas ou desocupadas.

Exemplos de trabalhos de mapeamento visual baseado em textura são os de Howard & Kitchen (1997) e Dollner & Hinrichs (2000). A utilização de características geométricas para construção de mapas visuais também é encontrada na literatura. Em Smith et al. (2006) linhas horizontais e verticais são extraídas como marcos. As verticais são utilizadas para extrair as portas e as horizontais para estimar o teto e o piso. Jeong & Lee (2006) comentam que a utilização de linhas é interessante, pois são menos sensíveis a ruídos quando comparadas a pontos. Isso acontece porque são constituídas por um conjunto de pontos, ou seja, o ruído em um ponto geralmente não afeta a posição e orientação da linha substancialmente. As linhas também fornecem informações sobre a geometria do ambiente permitindo mais sofisticadas inferências sobre a estrutura do mundo. Além disso, elas facilmente observadas de posições distintas do ambiente.

Alguns trabalhos de mapeamento visual utilizam segmentação de imagens para detectar áreas livres. Martin (2002) constatou que o chão em um ambiente indoor normalmente apresenta cor uniforme e que arestas são os melhores indicadores para as paredes do ambiente. Taylor et al. (2007) utilizam segmentação de imagem para incorporação de espaços livres em mapa global. Isto é realizado em duas etapas: primeiro, um extrator de bordas é utilizado (Canny) para identificar as fronteiras e em seguida, utilizando a informação da imagem segmentada, preenche-se no mapa os espaços livres. Além dessa informação visual um sonar é utilizado para dar mais confiança aos dados extraídos da imagem.

Sim & Dudek (2008) analisaram o problema de automaticamente construir um mapa visual do ambiente com especial atenção à seleção de uma política de exploração que equilibra precisão e eficiência. Eles apresentam vantagens e desvantagens de exploração baseada em trajetórias retas, circulares concêntricas, em formato de oito, randômica, triângulo e estrela.

Ahn & Chung (2007) propõem a construção de um mapa híbrido formado por um mapa absoluto e outro relativo. O mapa absoluto é definido em coordenadas do mundo. Este mapa é descrito como planos verticais formado por saliências visuais de características localizadas em paredes ou planos de mobiliário em um ambiente indoor. Já o mapa relativo é definido em coordenadas de cada plano. Contém informações associadas a cada plano individualmente. A vantagem desta abordagem consiste em diminuir o espaço de busca das características do ambiente já que esta busca é feita em um mapa local.

### 6.3 Sistema Proposto

A modelagem da grade de ocupação probabilística 2D obtida a partir de um sistema de visão monocular pressupõe o conhecimento da pose do robô, das incertezas da pose e de uma matriz de homografia. Neste caso, a obtenção destes três requisitos é possível a partir do SLAM visual apresentado no Capítulo 4.

O sistema de SLAM apresentado anteriormente usa uma matriz de homografia, calculada no início da exploração, para fazer diretamente o mapeamento imagem-mundo das linhas existentes no chão. Além disso, um algoritmo de segmentação é usado para classificar o mundo em “chão” e “não-chão”. A partir desta classificação é possível usar a matriz de homografia para mapear como células livres a parte “chão” da imagem e também é possível usá-la para mapear as fronteiras das partes “não-chão” como obstáculos do mundo. A figura 6.2 mostra o esquema desta proposta de construção de uma grade de ocupação visual.

As etapas necessárias para a construção da grade de ocupação apresentada nesta tese são: discretizar a imagem em células, classificar as células da imagem em “chão” e “não chão”, qualificar a parte “chão” como células livres e a parte “não chão” como células não-livres, reclassificar as células não-livres como *obstáculo* ou *occluso*, mapear as células *obstáculo* para o mundo usando uma matriz de homografia e, por fim, expandir as células *obstáculo* usando as incertezas da pose do robô.

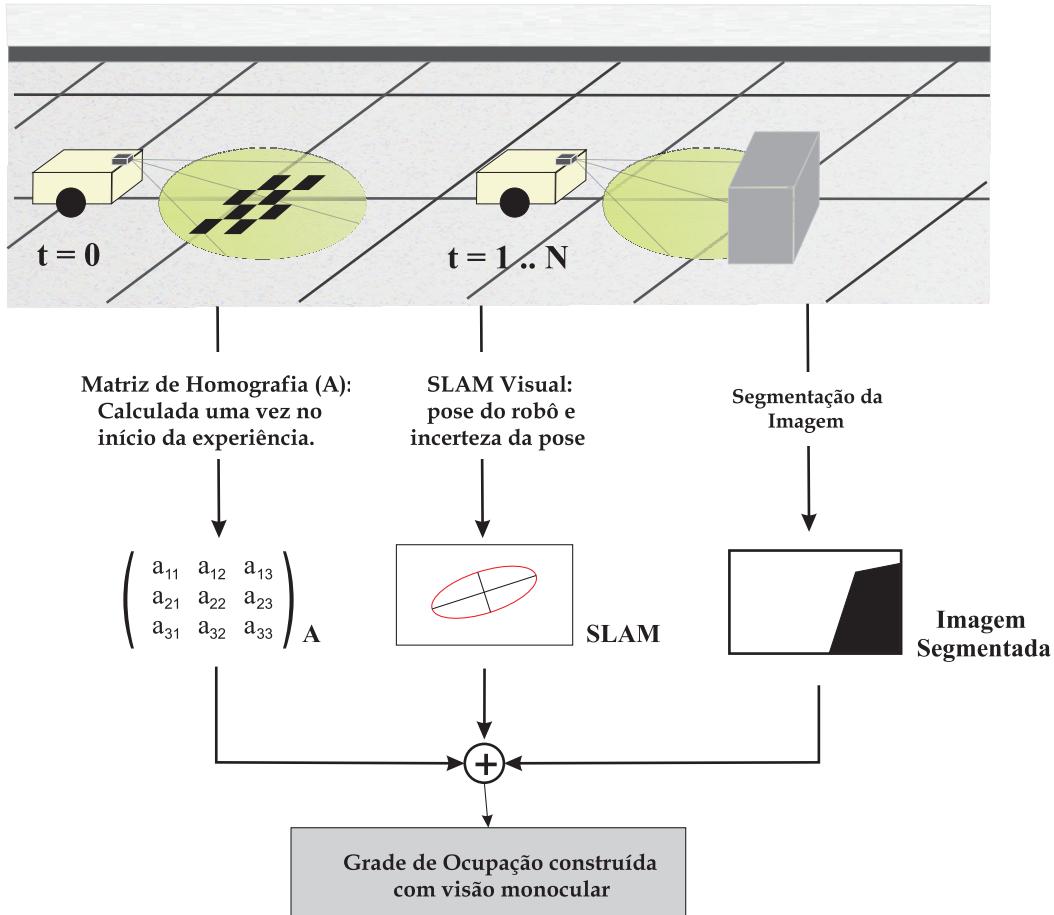


Figura 6.2: Esquema de construção da grade de ocupação.

## Discretização e Classificação

Nesta abordagem, o processamento de imagem é uma etapa fundamental. Para a discretização da imagem utilizou-se células (porções quadradas de tamanho fixo da imagem) e para a classificação foi feita uma análise das componentes de cor dessas células.

Utilizou-se a idéia de considerar a área imediatamente abaixo do robô como uma área livre e assim, esta área pode ser usada como padrão para a classificação. O padrão é mapeado na imagem em uma área pré-definida e o algoritmo *Mean Shift* é usado para que se possa calcular as médias e variâncias das componentes de cor desta região. A figura 6.3 ilustra o comentado.

A estimativa do gradiente de uma função de densidade, conhecida por *Mean Shift*, foi desenvolvida inicialmente por Fukunaga & Hostetler (1975). Para Comaniciu & Meer (1999), o *Mean Shift* pode ser definido como um estimador não-paramétrico de densidade de gradiente e é comumente empregado como filtro de preservação de descontinuidade e segmentação de imagens, ao longo do domínio espacial de cores e níveis de cinza.

Ukrainitz & Sarel (2004) propõem uma descrição intuitiva do algoritmo a partir de um exemplo simples. Supõe-se que, a partir de uma distribuição de bolas de bilhar idênticas, escolha-se uma região de interesse qualquer.

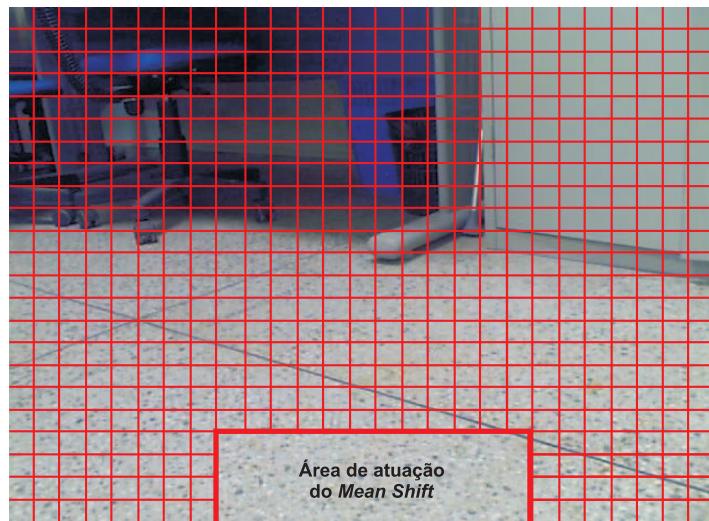


Figura 6.3: Área de atuação do algoritmo *Mean Shift*.

O algoritmo, através da utilização de um histograma de cores, detecta a região com maior densidade de pontos de uma determinada cor e calcula uma média ponderada para se achar o centro de massa desta região. Logo após é traçado um vetor entre o centro da região de interesse e o centro de massa obtido, conhecido como vetor de *Mean Shift*.

A seguir, a região de interesse inicial é transladada até que seu centro coincida com o centro de massa, atingindo uma convergência (figura 6.4). Este processo se repete, até que não se encontre regiões com maior densidade do que a região já obtida. O *Mean Shift* trata-se de um processo vantajoso, pois não é necessário uma avaliação da imagem por inteiro, de forma que se ganha em rapidez.

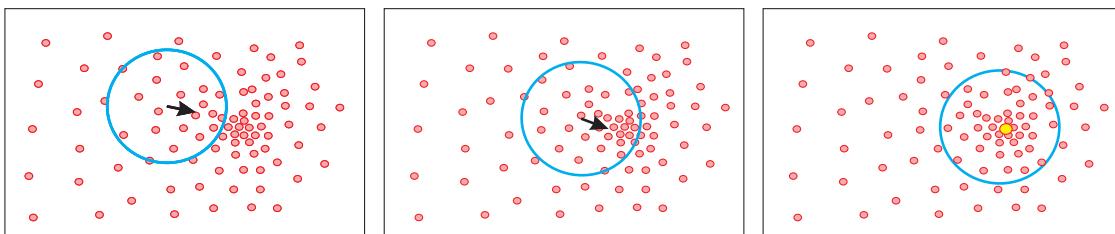


Figura 6.4: Funcionamento do algoritmo *Mean Shift*.

Uma vez calculada as variáveis de cor (média e variância de R, G e B) da região padrão usando o *Mean Shift*, estas variáveis são usadas como pontos de comparação entre esta região e as outras partes da imagem.

A imagem capturada pela câmera é discretizada em células e a comparação entre as componentes de cor da região padrão com as componentes de cor de cada célula é usada para classificá-las como livres ou não-livres.

A fim de encontrar um tamanho ideal da discretização de cada célula na imagem, ou seja, encontrar o tamanho da janela, em pixels, que será usada para fazer a comparação, foi realizado um estudo com diversas medidas.

Usando janelas de  $5 \times 5$  pixels o tempo médio de processamento foi de  $600ms$ ; janelas de  $10 \times 10$  pixels resultaram em um tempo médio de  $300ms$ ; com janelas de  $15 \times 15$  o tempo médio foi de  $220ms$  e com janela de  $20 \times 20$  pixels o tempo foi de  $170ms$ . Com base neste último valor, este foi o tamanho padrão das janelas utilizadas. A figura 6.5 mostra o gráfico do tempo de processamento usando cada uma das medidas descritas. Nela, além de constatar os valores apresentados, também é possível notar que o sistema possui tempo de processamento estável.

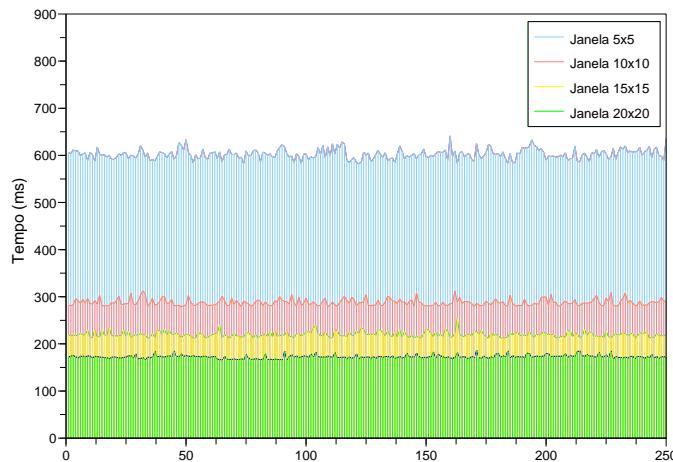


Figura 6.5: Gráfico do tempo de processamento do sistema de segmentação.

Para exemplificar o resultado do algoritmo de classificação usado neste trabalho, a figura 6.6 mostra a uma imagem contendo objetos de formas mais clássicas e a figura 6.7 mostra uma imagem contendo um objeto com uma forma mais complexa. A região em preto foi classificada como chão e a região em branco foi classificada como não chão.

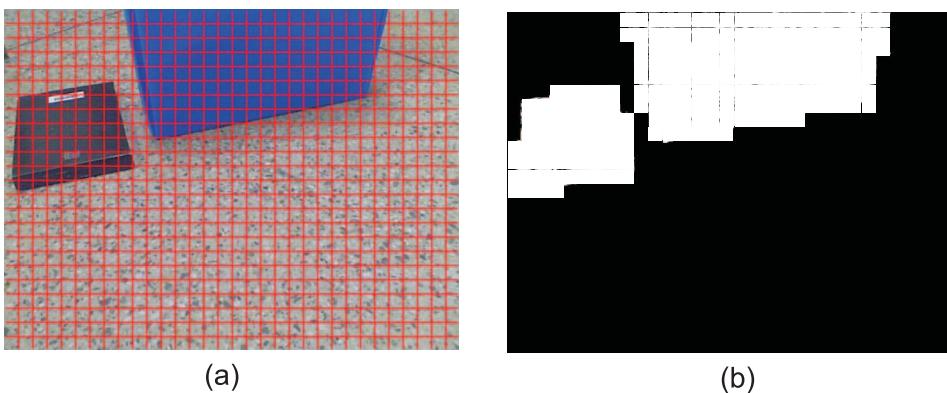


Figura 6.6: Classificação da imagem: (a) Imagem original discretizada em células, (b) Imagem classificada usando o algoritmo proposto.

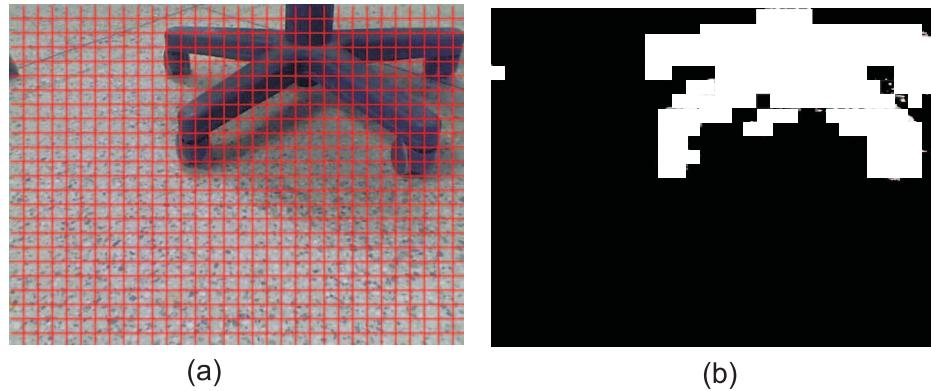


Figura 6.7: Classificação de imagem: (a) Imagem original discretizada em células, (b) Imagem classificada usando o algoritmo proposto.

### Mapeamento Imagem-Mundo e Reclassificação

Um problema inerente a esta abordagem é o efeito da inclinação da câmera que faz com que os pontos na imagem representem uma forma diferente no mundo. A figura 6.8(a) mostra pontos distribuídos uniformemente na imagem de  $640 \times 480$  pixels e a figura 6.8(b) apresenta o mapeamento destes pontos no mundo usando uma matriz de homografia. A matriz homográfica foi calculada para que o campo de visão da câmera fosse de 1m.

Observando essa figura é possível perceber que uma célula na imagem (conjunto de quatro pontos) pode ser mapeada em mais de uma célula da grade no mundo.

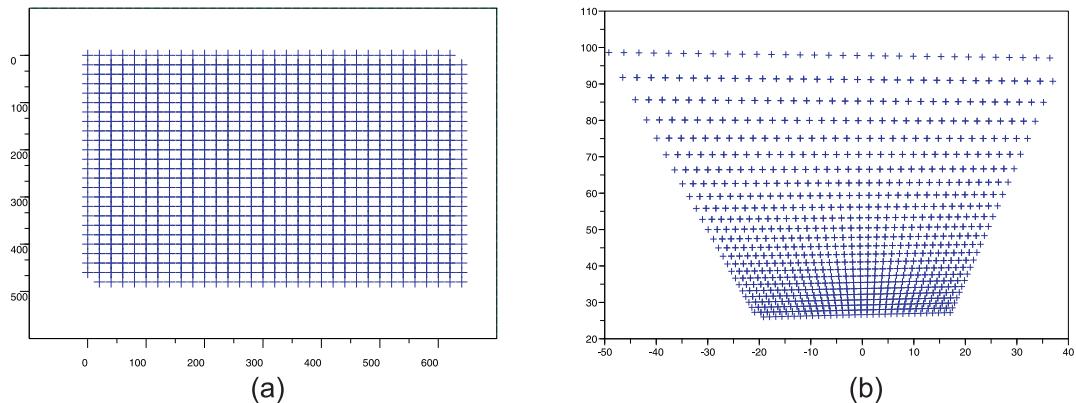


Figura 6.8: Efeito da inclinação da câmera em um conjunto de pontos.

A figura 6.9(a) ilustra uma grade em uma imagem onde os quadrados possuem 20 pixels de lado e a figura 6.9(b) mostra que o mapeamento de uma célula em uma grade de 10cm de discretização pode ser feito em mais de uma célula e que muitas células podem ser mapeadas em uma única célula da grade.

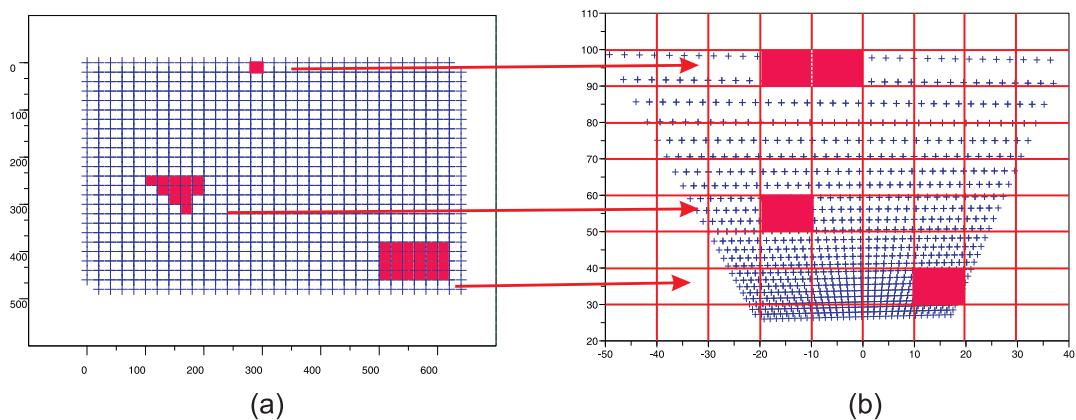


Figura 6.9: Efeito da inclinação da câmera no mapeamento imagem-mundo.

Outro problema desta abordagem é que um obstáculo na imagem é mapeado para o mundo de forma projetiva (figura 6.10). Assim, todas as células na imagem que são consideradas não livres não podem ser mapeadas na grade 2D, pois dispõe-se apenas de informação da região fronteira entre os obstáculos e o chão.

Para sanar este problema foi implementado um algoritmo para reclassificar as células não livres como *obstáculo* ou *oclusa*.

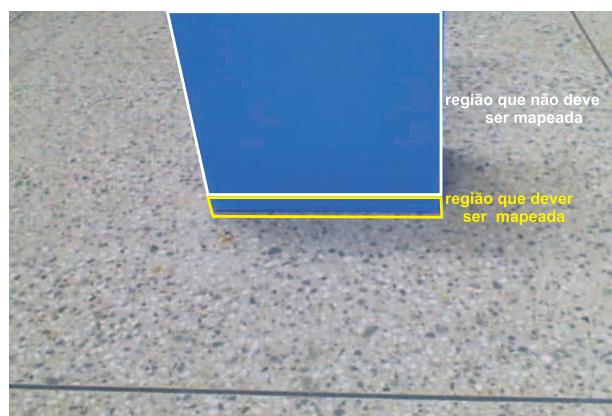


Figura 6.10: Região de mapeamento imagem-mundo.

A idéia do algoritmo consiste em calcular um vetor cujo módulo é metade do tamanho da discretização da grade e no sentido do ponto  $(0, 0)$  e, em seguida, testar se a extremidade do vetor está contida em uma célula livre ou em uma célula não livre. Se a extremidade estiver dentro de uma célula não livre esta célula é considerada *occlusa* e não será mapeada efetivamente na grade; caso contrário, a célula é considerada *obstáculo* e é mapeada na grade. A figura 6.11 ilustra o algoritmo.

Inicialmente todas as células não livres foram mapeadas na grade, em seguida foi calculado para cada célula não livre o vetor de fronteira para que pudesse servir de teste para a classificação em *occlusa* ou *obstáculo*. Por fim, apenas as células consideradas *obstáculo* foram utilizadas para a construção do mapa.

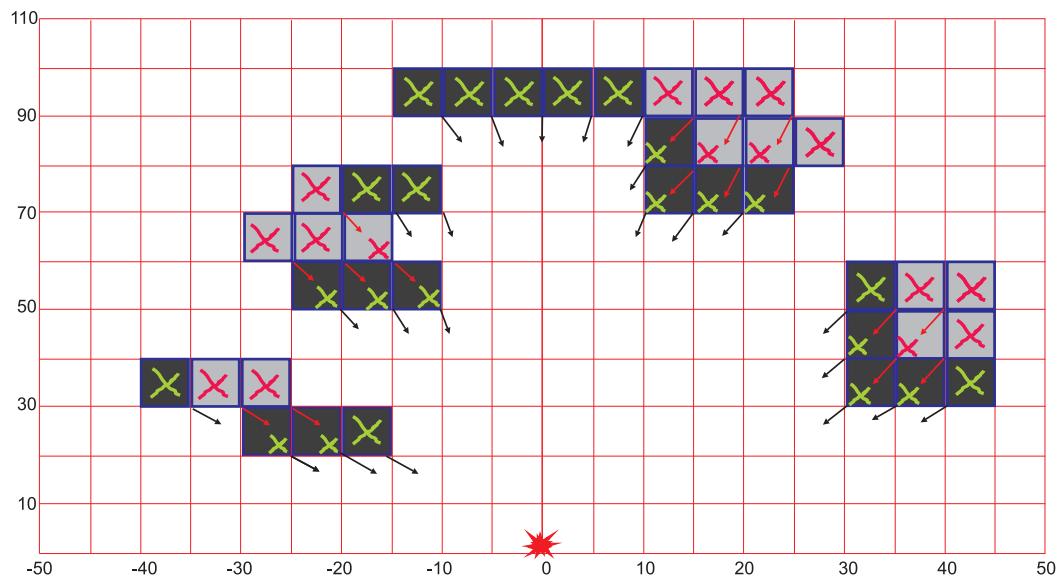


Figura 6.11: Ilustração do funcionamento do algoritmo de classificação.

## Expansão e Probabilidade de Ocupação

A fim de embutir no processo de construção da grade as incertezas da pose do robô, foram utilizadas as variâncias das componentes  $x$  ( $\sigma_x^2$ ) e  $y$  ( $\sigma_y^2$ ) em uma etapa de expansão dos obstáculos. Embora essas medidas não representem fielmente a incerteza do posicionamento do robô no mundo, pois a representação correta seria a elipse de incerteza, estas medidas servem para representa-la de forma aproximada.

A expansão é feita de forma simples: se  $\sigma_x$  for maior que metade da discretização da grade este valor é incorporado às componentes  $x$  de cada célula ocupada no instante  $t$ ; se  $\sigma_y$  for maior que metade da discretização da grade este valor é incorporado às componentes  $y$  de cada célula ocupada no instante  $t$ .

O passo seguinte consiste em calcular a probabilidade de ocupação de cada célula. Nesta ótica, a câmera utilizada é modelada como um sensor de distância e isto nos permite usar a formulação probabilística clássica proposta por Elfes. Ele comenta que a modelagem dos sensores deve ser feita respeitando o princípio de funcionamento dos mesmos e verificando as restrições que levam ao surgimento de imprecisões. Assim, a modelagem de um sensor de alcance pode ser dada por uma função Gaussiana (equação 6.11):

$$P(r|d_{x,y}) = \frac{1}{\sqrt{2\pi}\sigma_r} \exp\left[-\frac{1}{2}\left(\frac{(r-d_{x,y})^2}{\sigma_r^2}\right)\right] \quad (6.11)$$

onde  $r$  é a distância medida pelo sensor,  $d_{x,y}$  é a distância euclidiana entre a célula em que o sensor se encontra e a célula  $m_{x,y}$  que está sendo analisada e  $\sigma_r$  é variância que representa a imprecisão na distância medida pelo sensor.

O comportamento da variância do sensor é dado pela equação (6.12). Os valores das constantes foram calculado por experimentação sendo  $a = 0.0002$  e  $b = 3$ . A figura 6.12 mostra o gráfico do comportamento da variância considerando as constantes calculadas.

$$\sigma^2(r) = a \cdot e^{b \cdot r} \quad (6.12)$$

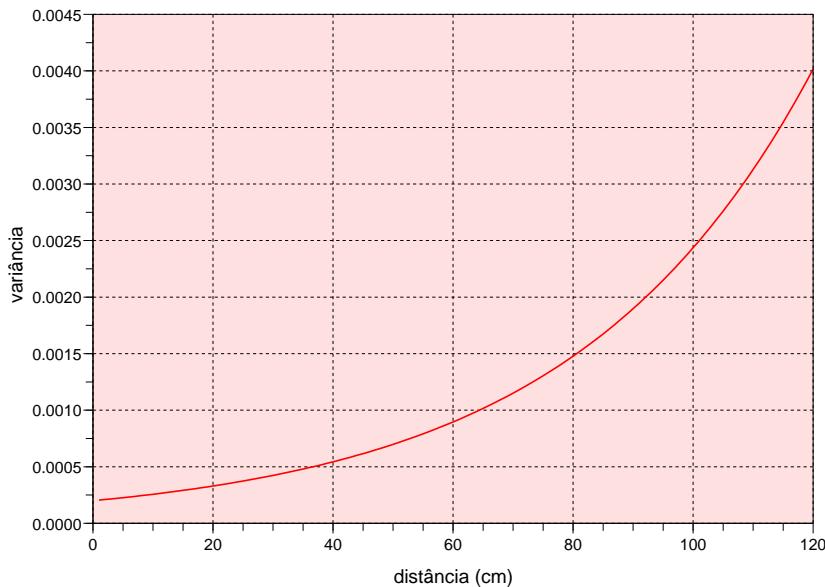


Figura 6.12: Comportamento da variância do sensor.

Finalizando esta seção, a figura 6.13 mostra um conjunto de experiências realizadas com sistema completo em funcionamento, ou seja, a partir das imagens uma grade de ocupação é construída. Ressalta-se que os resultados apresentados consideram o robô parado e com incerteza mínima.

Foram testadas situações em que os obstáculos se encontravam próximos uns dos outros e um pouco mais afastados (figura 6.13 (a) e (b)). Também foram testadas situações onde um obstáculo maior não interferia na visada de um obstáculo menor e a situação inversa (figura 6.13 (c) e (d)).

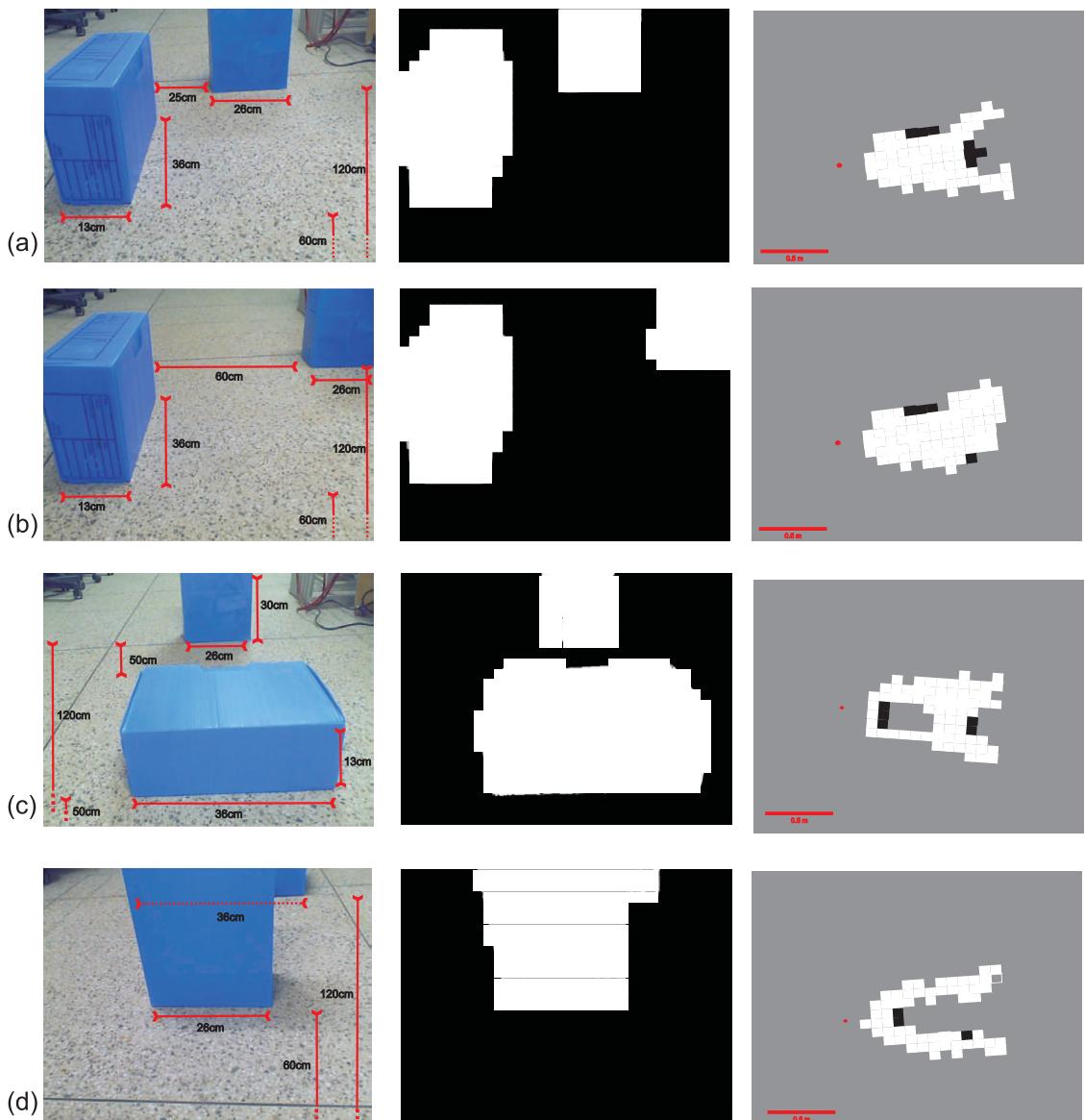


Figura 6.13: Construção de grade visual.

Este último resultado mostra uma vantagem do método proposto em relação às construções tradicionais de grade de ocupação com laser unidimensional ou sonar, porque com estes sensores os dois obstáculos não seriam mapeados simultaneamente dado que as leituras destes tipos de sensores são apenas em uma direção.

## 6.4 Resultados experimentais

Para validar o sistema foram realizadas duas experiências. A primeira foi em um típico ambiente interno onde os obstáculos não estão no mesmo plano de navegação do robô e a segunda foi em um ambiente de circuito fechado.

O algoritmo utilizado nestas experiências para a construção da grade é descrito resumidamente no quadro 6.4 cuja legenda das variáveis utilizadas é apresentada na Tabela 6.1.

Tabela 6.1: Símbolos do algoritmo *grade\_ocupação*

$\mathbf{M}_t$	mapa das células no instante $t$ ;
$\mathbf{I}_t$	imagem capturada pelo sistema de visão no instante $t$ ;
$\mathbf{S}_t$	vetor que contém a pose e as incertezas da pose do robô no tempo $t$ . $\mathbf{S}_t = \{x_t, y_t, \theta_t, \sigma_x, \sigma_y\}$ ;
$\mathbf{A}$	matriz de homografia calculada no início da navegação;
$\mathbf{D}$	vetor de descritores da região de atuação do <i>Mean Shift</i> que contém as médias e variâncias de R, G e B. $\mathbf{D} = \{\mu_R, \mu_G, \mu_B, \sigma_R^2, \sigma_G^2, \sigma_B^2\}$ ;
$\mathbf{d}$	vetor de descritores de cada célula da imagem que contém as médias e variâncias de R, G e B. $\mathbf{d} = \{\mu_R, \mu_G, \mu_B, \sigma_R^2, \sigma_G^2, \sigma_B^2\}$ ;
$v$	ponto da extremidade do vetor com módulo de metade da discretização da grade e aponta para a pose do robô mapeada no mundo;
$I_{C_i}$	conjunto de pontos que representam uma célula na imagem no instante $t$ . $I_{C_i} = \{(u_{min}, v_{min}), (u_{min}, v_{max}), (u_{max}, v_{min}), (u_{max}, v_{max})\}$
$M_{C_i}$	conjunto de pontos que representam uma célula no mundo no instante $t$ . $M_{C_i} = \{(x_{min}, y_{min}), (x_{min}, y_{max}), (x_{max}, y_{min}), (x_{max}, y_{max})\}$
$M_{C_i}$	conjunto de células $M_{C_i}$ resultantes da etapa de expansão de células a fim de incorporar as incertezas da pose do robô. $M_{C_i} = \{M_{C_1}, M_{C_1}, \dots, M_{C_N}\}$
$P_{liv}$	incremento na probabilidade logarítmica de uma célula a cada vez que ela é reconhecida como livre;
$P_{ocp}$	decremento na probabilidade logarítmica de uma célula a cada vez que ela é reconhecida como ocupada;

---

**Algoritmo 3** grade\_ocupaçao( $\mathbf{I}_t, \mathbf{S}_t, \mathbf{A}$ )

---

```

1:  $\mathbf{D} = \text{Calcula\_Descriptores}(ROI(\mathbf{I}_t));$ 
2: for cada célula ( ${}^I c_i$ ) da imagem do
3:    $\mathbf{d} = \text{Calcula\_Descriptores}({}^I c_i);$ 
4:   if ( $\mathbf{D} \cong \mathbf{d}$ ) then
5:      ${}^I c_i = \text{livre};$ 
6:      ${}^M c_i = \text{Converte\_Celulas}({}^I c_i, \mathbf{S}_t, \mathbf{A});$ 
7:      ${}^M C_i = \text{Expande\_Celulas}({}^M c_i, \mathbf{S}_t);$ 
8:      $\text{Atualiza\_Probabilidade}({}^M C_i, P_{liv});$ 
9:      $\mathbf{M}_t = \{{}^M C_i\};$ 
10:    else
11:       ${}^I c_i = \text{não livre};$ 
12:       ${}^M c_i = \text{Converte\_Celulas}({}^I c_i, \mathbf{S}_t, \mathbf{A});$ 
13:       ${}^M C_i = \text{Expande\_Celulas}({}^M c_i, \mathbf{S}_t);$ 
14:       $\mathbf{M}_t = \{{}^M C_i\};$ 
15:    end if
16:  end for
17:  for cada célula não livre de  $\mathbf{M}_t$  do
18:     $\mathbf{v} = \text{Calcula\_Vetor\_Direcao}({}^M c_i);$ 
19:    if  $\mathbf{v} \in \text{célula\_não livre}$  then
20:       $\text{Atualiza\_Probabilidade}({}^M C_i, P_{ocp});$ 
21:    end if
22:  end for

```

---

No tocante às funções utilizadas, ROI é um mapeamento simples da área de atuação do *Mean Shift* e *Calcula\_Descriptores* é usada para calcular as médias e variâncias das componentes de cor R, G e B. *Converte\_Celulas* usa a matriz de homografia e a pose do robô para mapear os pontos da imagem em pontos do mundo enquanto que *Atualiza\_Probabilidade* usa a idéia clássica apresentada por Elfes (1989).

A função *Expande\_Células* merece um comentário especial pois ela é usada para incorporar as incertezas da pose do robô nas células mapeadas. Esta função funciona da seguinte forma: se  $\sigma_x$  for maior que metade da discretização da grade, as coordenadas x da célula são modificadas para  $x_{min} - \sigma_x$  e  $x_{max} + \sigma_x$ , se  $\sigma_y$  for maior que metade da discretização, as coordenadas y da célula são modificadas para  $y_{min} - \sigma_y$  e  $y_{max} + \sigma_y$ . Por fim, as novas células também são usadas na atualização da probabilidade.

Apresentado o algoritmo, o passo seguinte é descrever a primeira experiência que foi realizada em um típico ambiente interno como mostra a figura 6.14. O espaço de navegação do robô possui obstáculos não contidos no plano de movimento do robô e linhas formadas pela junção das cerâmicas do piso são usadas como marcos no algoritmo de SLAM apresentado anteriormente.

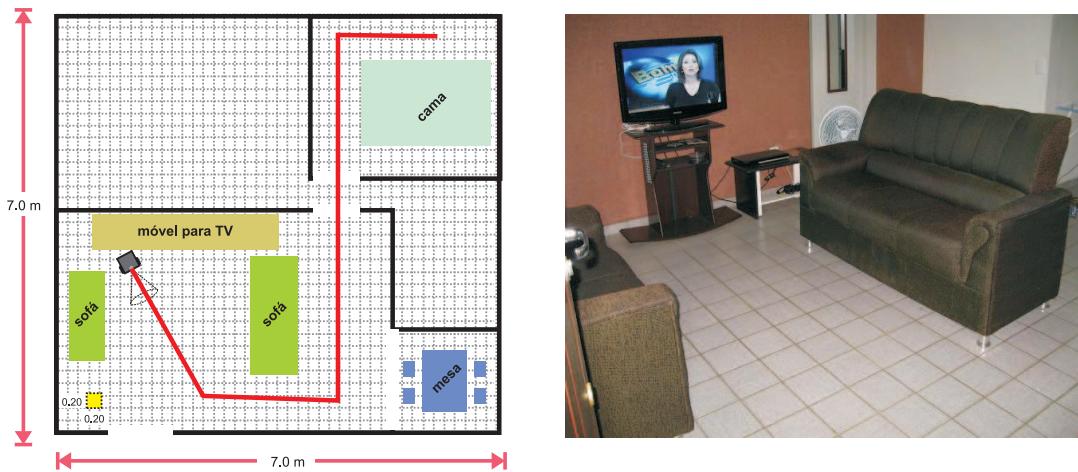


Figura 6.14: Descrição do ambiente do primeiro experimento.

Durante este experimento foram processadas 1238 imagens em um tempo médio de  $360ms$  incluindo o SLAM e a construção da grade. A figura 6.15 mostra diferentes imagens do ambiente tendo em vista a câmera acoplada ao robô.

No processo total de mapeamento dos 53 marcos detectados o índice de correspondência foi bem sucedido em 96% dos casos e cada linha foi observada em 13 imagens consecutivas, em média. Em 98% das imagens o sistema detectou linhas: três linhas em 5%, quatro linhas em 53%, cinco linhas em 24%, seis linhas em 8%, sete linhas em 5% e oito linhas em 3% das imagens.

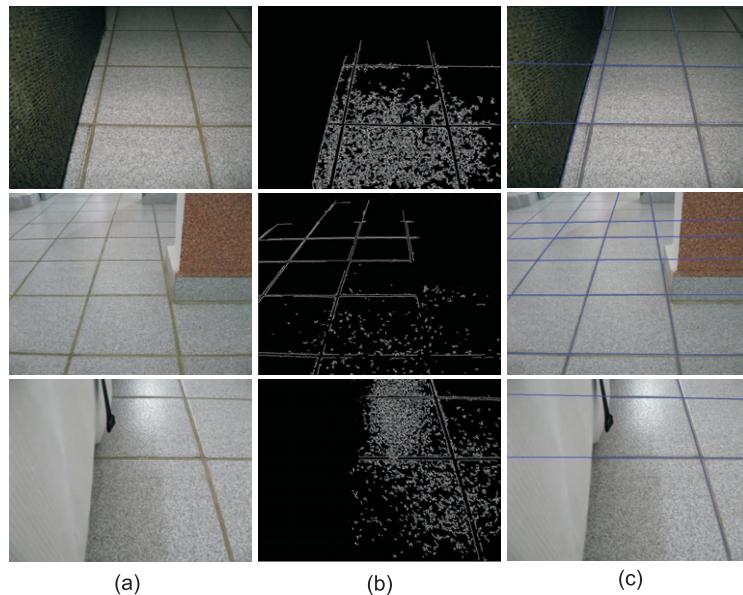


Figura 6.15: Imagem do piso: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

A grade de ocupação construída nesta experiência é exibida na figura 6.16. O resultado foi satisfatório uma vez que é possível identificar nesta imagem algumas características reais do ambiente mapeado como o hall entre os cômodos, as paredes deste hall, as extremidades da cama além da estrutura completa do ambiente navegado.

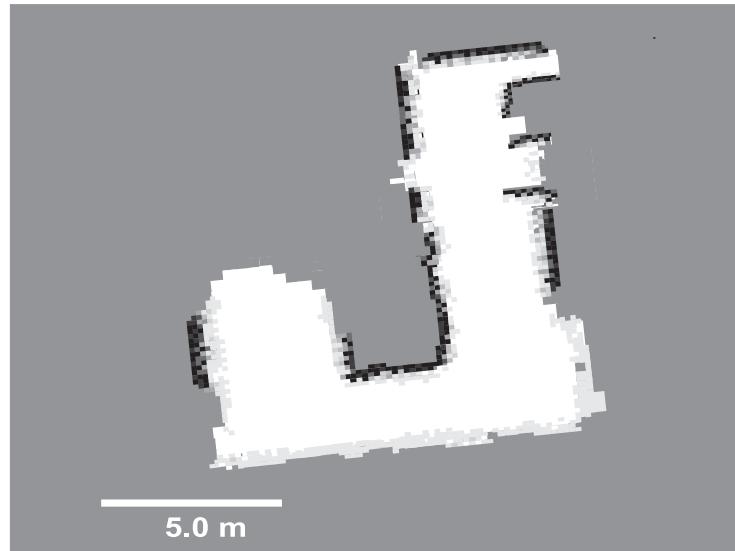


Figura 6.16: Grade de ocupação.

Por fim, com o sistema apresentado é possível construir um mapa híbrido formado pela grade de ocupação e pelo mapa de características com as linhas. O mapa híbrido desta experiência é apresentado na figura 6.17.

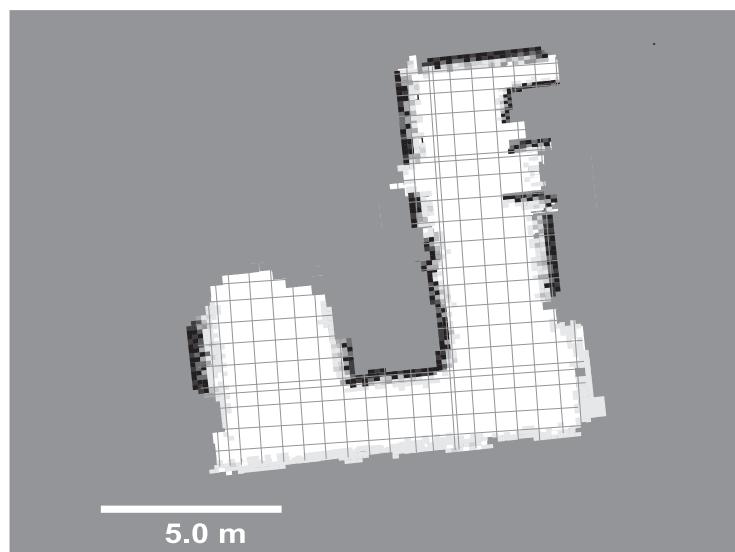


Figura 6.17: Mapa híbrido: Grade de ocupação e Mapa de Características.

Continuando os resultados, foi realizada uma segunda experiência para testar o sistema em uma outra situação. Neste caso, o experimento foi realizado usando os mesmos dados da experiência apresentada no Cenário C do Capítulo 5. Apenas para facilitar a leitura o esquema do local é reexibido na figura 6.18.

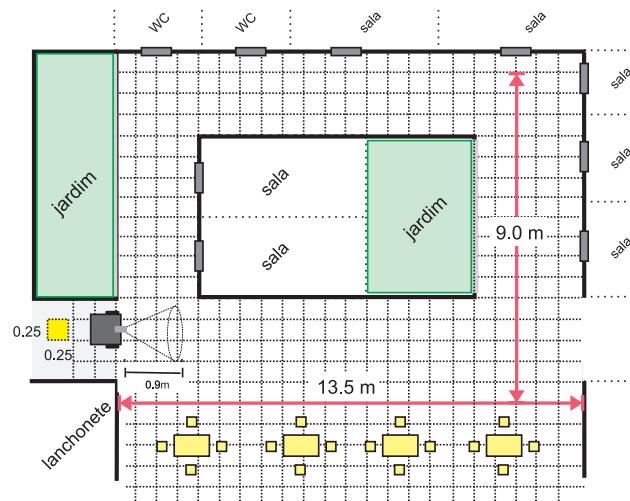


Figura 6.18: Descrição do ambiente.

A idéia deste experimento era mapear não só as paredes mas também as áreas que "não naveáveis" pelo robô. Foi assumido que as áreas com textura diferente do padrão usado para segmentação são áreas não navegáveis e, portanto, são mapeadas como obstáculos. A figura 6.19 mostra imagens ilustrando o comentado.

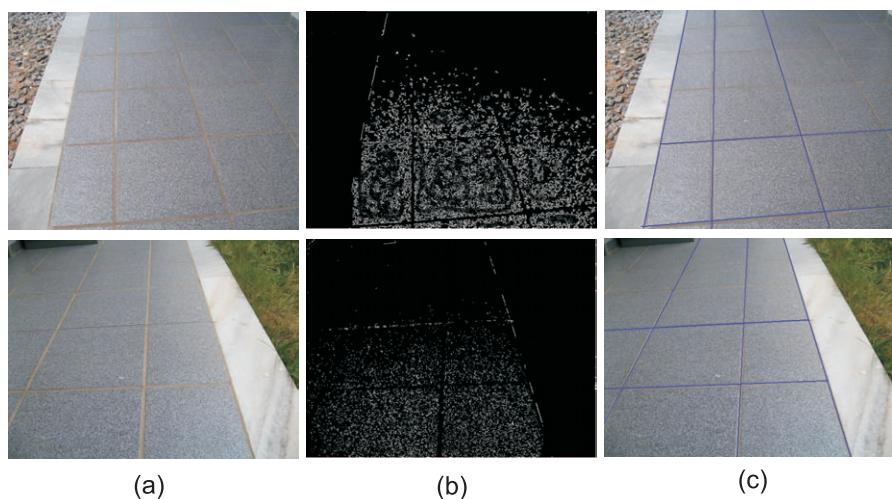


Figura 6.19: Imagem do piso: (a) Imagem original, (b) Imagem processada usando detector de bordas Canny, (c) Imagem processada usando TH.

As figuras 6.20 e 6.21 mostram, respectivamente, o mapa em grade de ocupação e o mapa híbrido deste experimento.

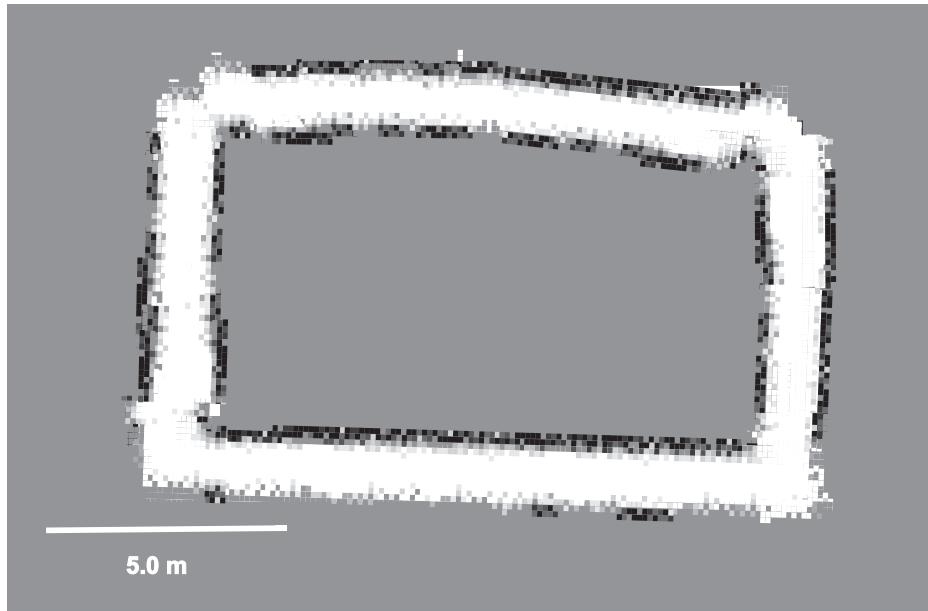


Figura 6.20: Grade de ocupação.

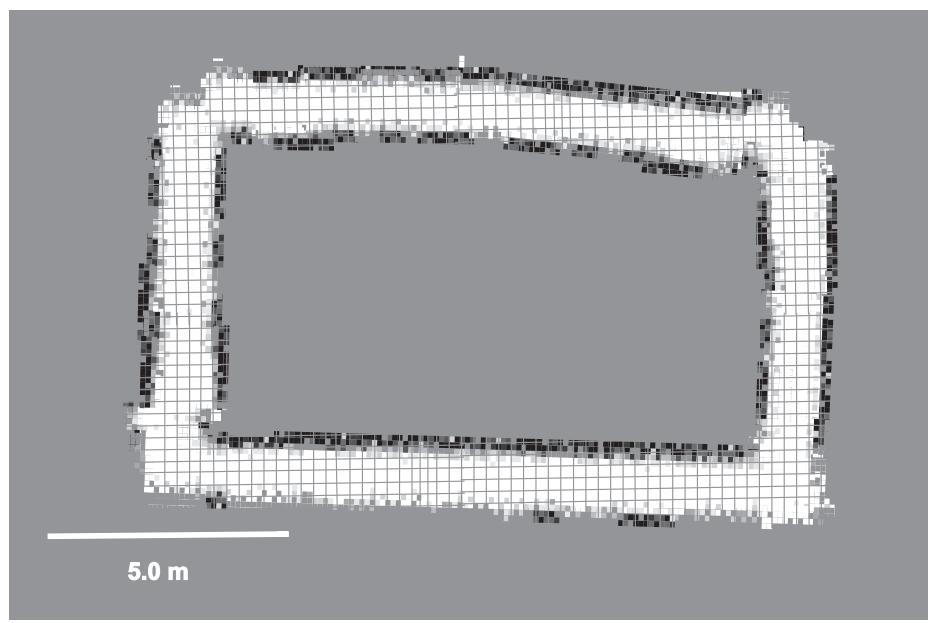


Figura 6.21: Mapa híbrido: Grade de ocupação e Mapa de Características.



---

# Capítulo 7

## Conclusões e Perspectivas

---

O processo de Localização e Mapeamento Simultâneos é um dos problemas fundamentais em tarefas de navegação autônoma de robôs, e por isso vem sendo alvo de extensa pesquisa. Uma solução para esse problema pode permitir o surgimento de robôs verdadeiramente autônomos, capazes de navegar de maneira segura por ambientes desconhecidos e cumprir objetivos sem a necessidade de auxílio externo de espécie alguma.

Com base nisso, esta Tese apresentou uma solução de SLAM adequada para ambientes planos com linhas presentes no chão, de modo a permitir que o robô navegue no ambiente fundindo informações de odometria e de visão monocular. Ressalta-se que a restrição do uso de informação planar e de linhas existentes no piso do ambiente não é tão restritiva pois são comuns em muitos ambientes como casas, museus, hospitais, universidades, shoppings, dentre outros.

O sistema que foi descrito faz inicialmente uma etapa de segmentação objetivando classificar as partes da imagem em “chão” e “não-chão”. Em seguida, o processador de imagem identifica linhas no “chão” e os parâmetros dessas linhas são usados como marcos no SLAM. Por fim, a pose do robô corrigida juntamente com parte “não-chão” da imagem são usadas para construir uma grade de ocupação gerando um mapa final híbrido de grade e características.

A representação híbrida do ambiente possui os benefícios dos mapas de características e dos mapas em grade de ocupação. A primeira representação é de fácil construção, manutenção e fornece boas informações para o cômputo da pose do robô. A segunda, por sua vez, é muito útil para a execução de tarefas de mais alto nível, como planejamento de caminhos e desvio de obstáculos. Assim, acredita-se que uma solução de SLAM usando um mapa de características e que simultaneamente constrói um mapa em grade de ocupação pode ser muito útil em aplicações de robôs autônomos.

Foram apresentadas várias experiências com características distintas para validar o sistema e este se comportou de forma satisfatória em todos os casos, inclusive no tocante ao fechamento de laço. Mostrou-se também uma comparação entre SLAM com linhas e SLAM com pontos e o uso de linhas se comportou melhor.

Em seguida foi apresentada uma proposta para a construções de uma grade de ocupação a partir de visão monocular e usando as informações do SLAM visual. Os resultados experimentais também foram satisfatórios e solidificaram a construção do mapa híbrido.

Em suma, as principais contribuições desse trabalho são:

- (a) sistema de SLAM Visual em tempo real adequado para ambientes planos com linhas no chão utilizando técnicas clássicas da literatura;
- (b) modelagem do sensor ótico de modo a permitir a utilização de parâmetros de retas extraído de imagens diretamente nas equações de Kalman sem etapas intermediárias de cálculo de pose e distância;
- (c) algoritmo para construção de grade de ocupação a partir de visão monocular;
- (d) um sistema de SLAM monocular com representação dual do ambiente em grade de ocupação e mapa de características.

Como trabalhos futuros pretende-se:

- (a) melhorar as propriedades de tempo real do algoritmo de detecção de retas através da adoção de alguma das variantes da transformada de Hough - *Fast Hough Transform*, por exemplo;
- (b) lidar com segmentos de reta com comprimento finito e incorporar as informações do ponto inicial e final do segmento como características a serem utilizadas no SLAM;
- (c) evoluir a proposta do uso de linhas para o espaço 3D usando *Coordenadas Plücker*, por exemplo;
- (d) testar a abordagem utilizando outros filtros estatísticos como por exemplo o *Unscented Kalman filter*;
- (e) aperfeiçoar o algoritmo de segmentação do terreno de forma a torná-lo adaptativo usando *Redes Neurais*, por exemplo;
- (f) formular matematicamente a construção simultânea dos mapas de grade de ocupação e de características no SLAM;
- (g) desenvolver um sistema de localização, mapeamento e planejamento simultâneos (SPLAM);
- (h) incorporar no sistema a representação topológica.

---

# Capítulo 8

## Produção Científica

---

### Periódicos - (em análise)

- André M. Santana & Adelardo A. D. Medeiros, '*Uma Abordagem Baseada em Retas para SLAM em Ambientes Planos Usando Visão Monocular*', Revista: IEEE América Latina. Submissão: Fevereiro de 2010, Status: **Aceito**.
- André M. Santana, Armando S. Sanca, Pablo J. Alsina & Adelardo A. D. Medeiros, '*SLAM monocular usando las líneas de piso*', Revista: Información Tecnológica. Submissão: Dezembro de 2010, Status: **Aceito**.
- André M. Santana & Adelardo A. D. Medeiros, '*Straight-Lines Modelling Using Planar Information For Monocular SLAM*', Revista: Robótica Cambridge. Submissão: Julho de 2010, Status: **Em revisão**.
- André M. Santana & Adelardo A. D. Medeiros, '*Occupancy Grid Maps 2D Using Planar Information and Monocular Vision*', Revista: Image and Vision Computing. Submissão: Fevereiro de 2011, Status: **Em revisão**.

### Capítulos de Livros

- Anderson A. S. Souza, André M. Santana, Luiz M. G. Gonçalves & Adelardo A. D. Medeiros, '*Capítulo 08: Mapeamento de Ambientes*', Livro: Robótica Móvel, Editores:, Editora: Campus/Elsevier, Status: **Em revisão**.
- André M. Santana, Anderson A. S. Souza, Luiz M. G. Gonçalves, Pablo J. Alsina & Adelardo A. D. Medeiros '*Fusion of Odometry and Visual Data to Localizing a Mobile Robot*', Livro: Sensor Fusion and Its Applications, Editor: Ciza Thomas, Editora: Sciyo, ISBN 978-953-307-101-5, pp. 407-422, 10/2010.
- Anderson A. S. Souza, André M. Santana, Luiz M. G. Gonçalves & Adelardo A. D. Medeiros '*Probabilistic Mapping by Fusion of Range-Finders Sensors and Odometry*', Livro: Sensor Fusion and Its Applications, Editor: Ciza Thomas, Editora: Sciyo, ISBN 978-953-307-101-5, pp. 423-442, 10/2010.
- André M. Santana & Adelardo A. D. Medeiros '*Simultaneous Localization and Mapping (SLAM) of a Mobile Robot Based on Fusion of Odometry and Visual Data Using Extended Kalman Filter*' Livro: Contemporary Robotics - Challenges and Solutions, Editor: A. D. Rodic), Editora: Intech, ISBN 978-953-307-038-4, pp. 129-146, 12/2009.

## Congressos Internacionais

- André M. Santana, Anderson A. S. Souza, Ricardo S. Britto, Pablo J. Alsina & Adelardo A. D. Medeiros' *Localization of a Mobile Robot Based on Odometry and Natural Landmarks Using Extended Kalman Filter*', International Conference on Informatics in Control, Automation and Robotics (ICINCO'08), Funchal, Madeira, Portugal, 08/2008.
- Anderson A. S. Souza, André M. Santana, Ricardo S. Britto, Luiz M. G. Gonçalves & Adelardo A. D. Medeiros' *Representation of Odometry Errors on Occupancy Grids*', International Conference on Informatics in Control, Automation and Robotics (ICINCO'08), Funchal, Madeira, Portugal, 08/2008.
- Ricardo S. Britto, André M. Santana, Anderson A. S. Souza, Adelardo A. D. Medeiros, & Pablo J. Alsina' *A Distributed Hardware-Software Architecture to Control an Autonomous Mobile Robot*', International Conference on Informatics in Control, Automation and Robotics (ICINCO'08), Funchal, Madeira, Portugal, 08/2008.
- Kelson, R. T. Aires, André M. Santana & Adelardo A. D. Medeiros ' *Optical Flow Using Color Information: preliminary results*' . Symposium on Applied Computing (SAC'08), Fortaleza, CE, Brasil, 03/2008.

## Congressos Nacionais e Regionais

- André M. Santana & Adelardo A. D. Medeiros ' *Uma abordagem baseada em retas para SLAM em ambientes planos usando visão monocular*' . Congresso Brasileiro de Automática (CBA'10), Bonito, MS, Brasil, 09/2010.
- André M. Santana & Adelardo A. D. Medeiros ' *Monocular SLAM using Floor Lines*' , Latin American Robotics Symposium (LARS'10), São Bernardo do Campo, SP, Brasil, 10/2010.
- André M. Santana, Kelson R. T. Aires & Adelardo A. D. Medeiros ' *Localização Relativa de um Robô Humanóide Utilizando Filtro de Kalman Estendido*' . Escola Regional de Computação Ceará - Maranhão - Piauí (ERCEMAPI'09), Parnaíba, PI, Brasil, 09/2009.
- André M. Santana, Gutemberg S. Santiago & Adelardo A. D. Medeiros ' *Real-Time Visual SLAM Using Pre-Existing Floor Lines as Landmarks and a Single Camera*' . Congresso Brasileiro de Automática (CBA'08), Bonito, MS, Brasil, 09/2010.
- André M. Santana & Adelardo A. D. Medeiros ' *Cooperative Path Planning for a Heterogeneous Robotic System Composed of a Humanoid Robot and a Wheeled Robot*' , Latin American Robotics Symposium (LARS'08), Salvador, BA, Brasil, 10/2008.
- André M. Santana & Adelardo A. D. Medeiros ' *Localização de um Robô Móvel Baseada em Odometria e Marcos Naturais Utilizando Filtro de Kalman Estendido*' . Simpósio Brasileiro de Automação Inteligente (SBAI'07), Florianópolis, SC, Brazil, 10/2007.

---

## Referências Bibliográficas

---

- Agrawal, M., K. Konolige & R. Bolles (2007), ‘Localization and mapping for autonomous navigation in outdoor terrains : A stereo vision approach’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4795-4800 .
- Ahn, S. & K. Chung (2007), ‘Efficient SLAM algorithm with hybrid visual map in an indoor environment’, *Conference on Control, Automation and Systems* .
- Ahn, S. & K. Lee (2007), ‘SLAM with visual plane: Extracting vertical plane by fusing stereo vision and ultrasonic sensor for indoor environment’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4787-4794 .
- Ahn, S., K. Lee, W. Chung & S. Sang-Rok Oh (2007), ‘SLAM with visual plane: Extracting vertical plane by fusing stereo vision and ultrasonic sensor for indoor environment’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Ahn, S., W. K. Chung & S. Oh (2007), ‘Construction of hybrid visual map for indoor SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Aires, K. R. T. & A. A. D. Medeiros (2010), ‘A plane segmentation system based on affine homography and optical flow’, *Conference on Graphics Patterns and Images (SIBGRAPI)* .
- Alsina, Pablo Javier, Luiz Marcos Garcia Gonçalves, Adelaro Adelino Dantas Medeiros, Diogo Pinheiro Fernandes Pedrosa & Frederico Carvalho Vieira (2002), ‘Navegação e controle de robôs móveis’, *Mini Curso - Congresso Brasileiro de Automática* .
- Andrade-Cetto, J. & A. Sanfeliu (2002), ‘Concurrent map building and localization on indoor dynamic environments’, *International Journal on Pattern Recognition and Artificial Intelligence*, Vol. 16, No. 3, pp. 361-374 .
- Anousaki, G. C. & K. J. kopoulos (2007), ‘Simultaneous localization and map building of skid-steered robots’, *IEEE Robotics and Automation Magazine*, pp. 79-89 .
- Armesto, L., G. Ippoliti, S. Longhi & J. Tornero (2008), ‘Probabilistic self-localization and mapping: An asynchronous multirate approach’, *IEEE Robotics and Automation Magazine*, pp.77-88 .
- Ayache, N. & O. Faugeras (1988), ‘Building, registering, and fusing noisy visual maps’, *International Journal on Robotics Research*, Vol. 7, No. 6, pp. 45-65 .

- Baker, S. & Matthews (2001), ‘Equivalence and efficiency of image alignment algorithms’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* .
- Barra, R. J., C. H. Ribeiro & A. H. R. Costa (2009), ‘Fast vertical line correspondence between images for mobile robot localization’, *IFAC Symposium on Robot Control*, pp. 153-158 .
- Bay, H., T. Tuytelaars & L. Van Gool (2006), ‘Object recognition from local scale-invariant features’, *European Conference on Computer Vision* .
- Berger, C. & S. Lacroix (2008), ‘Using planar facets for stereovision SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1606-1611 .
- Bernardes, M. C. & G. A. Borges (2010), ‘3D line estimation for mobile robotics visual servoing’, *Congresso Brasileiro de Automática (CBA)* .
- Blanco, J., J. Antonio, F. Madrigal & J. Gonzalez (2007), ‘A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2061-2067 .
- Blanco, J. & J. Gonzalez (2007), ‘A new method for robust and efficient occupancy grid-map matching’, *Iberian Conference on Pattern Recognition and Image Analysis* .
- Blanco, J., J. Madrigal & J. Gonzalez (2008), ‘A new approach for large-scale localization and mapping: Hybrid metric-topological SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2061-2067 .
- Bonin-Font, F., A. Ortiz & G. Oliver (2008), ‘Visual navigation for mobile robots: A survey’, *Journal of Intelligent and Robotic Systems*, Vol. 53, Issue 3 .
- Borenstein, J. & L. Feng (1996), ‘Measurement and correction of systematic odometry errors in mobile robots’, *IEEE Transactions on Robotics and Automation* .
- Borenstein, J. & Y. Koren (1991), ‘The vector field histogram: fast obstacle avoidance for mobile robots’, *IEEE Journal of Robotics and Automation*, pp. 278-288 .
- Borenstein, J. & Y. Koren (1997), ‘The vector field histogram fast obstacle avoidance for mobile robots’, *IEEE Journal of Robotics and Automation* .
- Bosse, M., R. Rikoski, J. Leonard & S. Teller (2005), ‘Vanishing points and 3D lines from omnidirectional video’, *International Conference Image Processing*, pp.513-516 .
- Botelho, S. S. C., G. Leivas & P. Drews (2009), ‘Odometry and mapping for underwater autonomous vehicles’, *Simpósio Brasileiro de Automação Inteligente (SBAI)* .
- Bradski, G. & A. Kaehler (2008), *Learning OpenCV: Computer Vision with the OpenCV Library*, O'Reilly Media.

- Braillon, C., C. Pradalier, K. Usher, J. Crowley & C. Laugier (2008), ‘Occupancy grids from stereo and optical flow data’, *Experimental Robotics, Springer*, 367-376 .
- Braillon, C., K. Usher, C. Pradalier, J. Crowley & C. Laugier (2006), ‘Fusion of stereo and optical flow data using occupancy grids’, *IEEE Intelligent Transportation Systems Conference*, pp. 1240-1245 .
- Brunskill, E. & N. Roy (2005), ‘SLAM using incremental probabilistic PCA and dimensionality reduction’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Buhmann, J., W. Burgard, A. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos & S Thrun (1995), ‘The mobile robot rhino’, *AI Magazine* .
- Burgard, W., A. Cremers, D. Fox, G. Hofmann, J. Lakemeyer, G. Schulz, W. Steiner & S Thrun (1997), ‘Experiences with an interactive museum tour-guide robot’, *Artificial Intelligence*, pp. 3-55 .
- Burgard, W., C. Stachniss, G. Grisetti, B. Steder, R. Kummerle, C. Dornhege, M. Ruhnke, A. Kleiner & J. Tardos (2009), ‘A comparison of SLAM algorithms based on a graph of relations’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2089-2085 .
- Calway, A. & W. Cuevas (2008), ‘Discovering higher level structure in visual SLAM’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 980-990 .
- Campbell, J., R. Sukhankar & I. Nourbakhsh (2004), ‘Techniques for evaluating optical flow for visual odometry in extreme terrain’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3704-3711 .
- Canas, J. & V. Matellan (2006), ‘Dynamic gridmaps: comparing building techniques’, *Mathware and Soft Computing* .
- Canny, J. (1986), ‘A computational approach to edge detection’, *IEEE Transaction Pattern Analysis and Machine Intelligence*, pp. 679-698 .
- Censi, A., L. Iocchi & G. Grisetti (2005), ‘Scan matching in the hough domain’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Chatila, R. & J. P. Laumond (1985), ‘Position referencing and consistent world modeling for mobile robots’, *IEEE International Conference on Robotics and Automation (ICRA)*. pp. 138-143 .
- Cheein, F., N. Lopez, C. Soria, F. Sciascio, F. Pereira & R. Carelli (2010), ‘SLAM algorithm applied to robotics assistance for navigation in unknown environments’, *Journal of NeuroEngineering and Rehabilitation (JNER)* .

- Chen, H., D. Sun, J. Yang & W. Shang (2008), ‘Orientation correction based monocular SLAM for a mobile robot’, *Int. Conf. on Advanced Intelligent Mechatronics*, pp. 1378-1383 .
- Chen, Z. & J. Samarabandu (2006), ‘A visual SLAM solution based on high level geometry knowledge and kalman filtering’, *Canadian Conference on Electrical and Computer Engineering*, pp. 1283-1286 .
- Cheong, H., S. Park & S. Park (2008), ‘Topological map building and exploration based on concave nodes’, *International Conference on Control, Automation and Systems (ICCAS)*, pp. 1115-1120 .
- Chli, M. & A. J. Davison (2008), ‘Active matching’, *International Conference on Computer Vision (ICCV)*, pp. 72-85 .
- Choi, Y. & S. Oh (2006), ‘Grid-based visual SLAM in complex environment’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Choi, Y. & S. Oh (2007), ‘Map building through pseudo dense scan matching using visual sonar data’, *Springer Science* .
- Choi, Y., T. Lee & S. Oh (2008), ‘A line feature based SLAM with low grade range sensors using geometric constraints and active exploration for mobile robot’, *Autonomous Robot*, Vol. 14, pp.13-27 .
- Choset, H. & D. Fox (2004), ‘The world of mapping’, *Workshop on Review of United States Research in Robotics, National Science Foundation* .
- Choset, H., K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki & S. Thrun (2005), *Principles of Robot Motion*, The MIT Press.
- Civera, J., A. J. Davison & J. M. Montiel (2008), ‘Inverse depth parametrization for monocular SLAM’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 932-945 .
- Clemente, L., A. Davison, I. Reid, J. Neira & J. Tardos (2007), ‘Mapping large loops with a single hand-held camera’, *Robotics: Science and Systems* .
- Comaniciu, D. & P. Meer (1999), ‘Mean shift analysis and applications’, *IEEE International Conference on Computer Vision (ICCV)*, pp.1197-1203 .
- Connette, C., O. Meister, M. Hagele & G. Trommer (2007), ‘Decomposition of line segments into corner and statistical grown line features in an EKF-SLAM framework’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Craig, J. J. (2004), *Introduction to Robotics: Mechanics and Control*, Pearson Education.
- Crowley, J. (1989), ‘World modeling and position estimation for a mobile robot using ultra-sonic ranging’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 674-681 .

- Csorba, M. (1997), Simultaneous Localisation and Map Building, Tese de doutorado, University Oxford.
- Csorba, M. & H. F. Durrant-Whyte (1996), 'A new approach to simultaneous localisation and map building', *SPIE Aerosense, Orlando, FL*.
- Dailey, M. & M. Parnichkun (2005), 'Landmark based simultaneous localization and mapping with stereo vision', *Asian Conference on Industrial Automation and Robotics*.
- Dao, N., B. You, S. Oh & M. Hwangbo (2003), 'Visual self-localization for indoor mobile robots using natural lines', *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Davison, A. (2002), 'SLAM with a single camera', *IEEE International Conference on Robotics and Automation (ICRA)*.
- Davison, A. J (2003), 'Real-time simultaneous localisation and mapping with a single camera', *International Conference on Computer Vision (ICCV)*.
- Davison, A. J. & D. W. Murray (2002), 'Simultaneous localization and map-building using active vision', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 865-880.
- Davison, A. J. & N. Kita (2001), 'Sequential localization and map building for real time computer vision and robotics', *Robotic Autonation System*, pp. 171-183.
- Davison, A. J., W. W. Mayol & D. W. Murray (2002), 'Real-time localisation and mapping with wearable active vision', *International Symposium on Mixed and Augmented Reality*.
- Davison, A. J, Y. G. Cid & N. Kita (2004), 'Real-time 3d SLAM with wide-angle vision', *Symposium on Intelligent Autonomous Vehicles*.
- de Saint Vincent, A. (1987), 'Visual navigation for a mobile robot building a map of the occupied space from sparse 3D stereo data', *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 4, pp. 1429-1435.
- Diosi, A., G. Taylor & L. Kleeman (2005), 'Interactive SLAM using laser and advanced sonar', *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1103-1108.
- Dollner, J. & K. Hinrichs (2000), 'Dynamic 3d map and theier texture', *International Conference on Computer Graphics (ICCG)*.
- Drews, P. & S. S. C. Botelho (2009), 'A visual system for distributed mosaics using AUV fleets', *Sea Technology*, Vol. 50, pp. 1-8.

- Durrant-Whyte, H., D. Rye & E. Nebot (1996), ‘Localisation of automatic guided vehicles’, *Robotics Research: The 7th International Symposium*, (Eds) G. Giralt and G. Hirzinger, Eds. New York: Springer Verlag, pp. 613 - 625 .
- Durrant-Whyte, H. F. (1988), ‘Uncertain geometry in robotics’, *IEEE Transactions on Robotics and Automation*, Vol. 4, No. 1, pp. 23-31 .
- Durrant-Whyte, H. & T. Bailey (2006a), ‘Simultaneous localization and mapping: Part i’, *IEEE Transactions on Robotics and Automation*, pp. 99-108 .
- Durrant-Whyte, H. & T. Bailey (2006b), ‘Simultaneous localization and mapping: Part i’, *IEEE Transactions on Robotics and Automation*, pp. 109-117 .
- Eade, E. & T. Drummond (2006), ‘Edge landmarks in monocular slam’, *British Machine Vision Conference (BMVC)* .
- Eade, E. & T. Drummond (2009), ‘Edge landmarks in monocular slam’, *Image and Vision Computing*, pp.588-596 .
- Elfes, A. (1987), ‘Sonar-based real-world mapping and navigation’, *Journal of Robotics and Automation*, pp. 249-265 .
- Elfes, A. (1989), Occupancy Grids: A Probabilistic Framework for Robot Perception and Navigation, Tese de doutorado, Electrical and Computer Engineering, Carnegie Mellon University.
- Estrada, C., J. Neira & J. D. Tards (2005), ‘Hierarchical SLAM: Realtime accurate mapping of large environments’, *IEEE Transactionson Robotics*, pp. 588-596 .
- Estrada, E., L. Silveira, V. Oliveira & S. S. C. Botelho (2010), ‘Sistema de navegação para um robô de inspeção de linhas subterrâneas de distribuição’, *Congresso Brasileiro de Automática (CBA)* .
- Ethan Eade, E. & T. Drummond (2007), ‘Monocular SLAM as a graph of coalesced observations’, *International Conference on Computer Vision (ICCV)* .
- Fang, F., X. Ma & X. Dai (2006), ‘Mobile robot localization based on improved model matching in hough space’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1541-1546 .
- Feng, H., F. Yongchun, W. Yutao & B. Tao (2008), ‘Practical feature based simultaneous localization and mapping using sonar data’, *Chinese Control Conference (CCC)*, pp. 16-18 .
- Fernandez, D. & A. Price (2004), ‘Visual odometry for an outdoor mobile robot’, *IEEE International Conference on Robotics, Automation and Mechatronics (RAM)*, pp. 816-821 .

- Fleuret, F., J. Berclaz, R. Lengagne & P. Fua (2008), ‘Multicamera people tracking with a probabilistic occupancy map’, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 2, pp. 267-282 .
- Flint, A., C. Mei, I. Reid & D. Murray (2010), ‘Growing semantically meaningful models for visual SLAM’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 467-474 .
- Folkesson, J., P. Jensfelt & H. Christensen (2005), ‘Vision SLAM in the measurement subspace’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Forsyth, D. & J. Ponce (2002), *Computer Vision A Modern Approach*, Prentice Hall.
- Frintrop, S. & P. Jensfelt (2008), ‘Attentional landmarks and active gaze control for visual SLAM’, *IEEE Transaction on Robotics*, Vol. 24, No. 5 .
- Frintrop, S., P. Jensfelt & H. I. Christensen (2006), ‘Attentional landmark selection for visual SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Fu, S., C. Liu, L. Gao & Y. Gai (2007a), ‘SLAM for mobile robots using laser range finder and monocular vision’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Fu, S. & G. Yang (2009), ‘Uncalibrated monocular based SLAM for indoor autonomous mobile robot navigation’, *International Conference on Networking, Sensing and Control*, pp. 663-668 .
- Fu, S., H. Liu, L. Gao & Y. Gai (2007b), ‘SLAM for mobile robots using laser range finder and monocular vision’, *IEEE International Conference on Robotics and Automation (ICRA), Roma, Itália*, pp. 91-96 .
- Fukunaga, K. & L. Hostetler (1975), ‘The estimation of the gradient of a density function, with applications in pattern recognition’, *IEEE Trans. Info. Theory*, Vol. 21, pp. 32-40 .
- Gee, A., D. Chekhlov, A. Calway & W. Mayol-Cuevas (2008), ‘Discovering higher level structure in visual SLAM’, *IEEE Transaction on Robotics*, Vol. 24, No. 5 .
- Gee, A. & W. Mayol-Cuevas (2007), ‘Real-time model-based SLAM using line segments’, *Advances in Visual Computing*, pp.354-363 .
- Georgiades, C., A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L.A. Torres, P. Zhang, M. Buehler, G. Dudek, M. Jenkin & E. Milios (2004), ‘An aquatic walking robot’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .

- Georgiades, C., A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L.A. Torres, P. Zhang, M. Buehler, G. Dudek, M. Jenkin & E. Milios (2005), ‘A visually guided swimming robot’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Gil, A., O. Mozos, M. Ballesta & O. Reinoso (2009), ‘A comparative evaluation of interest point detectors and local descriptors for visual SLAM’, *Machine Vision and Applications*, Vol. 21, No. 6, pp. 905-920 .
- Goldstein, M., F. Pin, G. Saussure & C. Weisbin (1987), ‘3D world modeling based on combinatorial geometry for autonomous robot navigation’, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 4, pp. 727-733 .
- Gonçalves, E., E. Estrada, R. Gules, S. S. C. Botelho, L. Silveira & V. Oliveira (2009), ‘Uma arquitetura para navegação autônoma de robôs reais e simulados para inspeção de linhas de distribuição de energia subterrâneas’, *Simpósio Brasileiro de Automação Inteligente (SBAI)* .
- Gonzalez, R. & R. Woods (2007), *Digital Image Processing*, Prentice Hall.
- Guerrero, J., R. Martinez-Cantin & C. Sagües (2005), ‘Visual map-less navigation based on homographies’, *Journal of Robotic Systems*, Vol. 22 , No. 10 .
- Guivant, J. E. & E. M. Nebot (2001), ‘Optimization of the simultaneous localization and map-building algorithm for real-time implementation’, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 242 - 257 .
- Guizilini, V. C. & J. Okamoto Junior (2009), ‘Solving the online SLAM problem with an omnidirectional vision system’, *Lecture Notes in Computer Science (LNCS)*, Vol. 5506, pp. 1110-1117 .
- Hafez, A., S. Bhuvanagiri, M. Krishna & C. Jawahar (2008), ‘On-line convex optimization based solution for mapping in VSLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Hamming, R. W. (1950), ‘Error detecting and error correcting codes’, *Bell System Technical Journal*, Vol. 02, pp. 147-160 .
- Han, C., Z. Xiang, J. Liu & E. Wu (2007), ‘Stereo vision based slam in outdoor environments’, *Int. Conf. on Robotics and Biomimetics*, pp. 15-18 .
- Harris, C. & M. Stephens (1998), ‘A combined corner and edge detector’, *Alvey Vision Conference* .
- Hartley, R. & A. Zisserman (2004), *Multiple View Geometry in Computer Vision*, Cambridge University Press.
- Hata, A. & D. Wolf (2010a), ‘Mapeamento e classificação de terrenos utilizando aprendizado supervisionado’, *XVII Congresso Brasileiro de Automática, Mato Grosso* .

- Hata, A. & D. Wolf (2010b), ‘Mapeamento e classificação de terrenos utilizando aprendizado supervisionado’, *Congresso Brasileiro de Automática* .
- Hata, A. Y., D. F. Wolf & F. S. Ososrio (2009), ‘Terrain mapping and classification in outdoor environments using neural networks’, *International Journal of u- and e-Service, Science and Technology*, Vol. 2, pp. 51-61 .
- Herath, D., K. Kodagoda & G. Dissanayake (2006), ‘Modeling errors in small baseline stereo for SLAM’, *International Conference on Control, Automation, Robotics and Vision (ICCARV)* .
- Horn, B. K. P. & B. G. Schunck (1981), ‘Determining optical flow’, *Artifitial Intelligence*, pp. 185-203 .
- Horswill, I. (1993), ‘Polly: a vision-based artificial agent’, *AAAI Conference on Artificial Intelligence* .
- Hough, P. (1962), ‘Method and means for recognizing complex patterns’, *US Patten 3069654, December 18* .
- Howard, A. & L. Kitchen (1997), ‘Fast visual mapping for mobile robot navigation’, *IEEE International Conference on Intelligent Processing Systems*, pp. 1251-1255 .
- Huang, G. P., A. I. Mourikis & S. I. Roumeliotis (2008), ‘Analysis and improvement of the consistency of extended kalman filter based SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 473-479 .
- Huang, L., B. He & T. Zhang (2010), ‘An autonomous navigation algorithm for under-water vehicles based on inertial measurement units and sonar’, *International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 311-314 .
- Huang, S. & G. Dissanayake (2007), ‘Convergence and consistency analysis for extended kalman filter based SLAM’, *IEEE Transaction on Robotics*, Vol. 23, No. 5, pp. 1036-1049 .
- Huang, S., Z. Wang & G. Dissanayake (2008), ‘Exact state and covariance sub-matrix recovery for submap based sparse eif SLAM algorithm’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1868-1873 .
- Jeong, W. & K. Lee (2006), ‘Visual SLAM with line and corner features’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.2570-2575 .
- Jing Wu, J. & H. Zhang (2007), ‘Camera sensor model for visual SLAM’, *IEEE Canadian Conference on Computer and Robot Vision* .
- Joan Sola, J., A. Monin, M. Devy & T. Vidal-Calleja (2008), ‘Fusing monocular information in multicamera SLAM’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 958-968 .

- Jones, S.D., C. Andresen & J.L. Crowley (1997), ‘Appearance based processes for visual navigation’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 551-557 .
- Julier, S. J., J. K. Uhlmann & H. F. Durrant-Whyte (1995), ‘A new approach for filtering nonlinear systems’, *American Control Conference*, pp. 1628-1632 .
- Jung, I.K. (2004), Simultaneous localization and mapping in 3D environments with stereovision, Tese de doutorado, Institut National Polytechnique - Toulouse - France.
- Kaess, M. & F. Dellaert (2006), ‘Visual SLAM with a multi-camera rig’, *Technical Report, Georgia Institute of Technology* .
- Kalman, R. E. (1960), ‘A new approach to linear filtering and predictive problems’, *Transactions of the Journal of Basic Engineering*, pp. 35-45 .
- Kim, A. & R. Eusticey (2009), ‘Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Kim, S. & S. Oh (2008), ‘SLAM in indoor environments using omni-directional vertical and horizontal line features’, *Journal of Intelligent Robotic System*, pp.31-43 .
- Kitanov, A., S. Bisevac & I. Petrovic (2007), ‘Mobile robot self-localization in complex indoor environments using monocular vision and 3D model’, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics* .
- Klippenstein, J. & H. Zhang (2007), ‘Quantitative evaluation of feature extractors for visual SLAM’, *Int. Conf. on Computer and Robot Vision* .
- Konolige, K. (1997), ‘Improved occupancy grids for map building’, *Autonomous Robots Vol.4*, pp. 351-367 .
- Kootstra, G. & L. Schomaker (2009), ‘Using symmetrical regions of interest to improve visual SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 930-935 .
- Kriegman, D., E. Triendl & T. Binfold (1987), ‘Generation of object descriptions from range data using feature extraction by demands’, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 4, pp. 402-408 .
- Kuipers, B. & Y. Byun (1988), ‘A robust, qualitative method for spatial learning in unknown environments’, *National Conference on Artificial Intelligence* , pp. 1-12 .
- Kuipers, B. & Y. Byun (1991), ‘A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations’, *Robotics and Autonomous Systems*, Vol. 8, pp. 47-63 .

- Lacroix, S., T. Lemaire & C. Berger (2008), 'More vision for SLAM', *Unifying Perspectives in Computational and Robot Vision*, Vol. 8, pp. 129-147 .
- Latombe, J. (1991), *Robot Motion Planning*, Kluwer Academic Publishers.
- Lee, D., M. Hebert & T. Kanade (2009), 'Geometric reasoning for single image structure recovery', *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 467-474 .
- Lemaire, T., C. Berger, I. Jung & S. Lacroix (2007), 'Vision-based SLAM: Stereo and monocular approaches', *International Journal of Computer Vision*, Vol. 74, Issue 3, pp. 2791-2796 .
- Lemaire, T. & S. Lacroix (2007), 'Monocular-vision based SLAM using line segments', *IEEE International Conference on Robotics and Automation (ICRA)* .
- Leonard, J. J. & H. F. Durrant-Whyte (1991), 'Mobile robot localization by tracking geometric beacons', *IEEE Transactions on Robotics and Automation*, Vol. 1, No. 3, pp. 376-382 .
- Leonard, J. J. & H. F. Durrant-Whyte (1992), 'Directed sonar navigation', *Norwell, MA: Kluwer* .
- Leonard, J. J. & H. J. S. Feder (2000), 'A computational efficient method for large-scale concurrent mapping and localisation', *The Ninth International Symposium*, pp. 169 -176 .
- Li, C., Y. Huang, Y. Kang & J. Yuan (2008), 'Monocular SLAM using vertical straight lines with inverse-depth representation', *World Congress on Intelligent Control and Automation*, pp. 3015-3020 .
- Lowe, D. (1999), 'Object recognition from local scale-invariant features', *International Conference on Computer Vision (ICCV)* .
- Lucas, B. D. & T. Kanade (1981), 'An iterative image registration technique with an application to stereo vision', *Proc. of DARPA Image Understanding Workshop*, pp. 121-130 .
- Manly, B. & L. Kelly (2005), *Multivariate Statistical Methods: A Primer*, Chapman-Hall (ISBN 0-412-28620-3).
- Mansinghka, V. K. (2004), 'Towards visual SLAM in dynamic environments', *Springer* .
- Marks, T., A. Howard, M. Bajracharya, G. Cottrell & L. Matthies (2008), 'Gamma-SLAM: Using stereo vision and variance grid maps for SLAM in unstructured environments', *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3717-3724 .

- Martin, M. C. (2002), ‘Genetic programming for robot vision’, *International Conference on the Simulation of Adaptive Behavior*, pp. 256-265 .
- Matsumoto, Y., M. Inaba & H. Inoue (1996), ‘Visual navigation using view sequenced route representation’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 83-88 .
- Medeiros, A. A. D. (1998), ‘A survey of control architectures for autonomous mobile robots’, *Journal of the Brazilian Computer Society* .
- Merat, F. & F. Wu (1987), ‘Generation of object descriptions from range data using feature extraction by demands’, *IEEE International Conference on Robotics and Automation (ICRA)*, Vol. 4, pp. 941-946 .
- Merhy, B., P. Payeur & E. Petriu (2008), ‘Application of segmented 2-d probabilistic occupancy maps for robot sensing and navigation’, *IEEE Transaction on Instrumentation and Measurement*, Vol. 57, No. 12, pp. 2827-2837 .
- Milford, M. J. & G. F. Wyeth (2008), ‘Single camera vision-only SLAM on a suburban road network’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Mitsou, N. & C. Tzafestas (2007), ‘Temporal occupancy grid for mobile robot dynamic environment mapping’, *Mediterranea Conference on Control and Automation* .
- Montemerlo, M. F., S. Thrun, D. Koller & B. Wegbreit (2002), ‘Fastslam: A factored solution to the simultaneous localization and mapping problem’, *National Conference on Artificial Intelligence*, pp. 593-598 .
- Montemerlo, M. F., S. Thrun, D. Koller & B. Wegbreit (2003), ‘Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges’, *International Conference on Artificial Intelligence (ICAI)*, pp. 1151-1156 .
- Moravec, H. (1996), ‘Robot spatial perception by stereoscopic vision and 3d evidence grids’, *CMU Robotics Institute Technical Report,Daimler Benz Research* .
- Moreira, M., H. Machado, C. Mendonca & G. Pereira (2007), ‘2d visual odometry method for global positioning measurement’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2468-2473 .
- Navarro, D., G. Benet & M. Martínez (2007), ‘Line based robot localization using a rotary sonar’, *Emerging Technologies and Factory Automation (ETFA)* .
- Nebot, E. (2002), ‘Simultaneous localization and mapping - summer school’, *Academic Press* .

- Neira, J. & J. Tardos (2001), ‘Data association in stochastic mapping using the joint compatibility test’, *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 6, pp. 890-897 .
- Neubert, P., P. Protzel, T. Vidal-Calleja & S. Lacroix (2008), ‘A fast visual line segment tracker’, *Emerging Technologies and Factory Automation (ETFA)* .
- Nguyen, V., A. Harati, A. Martinelli, R. Siegwart & N. Tomatis (2006), ‘Orthogonal SLAM: a step toward lightweight indoor autonomous navigation’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Nieto, J., J. Guivant & E. Nebot (2005), ‘The hybrid metric maps HYMMs: A novel map representation for denseSLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 391-396 .
- Nourani-Vatani, N., J. Roberts & M. Srinivasan (2008), ‘Imu aided 3d visual odometry for car-like vehicles’, *Australasian Conference on Robotics and Automation (ACRA)* .
- Oriolo, G., G. Ulivi & M. Vendittelli (1997), ‘Fuzzy maps: a new tool for mobile robot perception and planning’, *Robotics Systems*, Vol. 14, No. 3, pp. 179-197 .
- Oskiper, T., Z. Zhu, S. Samarasakera & R. Kumar (2007), ‘Visual odometry system using multiple stereo cameras and inertial measurement unit’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* .
- Ouellette, R. & K. Hirasawa (2007), ‘A comparison of SLAM implementations for indoor mobile robots’, *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1479 - 1484 .
- Paz, L. M., J. D. Tardos & J. Neira (2008), ‘Divide and conquer: Ekf SLAM in o(n)’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 1107-1120 .
- Paz, L. M., P. Jensfelt, J. D. Tardos & J. Neira (2007), ‘Ekf SLAM updates in o(n) with divide and conquer SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1657 -1663 .
- Paz, L., P. Pinies, J. Tardos & J. Neira (2008), ‘Large-scale 6-dof slam with stereo-in-hand’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 946-957 .
- Pink, O. (2008), ‘Visual map matching and localization using a global feature map’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* .
- Renken, W. D. (1993), ‘Concurrent localization and map building for mobile robots using ultrasonic sensors’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .

- Rituerto, A., L. Luis Puig & J. Guerrero (2010), ‘Visual slam with an omnidirectional camera’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 348-351 .
- Rocha, R. (2006), Building Volumetric Maps whit Cooperative Mobile Robots and Useful Information Sharing: A Distributed Control Approach based on Entropy, Tese de doutorado, PhD Thesis, Faculdade de Engenharia da Universidade do Porto, Portugal.
- Sabo, R. (2002), ‘Incremental map building using an occupancy grid for an autonomous monocular robot’, *International Conference on Control, Automation, Robotics And Vision*, pp. 613-618 .
- Santana, A. & A. Medeiros (2010), ‘Monocular-SLAM using floor lines’, *IEEE Latin American Robotic Symposium (JRI-LARS)* .
- Santos-Victor, J.and Sandini, G.and Curotto F. & S. Garibaldi (1993), ‘Divergence stereo for robot navigation: learning from bees’, *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)* .
- Scaramuzza, D. & R. Siegwart (2008), ‘Appearance-guided monocular omnidirectional visual odometry for outdoor ground vehicles’, *IEEE Transaction on Robotics*, Vol. 24, No. 5, pp. 1015-1025 .
- Schleicher, D., L. Bergasa, M. Ocaña, R. Barea & M. López (2009), ‘Real-time hierarchical outdoor slam based on stereovision and gps fusion’, *IEEE Transaction on Intelligent Transportation Systems*, Vol. 10, No 3, pp. 3717-3724 .
- Se, S., D. Lowe & J. Little (2001), ‘Vision based mobile robot localizations and mapping using scale invariant features’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2051-2058 .
- Selvatici, A., F. Dellaert & A. H. R. Costa (2009), ‘Building object-based maps from visual input’, *Simpósio Brasileiro de Automação Inteligente (SBAI)* .
- Servant, F., P. Houlier & E. Marchand (2010), ‘Improving monocular plane-based SLAM with inertial measures’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Silveira, G. & P. Rives (2009), ‘Registro direto de imagens para SLAM visual’, *Simpósio Brasileiro de Automação Inteligente (SBAI)* .
- Sim, R. & G. Dudek (2008), ‘Effective exploration strategies for the construction of visual maps’, *IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pp.69-76 .
- Skrzypczynski, P. (2009), ‘Simultaneous localizations and mapping: A feature based probabilistic approach’, *Journal of Applied Mathematics and Computer Science*, Vol. 19, No. 4, pp. 575-588 .

- Smith, P., I. Reid & A. Davison (2006), ‘Real-time monocular SLAM with straight lines’, *British Machine Vision Conference (BMVC)* .
- Smith, R., M. Self & P. Cheeseman (1987), ‘Estimating uncertain spatial relationships in robotics’, *IEEE Transactions on Robotics and Automation*, Vol. 5, pp. 850-850 .
- Smith, R. & P. Cheesman (1987), ‘On the representation of spatial uncertainty’, *International Journal of Robotics Research (IJRR)* , Vol. 5, No. 4, pp. 56-68 .
- Sola, J., A. Monin & M. Devy (2007), ‘Bicamslam: Two times mono is more than stereo’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4795-4800 .
- Sola, J., T. Vidal-Calleja & M. Devy (2009), ‘Undelayed initialization of line segments in monocular SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Stuckler, J. & S. Behnke (2008), ‘Orthogonal wall correction for visual motion estimation’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Tardos, J., J. Neira, P. Newman & J. Leonard (2002), ‘Robust mapping and localization in indoor environments using sonar data’, *International Journal of Robotics Research*, Vol. 21, No. 4, pp.311-330 .
- Taylor, T., W. Boles & S. Geva (2007), ‘Map building using cheap digital cameras’, *Biennial Conference of the Australian Pattern Recognition* .
- Thrun, S. (1998), ‘Learning metric-topological maps for indoor mobile robot navigation’, *Artificial Intelligence*, Vol. 99, Issue 1 , pp. 21-71 .
- Thrun, S. (2002), ‘Robotic mapping: A survey’, *Exploring Artificial Intelligence in the New Millennium. Morgan Kaufmann* .
- Thrun, S., D. Fox & W. Burgard (1998), ‘probabilistic approach to concurrent mapping and localization for mobile robots’, *Sach. Learning*, vol. 31, no. 1, pp. 29-53 .
- Thrun, S., W. Burgard & Fox. D. (2005), *Probabilistic Robotics*, MIT Press.
- Thrun, S. & Y. Liu (2003), ‘Multi-robot SLAM with eparse extended information filters’, *International Symposium of Robotics Research*, pp. 254-266 .
- Ukrainitz, Y. & B. Sarel (2004), ‘Mean shift: Theory and applications’, XXXX .
- Vidal-Calleja, T., J. Andrade-Cetto, A. J. Davison & D. W. Murray (2006), ‘Active control for single camera SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)* .
- Walter, M., R. Eustice & J. Leonard (2005), ‘A provably consistent method for imposing exact sparsity in feature based SLAM information filters’, *Journal of Robotics Research*, pp. 214-234 .

- Walter, M., R. Eustice & J. Leonard (2007), ‘Exactly sparse extended information filters for feature based SLAM’, *Journal of Robotics Research*, Vol. 26, No. 4, pp. 335-359 .
- Wan, E. & R. Merwe (2001), *Kalman Filtering and Neural Networks*, Wiley, ISBN 978-0-471-36998-1.
- Wang, Y., D. Hung & S. Cheng (2010), ‘Image feature tracker for SLAM with monocular vision’, *SICE Annual Conference*, pp. 2300-2307 .
- Welle, J., D. Schulz, T. Bachran & A. Cremers (2010), ‘Optimization techniques for laser-based 3D particle filter SLAM’, *IEEE International Conference on Robotics and Automation (ICRA)*, Alaska, USA, pp. 3525-3530 .
- Werneck, N. L. & A. H. R. Costa (2010a), ‘Monocular visual mapping with the fast hough transform’, *Workshop de Visão Computacional*, pp. 279-284 .
- Werneck, N. L. & A. H. R. Costa (2010b), ‘SLAM monocular com reconstrução de planos para ambientes internos’, *Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence* .
- Williams, B. & I. Reid (2010), ‘On combining visual SLAM and visual odometry’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3525-3530 .
- Williams, B., M. Cummins, J. Neira, P. Newman, I. Reid & J. Tardos (2008), ‘An image to map loop closing method for monocular SLAM’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1378-1383 .
- Wolf, D. F, E. V. Simões, F. S. Osório & O. Trindade Junior (2009), ‘Robótica móvel inteligente: Da simulação às aplicações no mundo real’, *Mini-Curso: Jornada de Atualização em Informática (JAI)*, Congresso da SBC. .
- Wu, E., L. Zhao, Y. Guo, W. Zhou & Q. Wang (2010), ‘Monocular vision SLAM based on key feature points selection’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1741-1745. .
- Wu, M., F. Huang, L. Wang & J. Sun (2009), ‘Cooperative multi-robot monocular-SLAM using salient landmarks’, *International Asia Conference on Informatics in Control, Automation and Robotics*, pp. 151-155 .
- Wurm, K., C. Stachniss, G. Grisetti & W. Burg (2007), ‘Improved simultaneous localization and mapping using a dual representation of the environment’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Yap, T. N. & C. R. Shelton (2009), ‘SLAM in large indoor environments with low-cost, noisy, and sparse sonars’, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12-17 .

- Yguel, M., O. Aycard & C. Laugier (2006a), ‘Efficient gpu-based construction of occupancy grids using several laser range-inders’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Yguel, M., O. Aycard & C. Laugier (2006b), *Wavelet Occupancy Grids: a Method for Compact Map Building*, P. Corke Fields and S. Sukkarieh (Eds.): Field and Services Robotics, 219-230, Springer-Verlag Berlin Heidelberg.
- Yguel, M., O. Aycard & Laugier. C. (2006), ‘Wavelet occupancy grids: a method for compact map building’, *P. Corke Fields and S. Sukkarieh (Eds.): Field and Services Robotics, 219-230, Springer-Verlag Berlin Heidelberg* .
- Yizong, C. (1995), ‘Mean shift, mode seeking, and clustering’, *IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 8, pp.790-799* .
- Zhang, J. & D. Song (2009), ‘On the error analysis of vertical line pair-based monocular visual odometry in urban area’, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* .
- Zhang, J. & D. Song (2010), ‘Error aware monocular visual odometry using vertical line pairs for small robots in urban areas’, *Special Track on Physically Grounded AI, AAAI Conference on Artificial Intelligence* .



---

## Apêndice A

# Modelagem do Sensor

---

Considere  $\mathbf{A}_{3 \times 3}$  uma matriz de Homografia e  $\mathbf{C}_{3 \times 3}$  a Matriz de Cofatores de  $\mathbf{A}$ , onde  $C_{ij} = (-1)^{i+j} \cdot \det(\mathbf{A})$  eliminando a linha  $i$  e a coluna  $j$ .

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad e \quad \mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{pmatrix} \quad (\text{A.1})$$

Sabe-se que um ponto  $(x^I, y^I, 1)$  pode ser mapeado em outro plano utilizando a relação expressa na equação (A.2). O fator de escala  $s$  é obtido fazendo  $a_{33} = 1$ .

$$s \cdot \begin{pmatrix} x^I \\ y^I \\ 1 \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (\text{A.2})$$

Usando a equação paramétrica da reta, um ponto  $(x, y) \in \text{reta}(\rho, \alpha)$  pode ser definido pela equação (A.3). O parâmetro  $(\rho)$  representa o comprimento do vetor que passa pela origem e é perpendicular à reta e  $(\alpha)$  é o ângulo que esse vetor forma com o eixo X.

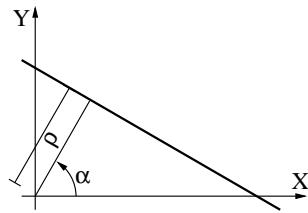


Figura A.1: Representação normal da reta.

$$\begin{cases} x = \rho \cdot \cos(\alpha) - \lambda \cdot \sin(\alpha) \\ y = \rho \cdot \sin(\alpha) + \lambda \cdot \cos(\alpha) \end{cases} \quad (\text{A.3})$$

Expandindo a equação (A.2) e usando a equação (A.3), nota-se que  $(x^I, y^I)$  é dado por:

$$x^I = \frac{K_x \cdot \lambda + C_x}{K_t \cdot C_t} \quad e \quad y^I = \frac{K_y \cdot \lambda + C_y}{K_t \cdot C_t} \quad (\text{A.4})$$

onde,

- $K_x = a_{12} \cdot \cos(\alpha) - a_{11} \cdot \sin(\alpha)$  e  $C_x = a_{11} \cdot \rho \cdot \cos(\alpha) + a_{12} \cdot \rho \cdot \sin(\alpha) + a_{13}$
- $K_y = a_{22} \cdot \cos(\alpha) - a_{21} \cdot \sin(\alpha)$  e  $C_y = a_{21} \cdot \rho \cdot \cos(\alpha) + a_{22} \cdot \rho \cdot \sin(\alpha) + a_{23}$
- $K_t = a_{32} \cdot \cos(\alpha) - a_{31} \cdot \sin(\alpha)$  e  $C_t = a_{31} \cdot \rho \cdot \cos(\alpha) + a_{32} \cdot \rho \cdot \sin(\alpha) + a_{33}$

Efetuando-se a divisão dos polinômios da equação (A.4), tem-se que:

$$\begin{cases} x^I = \frac{K_x}{K_t} + \frac{K_t \cdot C_x - K_x \cdot C_t}{K_t} \cdot \frac{1}{K_t \cdot \lambda + C_t} \\ y^I = \frac{K_y}{K_t} + \frac{K_t \cdot C_y - K_y \cdot C_t}{K_t} \cdot \frac{1}{K_t \cdot \lambda + C_t} \end{cases} \quad (\text{A.5})$$

faz-se, então:

$$t + \phi = \frac{\beta}{K_t \cdot \lambda + C_t} \quad (\text{A.6})$$

e obtém-se nova representação para  $(x^I, y^I)$  como mostra a equação (A.7):

$$\begin{cases} x^I = \frac{K_x}{K_t} + \phi \cdot \frac{K_t \cdot C_x - K_x \cdot C_t}{K_t \cdot \beta} + \left( \frac{K_t \cdot C_x - K_x \cdot C_t}{K_t \cdot \beta} \right) \cdot t \\ y^I = \frac{K_y}{K_t} + \phi \cdot \frac{K_t \cdot C_y - K_y \cdot C_t}{K_t \cdot \beta} + \left( \frac{K_t \cdot C_y - K_y \cdot C_t}{K_t \cdot \beta} \right) \cdot t \end{cases} \quad (\text{A.7})$$

Faz-se uma analogia com o sistema (A.3) e percebe-se que:

$$\begin{cases} \rho^* \cdot \cos(\alpha^*) = \frac{K_x}{K_t} + \phi \cdot \frac{K_t \cdot C_x - K_x \cdot C_t}{K_t \cdot \beta} \\ -\sin(\alpha^*) = \left( \frac{K_t \cdot C_x - K_x \cdot C_t}{K_t \cdot \beta} \right) \\ \rho^* \cdot \sin(\alpha^*) = \frac{K_y}{K_t} + \phi \cdot \frac{K_t \cdot C_y - K_y \cdot C_t}{K_t \cdot \beta} \\ \cos(\alpha^*) = \left( \frac{K_t \cdot C_y - K_y \cdot C_t}{K_t \cdot \beta} \right) \end{cases} \quad (\text{A.8})$$

O termo  $\beta$  é calculado usando a relação trigonométrica  $\sin^2(x) + \cos^2(x) = 1$ . Assim, ele pode ser expresso pela equação (A.9):

$$\beta = \frac{\sqrt{(K_x \cdot C_t - K_t \cdot C_x)^2 + (K_t \cdot C_y - K_y \cdot C_t)^2}}{K_t} \quad (\text{A.9})$$

Calcula-se  $\sin(\alpha^*)$  e  $\cos(\alpha^*)$  usando a equação (A.9) e faz-se a substituição no sistema (A.8) para obter o sistema (A.10).

$$\begin{cases} \rho^* \cdot \cos(\alpha^*) = \frac{K_x}{K_t} - \phi \cdot \sin(\alpha^*) \\ \rho^* \cdot \sin(\alpha^*) = \frac{K_y}{K_t} + \phi \cdot \cos(\alpha^*) \end{cases} \quad (\text{A.10})$$

A partir do sistema (A.10), expressa-se ( $\phi$ ) pela equação (A.11).

$$\phi = \frac{1}{\cos(\alpha^*)} \cdot \left( \rho^* \cdot \sin(\alpha^*) - \frac{K_y}{K_t} \right) \quad (\text{A.11})$$

Por fim, substituindo a equação (A.11) no sistema (A.10) define-se ( $\alpha^I$ ) e ( $\rho^I$ ):

$$\begin{cases} \alpha^I = \tan^{-1} \left( \frac{K_x \cdot C_t - K_t \cdot C_x}{K_t \cdot C_y - K_y \cdot C_y} \right) \\ \rho^I = \frac{K_x \cdot C_y - K_y \cdot C_x}{\sqrt{(K_x \cdot C_t - K_t \cdot C_x)^2 + (K_t \cdot C_y - K_y \cdot C_t)^2}} \end{cases} \quad (\text{A.12})$$

onde,

- $K_x \cdot C_t - K_t \cdot C_x = \rho \cdot C_{23} - \sin(\alpha) \cdot C_{22} - \cos(\alpha) \cdot C_{21}$
- $K_t \cdot C_y - K_y \cdot C_t = \rho \cdot C_{13} - \sin(\alpha) \cdot C_{12} - \cos(\alpha) \cdot C_{11}$
- $K_x \cdot C_y - K_y \cdot C_x = -\rho \cdot C_{33} + \sin(\alpha) \cdot C_{32} + \cos(\alpha) \cdot C_{31}$

ou,

$$\begin{cases} \rho^I = \frac{-\rho \cdot C_{33} + \sin(\alpha) \cdot C_{32} + \cos(\alpha) \cdot C_{31}}{\sqrt{(\rho \cdot C_{23} - \sin(\alpha) \cdot C_{22} - \cos(\alpha) \cdot C_{21})^2 + (\rho \cdot C_{13} - \sin(\alpha) \cdot C_{12} - \cos(\alpha) \cdot C_{11})^2}} \\ \alpha^I = \tan^{-1} \left( \frac{\rho \cdot C_{23} - \sin(\alpha) \cdot C_{22} - \cos(\alpha) \cdot C_{21}}{\rho \cdot C_{13} - \sin(\alpha) \cdot C_{12} - \cos(\alpha) \cdot C_{11}} \right) \end{cases} \quad (\text{A.13})$$

Sabendo que a câmera encontra-se acoplada à estrutura do robô, faz-se necessário expressar os parâmetros da reta em função da pose do robô. Para tal, definiu-se um sistema de coordenadas fixo ( $F$ ) e sistema de coordenadas móvel ( $M$ ), anexado ao robô, ambos ilustrados na Figura A.2.

A origem do sistema móvel tem coordenadas  $(x_M^F, y_M^F)$  no sistema fixo.  $\theta_M^F$  representa a rotação do sistema móvel com respeito ao sistema fixo. Deve-se notar que existe uma relação estreita entre essas variáveis  $(X_M^F, Y_M^F, \theta_M^F)$  e a pose do robô  $(x_t, y_t, \theta_t)$ , que é dada pela equação (A.14).

$$x_t = x_M^F \quad y_t = y_M^F \quad \theta_t = \theta_M^F + \pi/2 \quad (\text{A.14})$$

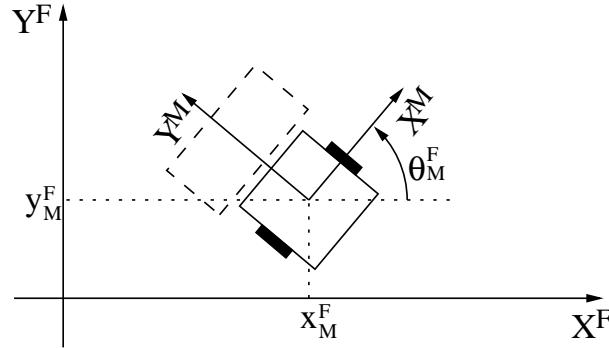


Figura A.2: Sistemas de coordenadas Fixo ( $F$ ) e Móvel ( $M$ ).

Usou-se a relação entre as coordenadas dos sistemas ( $M$ ) e ( $F$ ) (sistema A.15) e a equação  $\rho = x \cos(\alpha) + y \sin(\alpha)$  em ambos os sistemas de coordenada (equações A.16 e A.17).

$$\begin{cases} x^F = \cos(\theta_M^F)x^M - \sin(\theta_M^F)y^M + x_M^F \\ y^F = \sin(\theta_M^F)x^M + \cos(\theta_M^F)y^M + y_M^F \end{cases} \quad (\text{A.15})$$

$${}^i\rho^F = x^F \cos({}^i\alpha^F) + y^F \sin({}^i\alpha^F) \quad (\text{A.16})$$

$$\rho^M = x^M \cos(\alpha^M) + y^M \sin(\alpha^M) \quad (\text{A.17})$$

Substituindo o sistema (A.15) na equação (A.16), fazendo as equivalências com a equação (A.17) e substituindo algumas variáveis usando as equações (A.14), obtém-se os sistemas (A.18) e (A.19).

$$\begin{cases} \rho^M = {}^i\rho^F - x_t \cos({}^i\alpha^F) - y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t + \pi/2 \end{cases} \quad (\text{A.18})$$

$$\begin{cases} \rho^M = -{}^i\rho^F + x_t \cos({}^i\alpha^F) + y_t \sin({}^i\alpha^F) \\ \alpha^M = {}^i\alpha^F - \theta_t - \pi/2 \end{cases} \quad (\text{A.19})$$

Uma vez encontrada a relação entre  $\rho$  e  $\alpha$  e a pose do robô, o passo seguinte é substituir o sistema (A.18) ou o sistema (A.19) nas equações do sistema (A.12) a fim de mapear os parâmetros da reta diretamente da imagem para o mundo (sistema A.20).

$$\begin{cases} \alpha^I = \tan^{-1}\left(\frac{K_x^* \cdot C_t^* - K_t^* \cdot C_x^*}{K_t^* \cdot C_y^* - K_y^* \cdot C_y^*}\right) \\ \rho^I = \frac{K_x^* \cdot C_y^* - K_y^* \cdot C_x^*}{\sqrt{(K_x^* \cdot C_t^* - K_t^* \cdot C_x^*)^2 + (K_t^* \cdot C_y^* - K_y^* \cdot C_t^*)^2}} \end{cases} \quad (\text{A.20})$$

onde,

- $K_x^* \cdot C_t^* - K_t^* \cdot C_x^* = \rho^M \cdot C_{23} - \sin(\alpha^M) \cdot C_{22} - \cos(\alpha^M) \cdot C_{21}$
- $K_t^* \cdot C_y^* - K_y^* \cdot C_t^* = \rho^M \cdot C_{13} - \sin(\alpha^M) \cdot C_{12} - \cos(\alpha^M) \cdot C_{11}$
- $K_x^* \cdot C_y^* - K_y^* \cdot C_x^* = -\rho^M \cdot C_{33} + \sin(\alpha^M) \cdot C_{32} + \cos(\alpha^M) \cdot C_{31}$

ou,

$$\begin{cases} \rho^I = \frac{-\rho^M \cdot C_{33} + \sin(\alpha^M) \cdot C_{32} + \cos(\alpha^M) \cdot C_{31}}{\sqrt{(\rho^M \cdot C_{23} - \sin(\alpha^M) \cdot C_{22} - \cos(\alpha^M) \cdot C_{21})^2 + (\rho^M \cdot C_{13} - \sin(\alpha^M) \cdot C_{12} - \cos(\alpha^M) \cdot C_{11})^2}} \\ \alpha^I = \tan^{-1}\left(\frac{\rho^M \cdot C_{23} - \sin(\alpha^M) \cdot C_{22} - \cos(\alpha^M) \cdot C_{21}}{\rho^M \cdot C_{13} - \sin(\alpha^M) \cdot C_{12} - \cos(\alpha^M) \cdot C_{11}}\right) \end{cases} \quad (A.21)$$

Usando o sistema (A.20) para obter o modelo do sensor  $h(\cdot)$ , é possível calcular, por derivação deste modelo, a matriz  $\mathbf{H}$  usada na fase de atualização do filtro de Kalman. Logo, os elementos de  $\mathbf{H}$  são apresentados no sistema (A.22).

$$\left\{ \begin{array}{l} \frac{\partial \alpha^I}{\partial x^M} = -\cos(\alpha^F) \cdot \frac{\partial \alpha^I}{\partial \rho^F} \\ \frac{\partial \rho^I}{\partial x^M} = -\cos(\alpha^F) \cdot \frac{\partial \rho^I}{\partial \rho^F} \\ \frac{\partial \alpha^I}{\partial y^M} = -\sin(\alpha^F) \cdot \frac{\partial \alpha^I}{\partial \rho^F} \\ \frac{\partial \rho^I}{\partial y^M} = -\sin(\alpha^F) \cdot \frac{\partial \rho^I}{\partial \rho^F} \\ \frac{\partial \alpha^I}{\partial \theta^M} = -K \cdot \det(\mathbf{A}) \cdot f_1 \\ \frac{\partial \rho^I}{\partial \theta^M} = \sqrt{K^3} \cdot \det(\mathbf{A}) \cdot [\rho \cdot \cos(\alpha^M) \cdot f_3 - \rho \cdot \sin(\alpha^M) \cdot f_4 + f_5] \\ \frac{\partial \alpha^I}{\partial \rho^M} = -K \cdot \det(\mathbf{A}) \cdot f_2 \\ \frac{\partial \rho^I}{\partial \rho^M} = \sqrt{K^3} \cdot \det(\mathbf{A}) \cdot [\sin(\alpha^M) \cdot f_3 - \cos(\alpha^M) \cdot f_4] \\ \frac{\partial \alpha^I}{\partial \alpha^M} = y_F^M \cdot \frac{\partial \alpha^I}{\partial x^M} - x^M \cdot \frac{\partial \alpha^I}{\partial y^M} - \frac{\partial \alpha^I}{\partial \theta^M} \\ \frac{\partial \rho^I}{\partial \alpha^M} = y^M \cdot \frac{\partial \rho^I}{\partial x^M} - x^M \cdot \frac{\partial \rho^I}{\partial y^M} - \frac{\partial \rho^I}{\partial \theta^M} \end{array} \right. \quad (A.22)$$

onde,

- $K = [(K_x C_t - K_t C_x)^2 + (K_t C_y - K_y C_t)^2]^{-1}$
- $f_1 = \rho^M \cdot a_{31} \cdot \cos(\alpha^M) + \rho^M \cdot a_{32} \cdot \sin(\alpha^M) - a_{33}$
- $f_2 = a_{31} \cdot \sin(\alpha^M) - a_{32} \cdot \cos(\alpha^M)$
- $f_3 = T_{13} \cdot \rho^M - T_{12} \cdot \sin(\alpha^M) - T_{11} \cdot \cos(\alpha^M)$
- $f_4 = T_{23} \cdot \rho^M - T_{22} \cdot \sin(\alpha^M) - T_{21} \cdot \cos(\alpha^M)$
- $f_5 = T_{33} \cdot \rho^M - T_{32} \cdot \sin(\alpha^M) - T_{31} \cdot \cos(\alpha^M)$

e, os elementos  $T_{ij}$  são obtidos a partir da matriz  $\mathbf{T}$  definida na equação (A.23).

$$\mathbf{T} = \begin{pmatrix} h_{11} \cdot C_{21} + h_{21} \cdot C_{11} & h_{11} \cdot C_{22} + h_{21} \cdot C_{12} & h_{11} \cdot C_{23} + h_{21} \cdot C_{13} \\ h_{12} \cdot C_{21} + h_{22} \cdot C_{11} & h_{12} \cdot C_{22} + h_{22} \cdot C_{12} & h_{12} \cdot C_{23} + h_{22} \cdot C_{13} \\ h_{13} \cdot C_{21} + h_{23} \cdot C_{11} & h_{13} \cdot C_{22} + h_{23} \cdot C_{12} & h_{13} \cdot C_{23} + h_{23} \cdot C_{13} \end{pmatrix} \quad (\text{A.23})$$