Antonio Henrique Pinto Selvatici

Construção de Mapas de Objetos para Navegação de Robôs

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Engenharia Elé-
trica.

São Paulo
2009

Antonio Henrique Pinto Selvatici

Construção de Mapas de Objetos para Navegação de Robôs

Tese apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção
do Título de Doutor em Engenharia Elé-
trica.

Área de concentração:
Sistemas Digitais

Orientador:
Profa. Livre Docente Anna Helena
Realí Costa

São Paulo
2009

Este exemplar foi revisado e alterado em relação à versão original sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 22 de abril de 2009.

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Selvatici, Antonio Henrique Pinto

Construção de Mapas de Objetos para Navegação de Robôs/ A. H. P. Selvatici. – ed. rev. – São Paulo, 2009.
154 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Robótica 2. Visão computacional 3. Inferência Bayesiana e redes de crença I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

Para Giselle, por todo amor e cuidados

Agradecimentos

Agradeço ao suporte financeiro do CNPq, que patrocinou este trabalho através dos procesos 140868/2005-4 (bolsa GD) e 201571/2007-2 (bolsa SWE).

Resumo

Como a complexidade das tarefas realizadas por robôs móveis vêm aumentando a cada dia, a percepção do robô deve ser capaz de capturar informações mais ricas do ambiente, que permitam a tomada de decisões complexas. Entre os possíveis tipos de informação que podem ser obtidos do ambiente, as informações geométricas e semânticas têm papéis importantes na maioria das tarefas designadas a robôs. Enquanto as informações geométricas revelam como os objetos e obstáculos estão distribuídos no espaço, as informações semânticas capturam a presença de estruturas complexas e eventos em andamento no ambiente, e os condensam em descrições abstratas.

Esta tese propõe uma nova técnica probabilística para construir uma representação do ambiente baseada em objetos a partir de imagens capturadas por um robô navegando com uma câmera de vídeo solidária a ele. Esta representação, que fornece descrições geométricas e semânticas de objetos, é chamada *O-Map*, e é a primeira do gênero no contexto de navegação de robôs. A técnica de mapeamento proposta é também nova, e resolve concomitantemente os problemas de localização, mapeamento e classificação de objetos, que surgem quando da construção de *O-Maps* usando imagens processadas por detectores imperfeitos de objetos e sem um sensor de localização global. Por este motivo, a técnica proposta é chamada *O-SLAM*, e é o primeiro algoritmo que soluciona simultaneamente os problemas de localização e mapeamento usando somente odometria e o resultado de algoritmos de reconhecimento de objetos.

Os resultados obtidos através da aplicação de *O-SLAM* em imagens processadas por uma técnica simples de detecção de objetos mostra que o algoritmo proposto é capaz de construir mapas que descrevem consistentemente os objetos do ambiente, dado que o sistema de visão computacional seja capaz de detectá-los regularmente. Em particular, *O-SLAM* é eficaz em fechar voltas grandes na trajetória do robô, e obtém sucesso mesmo se o sistema de detecção de objetos posuir falhas, relatando falsos positivos e errando a classe do objeto algumas vezes, consertando estes erros. Dessa forma, *O-SLAM* é um passo em direção à solução integrada do problema de localização, mapeamento e reconhecimento de objetos, a qual deve prescindir de um sistema pronto de reconhecimento de objetos e gerar *O-Maps* somente pela fusão de informações geométricas e visuais obtidas pelo robô.

Palavras-chave: Robótica. Visão computacional. Inferência Bayesiana e redes de crença.

Abstract

As tasks performed by mobile robots are increasing in complexity, robot perception must be able to capture richer information from the environment in order to allow complex decision making. Among the possible types of information that can be retrieved from the environment, geometric and semantic information play important roles in most of the tasks assigned to robots. While geometric information reveals how objects and obstacles are distributed in space, semantic information captures the presence of complex structures and ongoing events from the environment and summarize them in abstract descriptions.

This thesis proposes a new probabilistic technique to build an object-based representation of the robot surrounding environment using images captured by an attached video camera. This representation, which provides geometric and semantic descriptions of the objects, is called O-Map, and is the first of its kind in the robot navigation context. The proposed mapping technique is also new, and concurrently solves the localization, mapping and object classification problems arisen from building O-Maps using images processed by imperfect object detectors and no global localization sensor. Thus, the proposed technique is called O-SLAM, and is the first algorithm to solve the simultaneous localization and mapping problem using solely odometers and the output from object recognition algorithms.

The results obtained by applying O-SLAM to images processed by simple a object detection technique show that the proposed algorithm is able to build consistent maps describing the objects in the environment, provided that the computer vision system is able to detect them on a regular basis. In particular, O-SLAM is effective in closing large loops in the trajectory, and is able to perform well even if the object detection system makes spurious detections and reports wrong object classes, fixing these errors. Thus, O-SLAM is a step towards the solution of the simultaneous localization, mapping and object recognition problem, which must drop the need for an *off-the-shelf* object recognition system and generate O-Maps only by fusing geometric and appearance information gathered by the robot.

keyword: Robotics. Computer Vision. Bayesian Inference and belief networks.

Lista de Figuras

1.1	Exemplo de um O-Map simples de uma sala de estar, mostrando a altura dos objetos e larguras médias, assim como o rótulo das classes. Todos os objetos representados são considerados como tendo sido detectados com probabilidade 1.	24
1.2	Efeito do erro de localização do robô no problema da associação de dados. Em (a), (c), e (e), as estrelas representam a verdadeira posição das características, e a circunferência representa a postura do robô, com o traço interno mostrando sua orientação: o traço contínuo indica a postura real, enquanto o traço tracejado mostra a suposta postura. Em (b), (d), e (f), o círculo preto mostra a origem das coordenadas de medição, as cruzes representam as medições previstas no tempo t , e os quadrados ligados à origem representam as medições reais	27
2.1	Classificação dos tipos mais comuns de mapas utilizados em robótica, de acordo com seu nível de abstração. (a) Mapa grade de ocupação, representando mapas métricos, no mais baixo nível (b) Mapa de características de pontos, indicando a localização das árvores e da trajetória do robô, extraído de (SELVATICI; COSTA, 2007a) (c) Mapa topológico, que interliga marcos terrestres com os comandos de controle utilizados para se deslocar de um para qualquer outro, extraído de (VASSALLO, 2004).	37
2.2	Exemplos de mapas de características geométricas. (a) Mapa de linhas de teto, extraído do (FOLKESSON; JENSFELT; CHRISTENSEN, 2005). Mapa de planos texturizados, extraído de (THRUN et al., 2004).	41
3.1	Exemplo de O-Map, tal como proposto nesta tese. Neste exemplo, os objetos são representados por paralelepípedos, descritos por sua largura, altura e espessura. O plano verde representa o chão, que não faz parte do mapa. Transparência na representação do objeto indica incerteza quanto à sua existência.	53

3.2	Ilustração de como um O-Map pode ser usado para criar um vocabulário comum para a interação homem-robô. Este exemplo é meramente ilustrativo, e não foi realizado experimentalmente. (a) Vista superior de parte de uma cozinha representada por seus objetos. (b) Amostra de uma imagem usada para a construção do mapa, mostrando os copos detectados. A associação entre cada detecção e o objeto correspondente no mapa é utilizada pelo usuário humano para especificar o copo desejado.	55
3.3	Ilustração do cenário considerado pela técnica O-SLAM. O contorno do objeto detectado no exemplo é simplificado como um retângulo, que corresponde à projeção da seção longitudinal de um cilindro.	56
3.4	Ilustração de como a função de correspondência J particiona o conjunto de medições em N subconjuntos (aqui, $N = 4$). Na figura, círculos representam a postura do robô em diferentes instantes $i = 1, 2, \dots, 8$, e as estrelas representam os objetos detectados com índices $j = 1, 2, 3, 4$. As linhas representam as medições disponíveis, indexadas pelos pares (i, k) , que são mostrados apenas em parte, para não carregar a figura. As linhas são distribuídas entre os objetos por J , de modo que $J(i, k) = j$. Por exemplo, temos $J(2, 1) = 1$ e $J(4, 2) = 4$ na figura.	58
3.5	Ilustração de como o conhecimento sobre o tamanho e posição do objeto podem ser utilizados para estimar a distância relativa da câmera. (a) A diminuição do tamanho da caixa de madeira nas imagens combinada com o conhecimento prévio sobre o tamanho das caixas informa sobre a sua crescente distância para a câmera. (b) modelo probabilístico gráficos que ilustram as dependências entre posição e tamanho dos objetos, e postura da câmera do robô (incógnitas são representados por elipses) dada a saída do algoritmo de detecção de objetos (retângulos).	61

4.1	Fragmento da Rede Bayesiana que representa o modelo probabilístico adotado por O-SLAM. Os retângulos representam variáveis conhecidas, e os círculos representam as incógnitas. Estamos basicamente assumindo que a classe do objeto c_j dá uma idéia aproximada sobre a sua geometria g_j , e pode dar pistas sobre sua localização l_j . Em SLAM geométrico, a classe detectada $a_{i,k}$ é considerada igual à real, $c_{J(i,k)}$, de forma que $P(a_{i,k} c_{J(i,k)}) = \delta(a_{i,k}, c_{J(i,k)})$. Este modelo não admite oclusão entre os objetos, nem objetos em movimento.	68
4.2	Representação de como casar a mesma identidade do objeto em dois mapas gerados assumindo diferentes soluções para a associação de dados. Objetos 1, 3 e 4 têm a mesma identidade em ambas as funções de correspondência representadas em (a) e (b), enquanto que o objeto 2 em (a) e os objetos 2 e 5, em (b), não podem ser igualados se utilizada a atual definição de identidade de objeto.	76
4.3	Exemplo gráfico de uma função sigmóide espelhada com parâmetros $\tau = 0.1$ e $t_0 = 0.3$	85
4.4	Exemplo ilustrativo que representa a probabilidade de associação de dados.	88
4.5	Gráfico dos resíduos res_1 e res_2 para os valores de σ_o e σ_r indicados. .	89
4.6	Comparação gráfica do modelo de evidência considerando as soluções da associação de dados J_1 e J_2 , com diferentes desvios padrão.	90
5.1	Ilustração de como o algoritmo MH amostra distribuições com picos de modo a gerar cadeias com mixagem ruim. (a) Densidade alvo, com picos centrados em $x = 4$ e $x = 12$. (b) Gráfico das amostras extraídas pelo algoritmo MH iniciando em $x = 4$ e propondo novos estados dentro de uma janela fixa em torno do estado atual. Os estados pertencendo ao pico à direita não são amostrados, indicando uma mixagem ruim. As figures foram extraídas de (RANGANATHAN, 2008).	103

5.2	Ilustração de como MC ³ com cadeias aquecidas ajudam a melhorar a mixagem quando distribuições com picos são amostradas. (a) Densidade alvo em temperaturas diferentes: $T = 1$ (linha preta), representando a densidade original, a mesma da Figura 5.1, $T = 11$ (azul), e $T = 21$ (vermelha). (b) Gráfico das amostras extraídas de pelo algoritmo MC ³ , usando o mesmo estado inicial e densidade de propostas do exemplo da Figura 5.1. Neste caso, o amostrador rapidamente alterna entre ambos os picos, indicando boa mixagem da cadeia. As figuras foram extraídas de (RANGANATHAN, 2008).	104
6.1	Exemplo de imagem anotada do conjunto de dados Cogniron. Os objetos são marcados com polígonos delimitadores, e receberam rótulos manualmente.	112
6.2	Ilustração dos dados medidos e do modelo de objeto usado nos experimentos de SLAM geométrico. (a) Modelo cilíndrico adotado para representar os objetos, descritos por sua altura, largura, e posição do ponto de base. (b) Posições de objetos detectados e tamanhos nas imagens. O tamanho é descrito por uma fatia delimitadora do objeto, extraída do polígono anotado, que consiste no tamanho radial e na largura angular. Os pequenos círculos indicam a projeção dos pontos de base do objeto, usados para descrever suas posições aparentes nas imagens.	113
6.3	Resultados de O-SLAM geométrico obtidos com o conjunto de dados Cogniron. (a) Visão perspectiva do O-Map obtido em conjunto com a trajetória inferida (b) Visão superior do modelo de objetos construído. A linha pontilhada preta representa a trajetória inferida, enquanto que a linha pontilhada verde-escura é a trajetória obtida por SLAM baseado em laser, que consideramos como nossa referência. A planta da casa foi extraída do artigo de Zivkovic, Booij & Kröse (2007).	115
6.4	Mais resultados obtidos com o conjunto de dados Cogniron, ilustrando o desempenho da abordagem proposta para SLAM geométrico. (a) Comparação entre as trajetórias. A trajetória inferida está mais próxima da trajetória de referência. (b) Exemplo de projeção de objetos do mapa no plano da imagem.	116

6.5	Fotos do robô e de um objeto presente no conjunto de imagens do LPA. (a) O robô Pioneer 3-DX carrega um sistema de imagem onidirecional, constituída por uma câmera de vídeo acoplada a um espelho hiperbólico. (b) Exemplo de um objeto "green box" detectado por CMVision nos experimentos. Cada classe de objeto é representada por uma cor diferente.	118
6.6	Posições de objetos detectados e tamanhos nas imagens pertencentes ao conjunto de dados LPA. Observa-se que o tamanho da "orange door" é detectado imprecisamente.	119
6.7	Resultados de SLAM geométrico obtidos com o conjunto de dados LPA ajustados manualmente. (a) Visão perspectiva do O-Map obtido em conjunto com a trajetória inferida. As cores do objeto são a média delas utilizadas para detectá-las em imagens. (b) Visão superior do modelo de objetos construídos. A linha preta pontilhada representa a trajetória inferida, enquanto a linha verde é a trajetória obtida por SLAM baseado em laser, considerada como referência. O mapa de grade de ocupação foi gerado utilizando-se DP-SLAM. Note que green box sobrepõe a maioria dos objetos detectados por laser.	121
6.8	Mais resultados obtidos com o conjunto de dados LPA, ilustrando o desempenho do modelo geométrico de O-SLAM. (a) A comparação entre as trajetórias. A inferida trajetória está mais próxima da referência. b) Exemplo de projecção de objectos do mapa no plano da imagem. . . .	122
6.9	Resultados de O-SLAM obtidos com o conjunto de dados LPA e classes de objetos conhecidas, embora com solução de associação de dados desconhecida. (a) visão perspectiva do O-Map obtido em conjunto com a trajetória inferida. As cores do objeto são a média daquelas utilizadas para detectá-las em imagens. Objetos espúrios estão circulados. (b) Visão superior do O-Map construído. A linha pontilhada preta representa a trajetória inferida, enquanto a linha verde é a trajetória obtida pela DP-SLAM. (c) Detalhe da região em torno do "green box" não-espúrio na Figura 6.9b, mostrando três objetos daquela classe (dois deles são perfeitamente sobrepostos). Os triângulos vermelhos indicam o centro. A transparência indicaria a probabilidade de existência do objeto, mas dois objetos são tão transparentes que não podem sequer serem vistos.	125

6.10 Vista superior do O-Map obtido considerando o modelo proposto de es- puriedade do objeto. (a) Mapa completo e trajetória, em que a transpa- rência indica a probabilidade de existência do objeto. (b) A parcela dos objetos com $\pi_{ID} > 0.5$	127
6.11 Exemplo de projeção dos objetos no mapa da Figura 6.10b sobre o plano da imagem. Observa-se que a perna da pessoa na imagem é detectada como "green box".	128
6.12 Comparação entre histogramas de amostras obtidos pelos algoritmos MC ³ e MH. (a) Histogramas obtidos com MC ³ . (b) Histogramas obtidos utilizando o algoritmo MH processando os mesmos dados.	129
6.13 O-Map e a trajetória obtida utilizando o conjunto de dados LPA, com o pressuposto errado de classes de objetos perfeitas. Os objetos exhibi- dos têm probabilidade de existência $\pi_{ID} > 0.1$. (a) Visão superior do mapa, mostrando também a grade de ocupação obtida com DP-SLAM. (b) Visão perspectiva do mapa, mostrando objetos sobrepostos corres- pondentes ao mesmo físico ("orange ball") detectou como pertencentes a duas classes distintas ("orange ball" e "orange door").	130
6.14 O-Map e trajetória obtida utilizando o conjunto de dados LPA e permi- tindo a confusão entre as classes. Os objetos exibidos têm a probabi- lidade de existência $\pi_{ID} > 0.1$. (a) Visão superior do mapa, mostrando também a grade de ocupação obtida com DP-SLAM. (b) Visão perspec- tiva do mapa. Agora, orange ball está corretamente mapeada.	131
6.15 Posições de objeto detectadas e tamanhos em imagens pertencentes ao conjunto de dados LPA. Observa-se que orange ball é detectada por duas vezes: uma vez como pertencente à classe correta, e outra vez como objeto "orange door".	132
6.16 Posições de objetos detectados e tamanhos em imagens pertencen- tes ao conjunto grande de dados LPA. (a) Imagem mostrando somente detecções não-espúrias. (b) Imagem mostrando medidas espúrias re- alizadas a partir da porta e dos objetos vermelhos na parte esquerda da imagem, que não se destinaram a serem detectadas no presente experimento.	133

6.17 Resultados de O-SLAM obtidos com conjunto grande de dados LPA, mostrando os objetos com $\pi_D > 0.5$. (a) Visão perspectiva de uma parte do O-Map obtido, mostrando os objectos detectados no interior do laboratório. Os objetos espúrios estão circulados. (b) Visão superior do modelo de objetos construído. A linha pontilhada preta representa a trajetória inferida, enquanto a linha verde é a trajetória obtida por SLAM baseado em laser, considerado como referência. O mapa de grade de ocupação foi gerado utilizando DP-SLAM. Note que nenhum objeto do lado de fora da sala é mostrado.	134
6.18 Mais resultados obtidos com o conjunto grande de dados LPA, ilustrando o desempenho do algoritmo proposto em estimar os parâmetros geométricos corretos. (a) Comparação entre as trajetórias. A trajetória inferida está mais próxima à referência. Nesta figura, são mostrados todos os objetos mapeados, representados por triângulos vermelhos. (b) Exemplos de projeção de objetos com baixa probabilidade de espuriidade no mapa no plano da imagem. A imagem à esquerda enfatiza projeções dos objetos mapeados corretamente, enquanto a imagem à direita enfatiza as projeções de objetos espúrios registrados com uma baixa probabilidade de espuriidade no mapa.	135
A.1 Ilustração das variáveis envolvidas no modelo de movimento do robô. O círculo representa o robô e o segmento de linha dentro indica a orientação com respeito ao sistema de global de coordenadas.	151
A.2 Representação esquemática da câmera omnidireccional usada nos experimentos do capítulo 6 (ZIVKOVIC; BOOIJ, 2005).	152
A.3 Representação esquemática de como um ponto no mundo é projetado no plano de imagem	154

Lista de Tabelas

6.1	Classes detectadas no conjunto de imagens Cogniron e a crença prévia assumida sobre os tamanhos de suas instâncias.	114
6.2	Classes detectadas no conjunto de imagens LPA e crença prévia assumida dos tamanhos de suas instância.	119
6.3	Comparação entre os tamanhos dos objetos inferidos com o algoritmo SLAM geométrico proposto e os tamanhos corretos. Cada objeto pertence a uma classe diferente. Os diâmetros corretos de objetos hexaédricos — volume com seis faces — especificamente, yellow box, green box e orange door, foram calculados utilizando-se a norma euclideana dos valores correspondentes de comprimento e largura na Tabela 6.2.	120
6.4	Classes detectadas no conjunto de imagens LPA e a crença prévia assumida sobre os tamanhos de suas instância.	133

Lista de Algoritmos

1	Algoritmo O-SLAM considerando detecção e reconhecimento de objetos perfeito.	73
2	Construção de um O-Mapa a partir de amostras de soluções para associação de dados J_n e medições Z	79
3	Construção de um mapa de objetos a partir de amostras (C_n, J_n)	84
4	O algoritmo Metropolis-Hastings para amostrar $P(C, J Z)$	94
5	Proposição de um movimento de fusão	98
6	Proposição de um movimento de fusão	99
7	Proposição do movimento de troca de classes	100
8	O algoritmo Metropolis-Coupled MCMC aplicado à amostragem de $P(C, J Z)$	105
9	Estratégia proposta para melhorar o processo de amostragem	110

Sumário

1	Introdução	20
1.1	Mapa baseado em objetos para navegação do robô	23
1.2	O problema de SLAM e fechamento de volta	25
1.3	SLAM baseado em objetos	29
1.4	Objetivos	30
1.5	Contribuições	32
1.6	Organização	34
2	Mapeamento Robótico	36
2.1	Tipos de Mapas	37
2.1.1	Mapas métricos	38
2.1.2	Mapas topológicos	39
2.1.3	Mapas de características	40
2.1.4	Mapas de objetos	42
2.2	O problema de SLAM	43
2.2.1	Problemas relacionados	44
2.2.2	Técnicas de estimação	46
2.2.3	Problema de associação de dados	48
3	O-Map	52
3.1	Aplicações de O-Maps	54
3.2	Arcabouço para O-SLAM	55

3.2.1	Hipóteses sobre o sistema de detecção de objetos	57
3.2.2	Acoplamento das incógnitas através de modelagem probabilística	59
3.2.3	Algoritmos de visão computacional compatíveis com O-SLAM .	61
3.3	Comparação com outros mapas baseados em objetos	63
4	Inferência em O-SLAM	65
4.1	Solução eficiente para o problema de SLAM geométrico	66
4.1.1	Usando modelos gaussianos linearizados	69
4.1.2	Inferência usando decomposição QR	70
4.2	Associação de dados probabilística	72
4.2.1	Identidades de objetos	75
4.2.2	Aproximando a crença sobre a associação de dados	78
4.3	Tratando detecção imperfeita de objetos	79
4.3.1	Modelo de confusão entre as classes	80
4.3.2	Filtragem de objetos espúrios	82
4.4	Distribuição de probabilidade sobre as soluções da associação de dados e classes dos objetos	85
4.4.1	Calculando a evidência	86
5	Amostragem por MCMC	92
5.1	O algoritmo de Metropolis-Hastings	93
5.2	Distribuição de propostas	95
5.2.1	Transição de estados proposta	96
5.3	Questões práticas sobre o algoritmo de amostragem proposto	101
5.3.1	Melhorando a mixagem com o algoritmo Metropolis-Coupled . .	102
5.3.2	Melhorando a eficiência da transição de estados	104
5.3.3	Guiando a convergência da cadeia	108

6	Resultados Experimentais de O-SLAM	111
6.1	Resultados experimentais de SLAM geométrico	111
6.1.1	Conjunto de dados Cogniron	111
6.1.2	Conjunto de dados LPA	117
6.2	Resultados sobre associação de dados probabilística	123
6.3	Resultados de O-SLAM assumindo confusão entre classes	128
7	Conclusão e Trabalhos Futuros	137
7.1	Contribuições	138
7.2	Perspectivas	140
	Referências	142
	Apêndice A – Modelos utilizados em SLAM geométrico	150
A.1	Modelo de movimento	150
A.2	Modelo de observação	152

1 Introdução

Robôs móveis são sistemas incorporados no mundo real que se movem autonomamente e interagem com ele para realizar suas tarefas. Instâncias de tais robôs são, até certo ponto, freqüentes hoje em dia, e as tarefas a eles designadas estão aumentando em complexidade. Desativação de minas, vigilância autônoma, limpeza de soalhos e assistência a idosos ou pessoas incapazes estão entre as tarefas realizadas atualmente pelos robôs, as quais requerem por parte deles o reconhecimento de sua localização espacial, bem como as ações a serem tomadas em cada possível situação.

Como as tarefas realizadas pelos robôs estão aumentando em complexidade, sua percepção deve ser capaz de capturar informações mais ricas do ambiente de tal forma que possam tomar decisões mais complexas. Como a pesquisa em robótica tem sido focada, durante a última década, principalmente no problema de navegação, a localização do robô e os mapas métricos de obstáculos constituíam o principal interesse nos trabalhos relativos à percepção dos robôs. Enquanto a localização e a construção de mapas métricos constituem ainda campos de pesquisa ativos, outros tipos de percepção são agora mais profundamente investigados, tais como reconhecimento e mapeamento de objetos, classificação do local, processamento da linguagem natural, reconhecimento da face e de gestos, entre outros. Particularmente, uma grande ênfase é dada no processamento da informação visual, pois a visão, sendo constituído por um sensor de baixo custo e com baixo consumo energético, é muito flexível em proporcionar abundantes informações que podem ser utilizadas para diversos propósitos. Além disso, um grande progresso tem sido alcançado no campo da visão computacional, de onde muitas técnicas podem ser empregadas na obtenção da percepção do robô.

Um tópico particular na visão computacional, que tem mostrado muitos avanços nos últimos anos, é o do reconhecimento de objetos e suas variantes, tais como a detecção dos objetos e sua categorização. Foram propostas ulteriormente, técnicas

probabilísticas que lidam com uma grande quantidade de dados para selecionar as características mais significativas das imagens, as quais produzem resultados impressionantes na segmentação de imagens e reconhecimento de objetos ou sua categorização (PINZ, 2005). Em robótica, a utilização de alguma técnica de visão computacional para o reconhecimento e mapeamento de objetos no ambiente tem se tornado freqüente, proporcionando boas taxas de detecção e de reconhecimento (EKVALL; KRAGIC; JENSFELT, 2007; FORSSEN et al., 2008).

Entretanto, muito poucos trabalhos exploram o movimento dos robôs nas tarefas de reconhecimento de objetos. O bom desempenho atual das técnicas de reconhecimento visual de objetos sugere que seu tratamento seja como a de outros sensores robóticos, de tal forma que suas saídas são utilizadas tal como se apresentam, considerando o algoritmo da visão computacional como uma caixa preta. Apesar disso, as técnicas de reconhecimento de objetos não são caixas pretas e estão sujeitas a falhas. O modelo de movimento do robô serve, portanto, como uma ferramenta poderosa que pode ser utilizada em complemento à visão computacional quando do reconhecimento e mapeamento de objetos. A interação do robô com o mundo, enquanto ele adquire dados continuamente, proporciona uma fonte rica de informações que pode corroborar ou contradizer a saída da visão computacional, conduzindo a resultados mais confiáveis no reconhecimento. Contudo, como uma consequência deste enfoque integrado, os objetos são reconhecidos em um certo lugar, ao invés de serem procurados em uma imagem ou rastreado em uma sequência de imagens.

O principal objetivo do presente trabalho é desenvolver uma estrutura que produz modelos probabilísticos simples dos objetos no ambiente, integrando a saída de uma classe específica de algoritmos de reconhecimento de objetos com informações sobre o movimento espacial do robô. Aqui, os algoritmos da visão computacional não são tratados como caixas pretas. Em vez disso, as identidades dos objetos no ambiente, bem como suas localizações e tamanhos, são consideradas as incógnitas do problema. O conhecimento sobre as limitações no reconhecimento do objeto, integrado ao conhecimento sobre como o robô se move e interage com o meio ambiente, restringe a localização tanto do robô quanto do objeto, bem como os tamanhos e as classes dos próprios objetos, tornando possível a inferência sobre tais incógnitas, permitindo ainda a atribuição de níveis de crença a diferentes valores dessas variáveis. Na literatura tradicional sobre robótica, as identidades dos objetos são consideradas como dados, produzido pelo algoritmo de reconhecimento do objeto.

Portanto, este trabalho é um avanço no estado da arte sobre pesquisa em robótica, pela proposição da primeira técnica probabilística de mapeamento de objetos no ambiente, ao mesmo tempo em que estima a localização do robô, sem a utilizar qualquer outro sensor externo, imagens estereográficas, nem outras características que não a detecção de objetos em imagens. A presente pesquisa representa um passo em direção a um cenário onde os objetos no ambiente são modelados fundindo-se informações extraídas diretamente pelas imagens com informações sobre o movimento do robô, abandonando a necessidade de fiar-se em algum algoritmo pronto de reconhecimento de objeto. Entretanto, neste trabalho, tal algoritmo ainda é necessário. A diferença entre trabalhos correlatos é que agora, fundindo-se a saída deles com o conhecimento sobre o movimento do robô, permite-se a revisão da informação proporcionada pelo reconhecimento do objeto, que restringe simultaneamente a localização do robô e as localizações e tamanhos dos objetos no ambiente.

Uma vez que o foco aqui é mostrar como o movimento do robô e as técnicas de visão computacional podem ser acopladas para mapear os objetos num determinado lugar de forma probabilística, sem a utilização de um sensor externo auxiliar, a abordagem proposta é demonstrada utilizando-se uma técnica de reconhecimento de objeto muito simples, que assume um ambiente controlado. Os erros no reconhecimento de objetos são então balanceados, integrando-se as informações reunidas pelo robô enquanto se move pelo ambiente, de tal forma que possa ser produzido um mapeamento que descreve coerentemente os objetos. Implementar um algoritmo de reconhecimento de objetos, robusto e de propósito geral, e integrá-lo à estrutura proposta para garantir a qualidade dos resultados produzidos em condições reais e não controláveis, está fora do escopo deste trabalho.

Ao mesmo tempo, esta tese pretende contribuir com a área de pesquisa sobre mapeamento robótico. A abordagem proposta gera um mapa contendo a geometria 3D dos objetos no ambiente, junto com suas classes. Uma classe de objetos é um rótulo semântico, indicando qual tipo de objeto se trata, como uma porta, uma mesa, uma bola, etc.. O mapa resultante é chamado *O-Map*, e constitui um avanço sobre outros mapas de objetos utilizado em robótica devido à representação consistente e compacta utilizada, permitindo a localização do robô sem a necessidade de quaisquer estruturas auxiliares além dos objetos em si. Portanto, a contribuição deste trabalho reside em ambos, o mapeamento proposto e o processo de construí-lo, onde as partes geométricas e semânticas são inferidas de forma acoplada.

1.1 Mapa baseado em objetos para navegação do robô

A navegação do robô é um tópico abrangente e envolve diferentes questões a serem endereçadas. Grosso modo, a navegação de um robô móvel corresponde à tarefa de autonomamente mover-se de uma posição específica, ou região, para uma outra posição ou região. Esta tarefa pode envolver simplesmente o movimento do quarto das crianças para a cozinha, em uma casa, o movimento entre um prédio e outro no campus da universidade, ou até uma viagem entre uma cidade e outra. Dependendo do método de locomoção do robô, existem também diferentes restrições ao caminho dele: um movimento terrestre do robô na cidade pode ser pela calçada — para robôs de baixa velocidade — ou ruas — para os de alta velocidade, porém usualmente, um robô aéreo pode ir diretamente ao seu destino. Se a área de movimentação do robô for restringida por paredes, objetos, pessoas, ou outros obstáculos, ele deve encontrar seu caminho para evitá-los e alcançar sua posição de destino. Finalmente, o robô deve sempre conhecer sua posição em relação ao seu objetivo final.

Muitas das questões envolvidas na navegação de robôs podem ser resolvidas com a ajuda de um mapa. Um mapa é uma representação espacial utilizada para descobrir a localização de elementos especiais. Na navegação autônoma, os elementos representados em um mapa podem ser utilizados para diversas finalidades. Por exemplo, eles podem ser usados para planejar um caminho entre a posição atual do robô e seu destino, especialmente se o mapa representar as áreas por onde ele for permitido navegar. Os mapas podem também ser utilizados para localizar o robô: comparando os elementos sensoriados no ambiente com aqueles registrados no mapa, o robô pode inferir o lugar ou possíveis lugares onde está.

Neste trabalho, um tipo novo de mapa para a navegação de robôs, denominado O-Map, é proposto. O O-Map contém informações geométricas e semânticas sobre certos objetos no ambiente, conforme mostrado na Figura 1.1. A informação geométrica corresponde ao tamanho e posição dos objetos, modelados por formas geométricas 3D simples, e a informação semântica é dada pelas classes de objetos. O mapa proposto proporciona também uma medida da certeza sobre a própria detecção dos objetos, característica que será explicada mais adiante no Capítulo 3. Ao longo deste trabalho, as geometrias dos objetos são assumidas como sendo bem descritas por cilindros, mas a abordagem proposta pode ser estendida a qualquer forma geométrica.

O O-Map tem algumas vantagens quando comparado com mapas tradicionais uti-

Figura 1.1: Exemplo de um O-Map simples de uma sala de estar, mostrando a altura dos objetos e larguras médias, assim como o rótulo das classes. Todos os objetos representados são considerados como tendo sido detectados com probabilidade 1.

lizados para localizar o robô. Usualmente, esses mapas registram pontos característicos ou linhas no ambiente, os quais estão presentes numa maior quantidade do que objetos. Embora este número de elementos dependa do método para detectar pontos e linhas em imagens ou varreduras de laser, é certo, em geral, que objetos são esparsos. A menor densidade de objetos do que elementos geométricos mais básicos numa imagem é favorável à eficiência dos algoritmos de localização do robô, pois o espaço de possíveis correspondências entre os elementos detectados pelo robô e aqueles registrados no mapa é muito menor neste caso. Encontrar as correspondências corretas constitui geralmente o gargalo dessa tarefa.

Entretanto, as aplicações dos O-Maps vão além da localização de robô. Uma vez que contém informação semântica, eles podem ser usados como entrada para alguma tarefa lógica ou algoritmo de planejamento de caminho baseado em objetos. O mapa proposto pode ser especialmente útil para sistemas que interagem com humanos, como em aplicações de Realidade Aumentada — em inglês, *Augmented Reality* (AR)—, interação humano-computador — em inglês, *Human-Computer Interaction* (HCI) —, e robôs de serviço. Nessas aplicações, a representação de mundo utilizada deve compartilhar símbolos com aquela dos humanos, e há evidências que as próprias pessoas utilizam objetos para representar espaços internos (VASUDEVAN et al., 2007). Mesmo que a representação geométrica não for realista, os rótulos semânticos junto com as posições aproximadas dos objetos constituem uma valiosa ferramenta de informação que permitem a um usuário entender o espaço representado e associar os objetos mapeados com aqueles mostrados numa imagem do mesmo.

Embora outros mapas que representam objetos no ambiente têm sido propostos na literatura de robótica, a representação adotada neste trabalho e o processo utilizado

em sua obtenção são novos, e constituem a principal contribuição desta tese. Ao invés de simplesmente inserir os objetos detectados num mapa, construído utilizando-se enfoques tradicionais, aqui todo o mapa e a trajetória do robô são inferidos juntos a partir dos dados adquiridos. Em especial, as partes semântica e geométrica do mapa influenciam-se mutuamente, sendo calculados de forma casada.

Esta tese propõe também um algoritmo para a construção de O-Maps através das imagens capturadas por um robô móvel que não conhece sua localização precisamente. Obter um mapa baseado em objetos concorrentemente com a localização do robô corresponde a um problema *SLAM*. SLAM é o acrônimo para *Simultaneous Localization and Mapping*, que em português traduz-se como localização e mapeamento simultâneos, que é o problema de determinar a localização do robô e um mapa para o ambiente através dos dados adquiridos durante sua exploração pelo próprio robô. Assim, o algoritmo proposto para a obtenção dos O-Maps é chamado *O-SLAM*, e propões soluções aos desafios encontrados pelas técnicas tradicionais de SLAM.

1.2 O problema de SLAM e fechamento de volta

O problema de SLAM emerge sempre que for requerido a um robô construir um mapa do ambiente sem o conhecimento preciso de sua localização. É um campo de pesquisa muito ativo, e muitas variantes do problema tem sido estudadas a tempos. A solução completa para este problema é considerado como um dos desafios em robótica (DISSANAYAKE et al., 2001; FOLKESSON; JENSFELT; CHRISTENSEN, 2005).

Dentre os sub-problemas que compõem o problema completo de SLAM, um muito desafiador é o *problema de associação de dados* (THRUN, 2002). Ele surge usualmente nos problemas de rastreamento e mapeamento, onde certas características detectadas nos dados de entrada dos sensores devem ser ou rastreados no tempo ou ter sua localização registrada num mapa. Nesses problemas, a medição gerada por uma certa característica detectada é utilizada para atualizar a localização estimada dessa característica. Em qualquer caso, todas as características detectadas devem ser associadas entre si, definindo quais detecções são devidas a uma característica específica, dando origem ao problema da associação de dados. Por exemplo, se o robô tem um detector de objetos do tipo "copo" e detecta um copo em duas imagens diferentes, o problema da associação de dados corresponde a determinar se ambas as detecções correspondem ao mesmo copo ou se dois diferentes copos foram detec-

tados, baseado nas medições geradas por essas detecções. Embora várias técnicas têm sido propostas para resolver este problema dentro do contexto SLAM, encontrar a solução ótima para a associação de dados é um problema intratável.

O problema da associação de dados é formalmente e exatamente expresso como um problema de particionamento de conjuntos, estudado na teoria dos conjuntos (ROTA, 1964). Dado um conjunto de detecções de características, onde uma característica pode ser um ponto, uma linha, um objeto ou qualquer estrutura elementar representada no mapa, o problema da associação de dados consiste em agrupá-las nos sub-conjuntos corretos, cada uma correspondendo a uma característica física diferente percebida no mundo. As medições geradas pelas detecções de características pertencentes a um determinado sub-conjunto é então atribuída a uma característica única no mundo, e são utilizadas para construir uma representação dessa característica no mapa. Desta forma, o tamanho do espaço para possíveis soluções da associação de dados é igual ao número de combinações das possíveis maneiras de particionar o conjunto das detecções de características. É interessante notar que, dada uma determinada solução para o problema da associação de dados, o número das características registradas no mapa é igual ao número de sub-conjuntos não vazios, nos quais as detecções de características são agrupadas.

Associar medições obtidas em sucessivas detecções de características é muito mais fácil do que associar medições obtidas em um longo período de tempo. Isto é ilustrado na Figura 1.2 para o problema específico de SLAM, onde o robô detecta a distância e a direção de pontos característicos especiais do cenário. No caso de SLAM, a forma da trajetória do robô é geralmente bem conhecida dentro de pequenos trechos. Assim, admitindo que algumas características são detectadas e medidas no instante de tempo t , as medições previstas dessas características fica muito próximo das medições reais realizadas nos instantes sucessivos $t + 1, t + 2, \dots, t + n$, como ilustrado na Figuras 1.2c e 1.2d. Neste caso, é fácil fazer corresponder as novas medições com as anteriores. No entanto, com o passar do tempo e a localização do robô tornar-se incerta, a projeção prevista das medições efetuadas no intervalo de tempo t vai recair para além das medições reais e então, decidir se a mesma característica ou outras foi detectada é uma tarefa mais difícil, como ilustrado nas Figuras 1.2e e 1.2f.

Este efeito ocorre em problemas de SLAM em geral, independentemente do tipo de sensor utilizado. Assim, para o caso mais simples, quando a associação de dados considera medições realizadas dentro de um curto período de tempo ou quando a

Figura 1.2: Efeito do erro de localização do robô no problema da associação de dados. Em (a), (c), e (e), as estrelas representam a verdadeira posição das características, e a circunferência representa a postura do robô, com o traço interno mostrando sua orientação: o traço contínuo indica a postura real, enquanto o traço tracejado mostra a suposta postura. Em (b), (d), e (f), o círculo preto mostra a origem das coordenadas de medição, as cruzes representam as medições previstas no tempo t , e os quadrados ligados à origem representam as medições reais

postura do robô não muda muito, técnicas do vizinho mais próximo (NN) funcionam bem. No entanto, um caso mais difícil é quando o robô revisita áreas previamente visitadas e novas medições têm de ser associadas com as obtidas anteriormente. O problema da associação de dados neste caso é também conhecido como problema do fechamento de volta, e requer técnicas mais sofisticadas para ser resolvido. Em geral, estas técnicas são exploratórias, e buscam o espaço das soluções da associação de dados para a melhor correspondência entre as detecções de acordo com algum critério. No entanto, enumerar todo o espaço de associações de dados é inviável, e qualquer técnica exploratória deve depender de uma boa estratégia de busca, e podar o espaço sempre que possível.

Neste trabalho, a abordagem utilizada para a associação de dados centra-se em resolver o problema do fechamento de volta. No entanto, em vez de determinar uma única associação de dados, este trabalho propõe que seja encontrado um punhado de possíveis soluções. Uma vez que esta abordagem gera diferentes mapas a partir dos mesmos dados, possivelmente com um número diferente de objetos cada, o O-Map final é construído pela combinação desses mapas, considerando-se a expectativa dos parâmetros do objeto mapeado. Isto pode ser útil no caso em que não há dados suficientes para determinar uma única e boa solução da associação de dados. Por exemplo, pode haver casos em que não é possível assegurar que exista um único "relógio" no quarto, ou se há dois "relógios" semelhantes.

Para encontrar prováveis soluções da associação de dados, espaço é amostrado utilizando as técnicas de *Markov Chain Monte Carlo* (MCMC), que utilizam as propriedades teóricas das cadeias de Markov para extrair amostras de qualquer distribuição alvo. Os métodos MCMC fornecem uma estrutura bem fundamentada para inferência aproximada nos casos em que uma inferência exata seja intratável, substituindo métodos *ad hoc* ou heurísticos que seriam de outra forma necessários (GILKS; RICHARDSON; SPIEGELHALTER, 1996a). Trabalhos relacionadas mostram que estes métodos podem fornecer algoritmos práticos para amostrar com êxito o espaço desafiador das soluções da associação de dados (DELLAERT, 2001; KAEISS; DELLAERT, 2005; RANGANATHAN, 2008), desde que bem sintonizados pela utilização de conhecimentos específicos sobre o problema.

1.3 SLAM baseado em objetos

Neste trabalho, as características representadas no mapa correspondem aos objetos detectados pelo robô enquanto explora o ambiente. A proposta do algoritmo O-SLAM considera que o robô captura imagens através de uma câmera de vídeo e executa algoritmos de reconhecimento do objeto, especializados na detecção de certas classes de objetos nessas imagens. Além das classes presumidas, as medições dadas pelos detectores consistem das posições e tamanhos aparentes dos objetos nas imagens.

Um importante pressuposto subjacente a este trabalho é que, a despeito dos bons resultados que os algoritmos modernos de reconhecimento de objeto podem produzir, eles ainda estão sujeitos a falhas. Portanto, ao mesmo tempo o O-SLAM utiliza detectores de objetos existentes como sensores básicos, a saída desses algoritmos não é totalmente confiável; em vez disso, ele é usado para alimentar um mecanismo probabilístico de inferência que revisa e filtra as medições feitas ao mesmo tempo em que constrói o mapa e encontra a trajetória do robô. Este trabalho modela o comportamento dos detectores de objeto assumindo que seus modos de erro correspondem a:

- tamanho e posição imprecisos do objeto na imagem;
- classificação errada do objeto, caracterizada pela detecção de um objeto pertencente a outra classe que não a presumida, e
- detecção espúria, caracterizada pela detecção de um pedaço da imagem que não corresponde a um objeto consistentemente detectado ao longo do tempo.

Este trabalho propõe que seja utilizada uma técnica MCMC para realizar uma inferência aproximada das classes de objeto, de modo análogo ao que é feito para a solução da associação de dados. A idéia-chave para isso é acoplar a probabilidade das classes de objeto com os parâmetros geométricos do mapa. Portanto, o modelo desenvolvido para descrever sua distribuição de probabilidade não permite a inferência independente das classes de objeto, suas localizações e geometrias, e da localização do robô. Como consequência, o procedimento da inferência deve explorar o espaço conjunto das classes de objeto, analogamente à inferência na solução da associação de dados. Para fazer isso, O-SLAM utiliza métodos MCMC para amostrar o espaço

conjunto das classes de objetos e das soluções da associação de dados, adicionando uma nova dimensão ao problema.

Dada uma amostra do par solução da associação de dados e classes de objeto, O-SLAM trata a imprecisão das medições geométricas utilizando o arcabouço de *Square-Root Smoothing and Mapping* ($\sqrt{\text{SAM}}$) (DELLAERT; KAESSE, 2006). $\sqrt{\text{SAM}}$ é uma técnica do estado-da-arte de estimação geométrica no problema de SLAM que modela o problema de obter conjuntamente a trajetória do robô e o mapa como otimização não-linear dos mínimos quadrados (LS), tratado por métodos iterativos que resolvem uma versão linearizada do problema, a cada etapa. Desde que a matriz hessiana do sistema linearizado é esparso, ele é resolvido eficientemente usando a fatorização QR, uma técnica da álgebra linear especialmente adequada para sistemas esparsos (GOLUB; LOAN, 1996). $\sqrt{\text{SAM}}$ também possui as versões incremental (KAESSE; RANGANATHAN; DELLAERT, 2007) e distribuída (NI; STEEDLY; DELLAERT, 2007). O algoritmo proposto utiliza a mesma estrutura para modelar as relações geométricas entre as posições e tamanhos aparentes do objeto e a postura do robô, e aplica técnicas similares para resolver o problema de LS linearizado resultante.

Finalmente, medições espúrias são tratadas através da identificação dos objetos espúrios no mapa. Alguns objetos ou fragmentos de objetos no ambiente podem ter um aspecto semelhante aos exemplos de uma certa classe, e às vezes são detectados como tal. No entanto, se os detectores de objeto são bem concebidos, o que é uma premissa desta tese, esses objetos não são detectados de forma consistente, de modo que soluções altamente prováveis da associação de dados atribuem poucas medições para eles. Além disso, esses objetos são geralmente detectados apenas a partir de um ponto de vista restrito. Uma vez que uma boa estimativa da localização do objeto só pode ser alcançada se são consideradas medições a partir de vários pontos de vista, esperamos que a incerteza sobre a localização desses objetos espúrios seja elevada. Assim, a probabilidade de um objeto ser espúrio dada uma amostra do par solução da associação de dados e classes de objeto é modelado como dependente da matriz de covariância posterior de sua localização.

1.4 Objetivos

Os objetivos desta tese são:

1. introduzir os O-Maps;

2. definir o O-SLAM, um algoritmo SLAM probabilístico off-line que constrói O-Maps a partir dos dados de saída por detectores especializados de objeto e odômetros do robô; e
3. mostrar que, se as medições forem obtidas dos objetos de interesse, a partir de diferentes pontos de vista, o O-SLAM pode construir O-Maps consistentes e fornecer a trajetória aproximadamente correta do robô para o período em que os dados dos sensores foram coletados.

O algoritmo proposto, O-SLAM, considera que, o robô captura imagens através de uma câmera de vídeo, enquanto explora o ambiente, e tenta detectar determinadas classes de objetos neles. Para outros objetos que não a da presumida classe, as medições enviadas pelo detectores de objeto consiste de objetos com tamanhos e posições aparentes nas imagens. Aqui, os modos assumidos de erro do detector de objeto correspondem à imprecisão dos tamanhos e posições detectados, e detecções indevidas — detecções que não correspondem a um exemplo consistente de uma classe desejada. Assim, o O-SLAM depende de modelos probabilísticos descrevendo os modos de erro do sensor para inferir classes de objeto e construir seus modelos geométricos.

Conforme mencionado na Seção 1.2, a associação de dados é abordada como um problema de amostragem. Em vez de buscar pela melhor solução para a associação de dados, o algoritmo SLAM proposto faz a amostragem do espaço utilizando uma técnica de MCMC. Estendendo os resultados de outros trabalhos, o espaço das atribuições da classe de objeto é também amostrado, acrescentando uma nova dimensão ao problema. As amostras geradas são então processadas a posteriori para construir um mapa único do ambiente.

Neste trabalho, os O-Maps são considerados consistentes caso forem mapeados, com alto grau de certeza, apenas objetos que de fato existam no ambiente. Trajetórias aproximadamente corretas são aquelas que não divergem da verdadeira trajetória no solo, de forma que o erro na localização do robô durante todo o tempo permanece delimitada. O importante dessa definição é que a trajetória do robô reflita a capacidade do algoritmo de reconhecer quando retorna a um lugar onde havia navegado anteriormente. Cabe ressaltar que alcançar esses resultados depende principalmente da utilização de bons detectores de objeto. Por exemplo, se eles não detectarem alguns objetos de interesse nas imagens, ou detectá-los poucas vezes ou a partir de um ponto de vista bem restrito, esses objetos não serão mapeados, ou terão uma alta

probabilidade de serem espúrios.

A fim de demonstrar as capacidades do O-SLAM, o objetivo número 3 é dividido em três sub-objetivos:

demonstrar a consistência do modelo geométrico: se forem fornecidas a correta solução da associação de dados e as atribuições da classe de objeto, e utilizadas somente medições não espúrias, o O-SLAM gera mapas consistentes, incluindo suas localizações e tamanhos no ambiente, e corrige aproximadamente as trajetórias do robô;

demonstrar o sucesso de O-SLAM no fechamento de volta: a abordagem MCMC fornece algoritmos práticos para a resolução do problema da associação de dados com o SLAM baseado em objeto, mesmo quando o robô fecha voltas na trajetória;

demonstrar o efeito da modelagem dos erros nos detectores de objetos: se a solução da associação de dados e as classes corretas de objeto são desconhecidas, mapas consistentes baseados em objeto são possíveis de se obter pela inferência conjunta da informação geométrica e semântica, modelando o problema da detecção imperfeita do objeto.

1.5 Contribuições

Esta tese apresenta duas principais contribuições: 1) a introdução de O-Maps e 2) a definição do algoritmo O-SLAM. O O-Map constitui uma nova representação do ambiente, no contexto da navegação do robô, e apresenta algumas potenciais vantagens quando comparada às tradicionais representações. O O-SLAM é um novo algoritmo SLAM *off-line* que resolve os problemas colocados pela construção de O-Maps, tal como encontrar as classes mais prováveis para os objetos e atribuir uma probabilidade de ser espúrio a cada um. Nenhum outro algoritmo SLAM utiliza apenas a saída dos detectores de objeto e dos odômetros para construir mapas baseados em objetos e localizar o robô de uma só vez.

A primeira contribuição desta tese é a introdução de O-Maps como uma alternativa a mapas de objetos tradicionais para representação semântica e geométrica dos mesmos para tarefas envolvendo navegação do robô e interação com o usuário. Mesmo

existindo outros tipos de mapas que descrevem os objetos no ambiente, a representação proposta é nova no contexto da robótica, uma vez que nenhum outro tipo de mapa informa as posições, tamanhos e classes do objeto de uma só vez, além de representar a incerteza dos parâmetros. O mapa baseado em objeto proposto apresenta algumas potenciais vantagens sobre outros tipos de mapas propostos na literatura, em tarefas relacionadas com a navegação do robô, planejamento do caminho e tarefa, e interação com os seres humanos.

Conforme mencionado na Seção 1.2, a maior dispersão de objetos quando comparado com características geométricas elementares como pontos ou linhas é favorável à localização do robô porque a associação de dados é menos difícil de resolver no caso disperso. Além disso, comparando o mapa baseado em objeto proposto com outros mapas de objetos, propostos na literatura, a primeira permite ao robô lidar melhor com a percepção ambígua, pois os tamanhos aparentes de objeto conduz naturalmente a uma noção de distância do robô aos objetos. O planejamento do caminho e da tarefa tira proveito da informação semântica representada pelas classes de objeto, uma vez que podem ser obtidos mais planejamentos de alto nível. Finalmente, os seres humanos podem se comunicar melhor com os robôs utilizando o vocabulário comum dos objetos, os quais são referidos por seus nomes na linguagem natural.

Outra característica nova dos O-Maps é a representação da probabilidade dos objetos de serem não espúrios. Uma vez que o mapa proposto é construído a partir de diferentes hipóteses sobre associação de dados, o número de objetos pode ser incerto. Aqui, esta incerteza é representada pela atribuição de uma probabilidade marginal a cada objeto no mapa, que tenha sido realmente detectado no ambiente. Como ele envolve a correspondência dos mesmos objetos físicos entre os mapas obtidos com diferentes hipóteses sobre a solução da associação de dados, o algoritmo SLAM proposto define um conceito de identidade do objeto válido entre as diferentes associações. A presença de uma mesma identidade de objeto nos mapas gerados, assumindo a solução mais provável da associação de dados, aumenta probabilidade de ser não espúrio. Além disso, os parâmetros dos objetos com a mesma identidade, obtidos assumindo diferentes soluções da associação de dados, são combinados para definir aqueles parâmetros do objeto no O-Map.

A outra contribuição desta tese é a técnica O-SLAM, que constrói o mapa baseado em objeto proposto a partir de dados de entrada visuais e recupera a trajetória do robô. A novidade da técnica proposta reside na forma como a informação geométrica

e semântica é utilizada em conjunto para a construção do mapa, de tal forma que as geometrias e classes do objeto estão intimamente acopladas. Assim, as classes de objeto influenciam e são influenciadas pelos parâmetros geométricos do mapa. Outra nova característica do O-SLAM é que, em vez de construir um mapa com características e pontuais, o algoritmo proposto constrói mapas constituídos por simples modelos geométricos de objetos no ambiente.

Embora os mapas e trajetórias gerados por O-SLAM não sejam tão geometricamente precisos quanto os gerados por algoritmos de SLAM convencionais, não há perda significativa da informação necessária à tarefa de navegação em ambientes com objetos. O próprio ser humano não conhece as posições e dimensões exatas dos objetos no ambiente, apenas possui uma noção vaga, que o mapa proposto pretende representar. Embora a representação humana possua conteúdo semântico mais complexo — os objetos são representados por uma descrição semântica complexa e vários níveis de localização como o cômodo, o endereço da casa, etc. — apenas as posição relativas e um tamanho imprecisos, além dos rótulos semântico são necessários para representar objetos simples em um nível local.

Entre os procedimentos envolvidos no algoritmo O-SLAM, é importante ressaltar a contribuição específica de como utilizar eficientemente as técnicas MCMC para amostrar o espaço da associação de dados. Embora muito foi inspirado em trabalhos anteriores (KAESS; DELLAERT, 2005; RANGANATHAN, 2008), alguns procedimentos novos são propostos, concebidos para aumentar a eficiência do processo de amostragem e para permitir a amostragem sobre o espaço aumentado das soluções da associação de dados e classes de objeto.

1.6 Organização

O restante desta tese está organizado em três partes. A primeira, compreendendo o Capítulo 2 e o Capítulo 3, introduz os conceitos utilizados aqui e situa este trabalho na literatura relacionada. O Capítulo 2 faz uma breve revisão do campo de mapeamento robótico, introduzindo os principais problemas enfrentados nesta área, os conceitos envolvidos e as mais relevantes soluções propostas. O Capítulo 3 descreve o novo mapa proposto nesta tese, relacionando-o com trabalhos similares na literatura.

A segunda parte, formada pelo Capítulo 4 e o Capítulo 5, constituem a parte técnica do trabalho, onde o algoritmo SLAM baseado em objeto proposto é detalhado.

No Capítulo 4, a solução para inferir no mapa, compreendendo as posições, geometrias e classes dos objetos, bem como a trajetória do robô, é detalhado. Para tal, amostras de distribuição de classes de objetos e soluções da associação de dados são consideradas dadas. O Capítulo 5 descreve a solução proposta para gerar tais amostras utilizando a abordagem MCMC, e discute também algumas questões para implementação prática.

Finalmente, o Capítulo 6 e o Capítulo 7 formam a última parte deste trabalho, onde os resultados e as conclusões da técnica proposta são discutidos. O Capítulo 6 descreve os resultados dos experimentos realizados com imagens capturadas por robôs reais, utilizando um detector de objeto simples e algoritmo segmentado. As conclusões são apresentados no Capítulo 7, o qual discute também possíveis desdobramentos do presente trabalho.

2 Mapeamento Robótico

Este capítulo situa este trabalho no campo de investigação do mapeamento robótico. Ele tem sido um campo muito ativo durante as últimas três décadas, e ainda está em contínuo desenvolvimento. Uma vez que existem muitas aplicações que envolvem mapeamento por robôs, muitos ramos neste campo de pesquisa têm sido propostos, investigando aspectos específicos da área. Assim, o objetivo deste Capítulo é especificar os ramos que melhor se adequam a esta tese, e para os quais ela mais contribui.

Mapeamento é o nome dado à tarefa de construir representações do ambiente com dados obtidos a partir dele. Este tipo de tarefa é comum em várias atividades humanas, como a arquitetura, construção civil, cartografia, aviação, geografia, geologia, atividades militares, e outras. Quando se trata de mapeamento robótico, este termo corresponde à tarefa de construir um mapa ambiental, de forma autônoma, por um robô móvel, utilizando seus sensores para esta finalidade. Neste caso, o robô processa os dados de seu sensor, extraindo informações relevantes, e registra-os no mapa.

Os mapas expressam a informação que eles contêm atribuindo valores aos seus elementos. A natureza desses elementos varia dependendo do tipo de mapa, e pode corresponder a uma em um amplo espectro de possíveis estruturas. Assim, a tarefa de mapeamento corresponde à atribuição de valores aos elementos do mapa, relacionando cada um a uma certa posição nele. Mapas utilizados em robótica podem receber nomes diferentes de acordo com os elementos que eles usam para descrever o ambiente. Na Seção 2.1, algumas das classes mais comuns de mapas são discutidas. A Seção 2.2 discute o processo de mapeamento em robótica, enfatizando os que não assumem um conhecimento preciso da localização do robô.

Figura 2.1: Classificação dos tipos mais comuns de mapas utilizados em robótica, de acordo com seu nível de abstração. (a) Mapa grade de ocupação, representando mapas métricos, no mais baixo nível (b) Mapa de características de pontos, indicando a localização das árvores e da trajetória do robô, extraído de (SELVATICI; COSTA, 2007a) (c) Mapa topológico, que interliga marcos terrestres com os comandos de controle utilizados para se deslocar de um para qualquer outro, extraído de (VASSALLO, 2004).

2.1 Tipos de Mapas

Vários tipos de mapas foram propostos na literatura robótica. O tipo ideal de mapa a ser utilizado ou construído por um robô móvel depende da tarefa e do ambiente onde este esteja inserido. Também depende das características do robô, tais como os tipos de sensores que ele tem, bem como a forma com ele se move.

Em qualquer caso, é possível classificar os tipos básicos de mapas de acordo com o nível de abstração dos elementos utilizados na sua construção. A Figura 2.1 ilustra esta classificação, tendo por base a taxonomia de mapas proposta por Dudek & Jenkin (2000). Mapas métricos são aqueles no mais baixo nível de abstração, e seus elementos correspondem a células indicando o valor de alguma característica local. Mapas topológicos ou relacionais estão no outro extremo, apresentando o mais alto nível abstração. Representam lugares ou objetos e conecta-os por algum tipo de relacionamento, não concernente à sua localização precisa. Mapas geométricos ou de características estão em um nível intermediário abstração, uma vez que representam a localização de algumas características presentes no ambiente, e portanto não concernente em atribuir um valor para cada ponto no mapa.

Apesar do fato de que existe uma grande variedade de tipos de mapas que podem ser construídos, cada um adequado a uma determinada tarefa ou ambiente, há uma distinção muito clara entre o processo de construção de mapas métricos e topológicos.

No caso da construção de mapas métricos, o objetivo é obter um mapa detalhado

do ambiente, com informações sobre a forma e tamanho dos objetos e os limites das áreas livres para a navegação, tais como corredores, quartos, trilhas e estradas. Para representar essa complexa e detalhada informação, o mapa é geralmente dividido em uma densa rede de forma que cada célula contenha informações sobre a ocupação desse espaço por um objeto e, possivelmente, outras características ambientais que estão sendo mapeadas.

No caso de mapas topológicos, o objetivo é construir uma estrutura relacional, normalmente um grafo, relacionando determinados locais no ambiente e objetos definidos ou marcas, mesmo que sejam naturais ao ambiente ou construído com o propósito de ser mapeado. É fácil perceber que as informações contidas no mapa topológico é menos detalhado e distribui-se de forma dispersa, concentrando-se apenas em alguns pontos de interesse. Assim, é muito seletivo com relação à informação que ele registra.

A diferença expressiva entre mapas métricos e topológicos muitas vezes leva a soluções práticas que tentam conciliar elementos métricos em mapas relacionais, ou vice-versa, numa tentativa de obter vantagens das diferentes abordagens, quer através da construção de mapas híbridos (NIETO; GUIVANT; NEBOT, 2004; BLANCO; FERNANDEZ-MADRIGAL; GONZALEZ, 2008) ou utilizando múltiplos níveis de abstração para representar diferentes aspectos do ambiente (DUDEK, 1996).

2.1.1 Mapas métricos

O objetivo dos mapas métricos é representar o ambiente com um alto nível de detalhes, codificando informação em relação ao tamanho e forma dos objetos e os limites das áreas livres para a navegação. Para tal, os mapas métricos dividem o espaço 2D ou 3D em células regulares, as quais podem representar qualquer característica local de interesse, funcionando, por exemplo, como uma planta de uma sala ou pavimento de edifício, indicando os objetos e limitando as áreas onde o robô pode navegar. Um exemplo clássico de mapa métrico é o mapa de grade de evidência (ELFES, 1987), que representa o ambiente como uma matriz de forma que cada célula contém a probabilidade de que esse espaço seja ocupado por um obstáculo para o robô. Este conceito foi estendido para mapas de evidência 3D, que dividem o espaço 3D em células representando a probabilidade de ocupação (MORAVEC, 1996). Um exemplo menos comum de mapa métrico é o mapa de aspecto visual, o qual representa a imagem capturada pelo robô para cada postura que este possa assumir. Neste caso, cada célula do

mapa representa diretamente a informação sensorial que o robô percebe, em vez de uma característica do local do ambiente que não depende da percepção do robô. Um exemplo desse mapa é o proposto por Porta & Kröse (2006), onde as imagens são codificadas pelo Transformada Discreta do Cosseno.

Em geral, os mapas métricos exigem um grande espaço na memória e cada etapa da atualização no processo de mapeamento é computacionalmente custosa, pois envolve o processamento de várias células a cada vez. Adicionalmente, mesmo se cada interação do mapeamento for simples e envolver poucas células por vez, a construção completa do mapa requer operações de interação sobre cada célula. No entanto, como o mapa métrico salva informações no próprio domínio do sinal do sensor, a tarefa de localização é executada de uma forma simples. Além disso, devido à alta densidade de informação nesses mapas, o resultado da localização é mais preciso e menos sujeito a ambigüidades.

2.1.2 Mapas topológicos

No outro extremo da taxonomia apresentada na Figura 2.1 existem mapas topológicos, também chamados de mapas relacionais, que fornecem as relações entre os vários locais ou objetos presentes no ambiente, geralmente sob a forma de um gráfico. Mapas topológicos, de forma geral, registram informações sobre determinados elementos ou locais do ambiente, chamados marcos. Assim, marcos e relações são os elementos dos mapas topológicos.

Essas relações podem ser de vários tipos. Eles podem indicar o deslocamento relativo entre dois marcos, a existência de um caminho entre eles, os comandos de controle que movem o robô de um marco para outro, rótulos semânticos mostrando instruções de alto nível para ir de um ao outro, e assim por diante. Exemplos de mapas topológicos incluem a obra seminal de Kuipers (1978) sobre o modelo TOUR, que consiste de rótulos semânticos indicando a estratégia de alto nível, para uma pessoa ou robô, para ir de um marco ao outro; Mapas Probabilísticos Topológicos (PTMs) (RANGANATHAN; DELLAERT, 2004), que mostram marcos métricos e os caminhos entre eles; e mapas motor-visuais (VASSALLO, 2004), que descrevem marcos por uma característica panorâmica da imagem, e os ligam entre si pelas arestas anotadas com comandos de controle de orientação que leva o robô de um para o outro.

Mapas topológicos exigem muito menos memória quando comparado aos mapas

métricos. Por esta razão, eles são os mais apropriados para o mapeamento de grandes áreas. Por outro lado, mesmo se uma interação de qualquer algoritmo que constrói mapas topológicos deve processar alguns nós de um grafo, o processo de mapeamento é um pouco mais complexo. Este fato é devido à necessidade de processar os dados do sensor do robô de uma forma mais abstrata, a fim de extrair informação sobre as localidades mais relevantes que devem ser consideradas como marcos. Além disso, deve ser realizado um processamento extra para determinar as relações entre os marcos.

De fato, há um compromisso entre a informação complexa contida em mapas relacionais e a grande quantidade de elementos que precisam ser registrados nos mapas métricos. Um mapa contendo relações simples pode exigir um tratamento simples dos dados, mas exige também um maior número de elementos para fornecer informações suficientes. Num caso limite, temos mapas métricos, em que as relações entre os elementos do mapa são desprezíveis e a informação que eles contêm é tão pouca quanto a característica do local. No outro caso limite, quando devem ser representados muito poucos lugares no ambiente, mas as informações sobre eles e as relações entre os lugares tornam-se muito complexas, temos os mapas puramente relacionais.

2.1.3 Mapas de características

Os extremos representados pelas abordagens métrica e topológica conduzem ao uso de soluções intermediárias. Uma possibilidade corresponde aos mapas de características, que representam algumas características especiais do ambiente, geralmente representado por elementos geométricos. A Figura 2.2 mostra exemplos deste tipo de mapa. Diferentemente dos mapas puramente métricos, o espaço ambiental não é plenamente representado pelos mapas de características, mas apenas algumas estruturas especiais, tais como pontos, paredes fronteiriças, ou objetos, como no caso deste trabalho. Por outro lado, as características representadas tem sua posição absoluta atribuída, bem como outras eventuais informações relacionadas a elas, não representando as relações entre as diferentes características.

Apesar de serem classificados como um tipo de mapa métrico em geral, mapas de características têm fortes semelhanças com mapas topológicos, especialmente no que concerne ao respectivo processo de mapeamento. Enquanto no mapeamento topológico, os lugares onde o robô estiveram são conectados através de relações que indicam como se dá a locomoção de um lugar a outro, no mapeamento de caracte-

Figura 2.2: Exemplos de mapas de características geométricas. (a) Mapa de linhas de teto, extraído do (FOLKESSON; JENSEFELT; CHRISTENSEN, 2005). Mapa de planos texturizados, extraído de (THRUN et al., 2004).

rísticas a localização daquelas que o robô viu são encontradas relacionando-as com os locais visitados pelo robô quando essas características foram vistas. Colocado de forma diferente, o mapeamento topológico envolve a construção de um grafo interligando os locais por onde o robô passou, e o mapeamento de características envolve a construção de um grafo interligando as características detectadas pelo robô aos locais em que foram detectadas por ele (DELLAERT, 2005).

As características mais comuns utilizadas nos mapas para navegação de robô são pontos especiais detectados no ambiente. Características pontuais 2D ou 3D foram utilizados em algumas trabalhos seminais (SMITH; SELF; CHEESEMAN, 1990; LEONARD; COX; DURRANT-WHYTE, 1992; DISSANAYAKE et al., 2001; SE; LOWE; LITTLE, 2002; DAVISON, 2003) e desde então, têm sido amplamente adotados. Eles conduzem pouca quantidade de informação, especialmente quando detectados por sensores de alcance como os sonares ou varreduras de laser, uma vez que informam somente as posições pontuais no plano de movimentação do robô ou no espaço. Assim, muitos pontos característicos são necessários, para descrever o ambiente com detalhes suficientes, para tarefas como a navegação do robô. Se forem detectados pontos nas imagens capturadas pelo robô, alguns descritores locais de imagem podem também ser atribuídos às características mapeadas.

Linhas e planos são outras características geométricas comuns utilizadas no mapeamento robótico (PFISTER; ROUMELIOTIS; BURDICK, 2003; THRUN et al., 2004; FOLKES-

SON; JENSFELT; CHRISTENSEN, 2005). Eles descrevem estruturas mais complexas do que pontos, exigindo menos espaço de memória para representar a mesma área. Por exemplo, uma parede num corredor pode ser descrito utilizando apenas um segmento de reta — em mapas 2D — ou pedaço de plano — em mapas 3D —, enquanto vários pontos seriam necessários para fazer o mesmo. No entanto, utilizar linhas ou planos para descrever o ambiente requer a detecção de estruturas mais complexas e correspondê-las entre diferentes varreduras ou imagens. Assim, há um compromisso entre utilizar características mais simples ou mais complexas, de modo análogo ao compromisso existente entre mapas métricos e topológicos.

2.1.4 Mapas de objetos

Recentemente, os objetos também foram considerados como características em mapas de navegação de robô. Apesar do reconhecimento e reconstrução de objetos geométricos no ambiente já ter sido um tópico de investigação em alguns trabalhos isolados — por exemplo, (ALLEN; LEGGETT, 1995) —, esse tópico não entrou no cerne da pesquisa robótica até o desenvolvimento de técnicas robustas de reconhecimento de objeto por visão computacional. Em contraste com elementos geométricos simples, os objetos incorporam informação semântica, aumentando consideravelmente as aplicações potenciais dos mapas.

Os trabalhos de Vasudevan et al. (2007) e Ranganathan & Dellaert (2007) são exemplos de mapas de características que modelam os objetos no ambiente. Enquanto o primeiro trabalho representa a posição dos objetos detectados em grandes espaços interiores com relação a um sistema de coordenadas local, o último incrementa a representação dos objetos com o parâmetro tamanho, embora apenas os objetos representados em uma única imagem sejam modelados. Apesar da imagem poder ser o resultado de um único disparo de uma câmera, ou uma panorâmica resultante do alinhamento de imagens com objetos não sobrepostos, o método proposto por Ranganathan & Dellaert (2007) não pode construir um único mapa a partir de várias imagens de uma grande área com objetos sobrepostos.

No entanto, muitos dos trabalhos relacionados com o mapeamento baseado em objetos fiam-se em estruturas auxiliares para representar o ambiente, usando objetos para incrementá-los. Ekvall, Kragic & Jensfelt (2007) utiliza objetos reconhecidos nas imagens para incrementar um mapa de linhas características, enquanto Forssen et al. (2008) utiliza um mapa de grade de ocupação. Ambos os trabalhos representam os

objetos por sua identidade e localização reconhecidos no mapa.

Alguns trabalhos apresentam diferentes graus de dependência de outras estruturas de mapa. Galindo et al. (2005) descreve uma representação hierárquica que utiliza descrições de objetos e mapas métricos como uma camada perceptível, subordinada a locais rotulados semanticamente, que correspondem à camada cognitiva. Os objetos são descritos utilizando características de baixo nível simples necessárias para reconhecê-los nas imagens. Os trabalhos de Limketkai, Liao & Fox (2005) e Friedman, Fox & Pasula (2007) nem mesmo necessitam de representações visuais de objetos. Em ambos os trabalhos, respectivamente, objetos e locais são reconhecidos em mapas métricos pré-construídos, utilizando redes de crenças discriminativas.

No entanto, nenhum dos trabalhos apresentados trata da representação imprecisa dos objetos. É possível que o método para detectar e identificar objetos no ambiente ou em mapa pré-elaborado não seja plenamente confiável, de forma que uma medida de confiabilidade a respeito dos objetos mapeados é desejável. O mapa de objetos proposto no presente trabalho, o O-Map, permite a atribuição de um grau de confiabilidade a cada objeto detectado no ambiente.

O O-Map também utiliza somente objetos como características, sem depender de qualquer outra estrutura para representar o meio ambiente. Analogamente ao trabalho de Ranganathan & Dellaert (2007), os objetos são representados com suas localizações 3D, tamanhos e classes. No entanto, o tamanho dos objetos neste trabalho compreende os parâmetros de algum volume geométrico, aumentando a dimensionalidade do problema do mapeamento. Além disso, como no trabalho de Vasudevan et al. (2007), o mapa proposto pode representar os objetos presentes em grandes áreas.

2.2 O problema de SLAM

Uma vez que a tarefa do mapeamento consiste em relacionar informações extraídas do ambiente, com uma localização no mapa, os erros de mapeamento correspondem ao acúmulo de erros sensoriais e da localização do robô. A tarefa de mapeamento não representa um problema difícil quando a história da localização do robô é conhecida de forma precisa (THRUN, 2002). Neste caso, o erro de mapeamento está intrinsecamente ligado ao erro dos sensores que capturam informações do ambiente. Como as medições destes sensores, em dois momentos distintos, são geralmente independentes, os erros de mapeamento são limitados. O mesmo se aplica aos casos

em que o erro na localização do robô é limitado por um valor aceitável.

Por outro lado, quando a localização do robô é informada com erros ilimitados e cumulativos, os erros de mapeamento tornam-se ilimitados também. Este é o caso em que a localização é estimada por *dead reckoning*, uma técnica que calcula a localização do robô integrando apenas os dados dos sensores internos, que detectam deslocamentos sem nenhuma referência a um elemento externo (DUDEK; JENKIN, 2000).

Neste caso, a fim de obter o mapa, é necessário corrigir o erro de localização utilizando dados fornecidos pelos sensores externos. No entanto, isso corresponde exatamente à tarefa de localização, que requer a construção de um mapa anterior do ambiente. Como a execução de duas tarefas sozinhas conduz a erros que são acumulados ao longo do tempo, o mapeamento e a localização devem ser realizadas em conjunto, o qual conduz ao problema SLAM.

Esta Seção aborda o problema SLAM de uma forma mais pormenorizada, compreendendo uma revisão bibliográfica sobre o tópico. Embora a pesquisa em mapeamento robótico tenha sido ativa desde a década de 70, pelo menos, o primeiro tratamento apropriado para o problema SLAM data de meados da década de 80, com o trabalho seminal de Smith & Cheeseman (1986), que apresenta o desenvolvimento de uma estrutura probabilística para tratar das relações de incerteza espacial envolvendo as posições do robô e marcos detectados no ambiente.

2.2.1 Problemas relacionados

O complexo problema do SLAM pode ser abordado de acordo com seus diferentes sub-problemas relacionados (THRUN, 2002). O primeira corresponde ao problema da estimação: os erros dos sensores internos e externos contribuem para o erro de ambos, mapeamento e localização, durante o processo de mapeamento, os quais tornam-se dependentes um do outro. Assim, esses erros devem ser resolvidos de uma forma conjunta, pela inferência mútua da localização do robô e dos parâmetros do mapa, caracterizando o problema da localização e do mapeamento simultâneos.

Por outro lado, a solução do problema da estimação apresenta um outro: o problema da associação de dados, mencionado na Seção 1.2 para o caso de mapeamento de características. Quando do problema da localização do robô somente, onde um mapa do ambiente está disponível, o problema da associação de dados surge quando o robô tem de decidir a quais elementos do mapa correspondem à percepção

de seus sensores. No problema SLAM, esse problema é agravado pelo uso de um mapa incompleto para localizar o robô. Neste caso, o robô tem ainda de decidir se a sua percepção de fato corresponde a elementos que já existem no mapa, ou se novos elementos devem ser adicionados a ele.

Outro problema que afeta diretamente os problemas do mapeamento e da localização em geral, mas torna-se crítico no caso de SLAM, é o problema da dimensionalidade. Enquanto o número de elementos do mapa a serem calculados pode ser muito grande — por exemplo, da ordem de milhões de células em alguns mapas métricos — a aquisição conjunta do mapa e localização demanda um maior esforço computacional do que a execução de uma das duas tarefas individualmente. Assim, um dos desafios relacionados com o desenvolvimento de uma técnica para resolver o problema SLAM é manter a sua complexidade computacional tratável.

O problema SLAM torna-se ainda mais difícil quando um ambiente dinâmico for considerado. Quando permitimos mudanças ambientais durante o processo de mapeamento, são permitidas novas explicações para os dados percebidos. Isto torna o problema da associação de dados ainda mais difícil de resolver. Por exemplo, se o robô obtém dados inesperados do sensor, ele deve decidir se é devido ao erro de localização, ao erro do sensor, a um local não visitado ou a uma mudança de estado do ambiente. Este trabalho assume que os objetos sendo mapeados estão estáticos. De fato, a maioria das modernas técnicas SLAM ainda assumem um ambiente estático.

Além de todos os problemas envolvidos no processamento de dados, existe também o problema relacionado com o modo como esses dados são obtidos. Como o mapa ambiental não é conhecido inicialmente, não é possível realizar um plano prévio a ser usado pelo robô para explorar o ambiente. Assim, a exploração autônoma só pode ser obtida programando os comportamentos do robô, que são módulos de processamento que mapeiam as percepções do robô para execução de uma ação (MURPHY, 2000). É desejável que os comportamentos exploratórios sejam propostos, de forma que a maneira como os dados são adquiridos faz com que a solução do problema SLAM seja mais fácil.

Em geral, as técnicas SLAM não apresentam uma solução integrada para a exploração do ambiente, mas técnicas exploratórias são baseadas em algum método SLAM, a fim de otimizar os comandos de movimento, bem como partes da trajetória, de forma a minimizar erros de mapeamento ou de acelerar a construção do mapa (STACHNISS; GRISETTI; BURGARD, 2005; KOLLAR; ROY, 2006). Portanto, o tema da exploração

no problema SLAM é tradicionalmente tratado separadamente dos outros. Tal como no problema do ambiente dinâmico, este trabalho não trata do tema da exploração em SLAM.

2.2.2 Técnicas de estimação

Embora a literatura apresente diferentes modelos probabilísticos para a estimação de variáveis no problema SLAM, virtualmente todos eles podem ser derivados de uma formulação de âmbito geral, utilizando elementos de um *Hidden Markov Model* (HMM). A teoria envolvendo HMMs é tradicionalmente utilizada em processamento de sinais, como é o caso do problema de reconhecimento da escrita e da fala (RABINER, 1989). A literatura sobre esses problemas está bem estabelecida, e várias técnicas têm sido propostas. Entretanto, devido principalmente às diferenças no domínio de aplicação, existem poucos exemplos da utilização direta das técnicas HMM para o mapeamento e localização do robô. Neste contexto, as técnicas mais bem conhecidas são a *Markov Localization* (FOX, 1998) e a utilização de *Expectation Maximization* (EM) para SLAM, no trabalho de Thrun, Fox & Burgard (1998). Todas estas técnicas adotam o mesmo modelo geral para o sistema, no qual um espaço discreto para a localização do robô e o mapa é assumido, e não há restrições envolvendo a distribuição de probabilidade das variáveis ou o comportamento do movimento e modelos de observação. Assim, a complexidade computacional destes algoritmos é elevada em comparação com técnicas afins, os quais assumem algumas simplificações no domínio do problema.

Por razões históricas e garantias teóricas (DISSANAYAKE et al., 2001), uma das mais populares técnicas de estimação em SLAM é o *Extended Kalman Filter* (EKF), uma técnica de inferência aplicável a determinados HMMs com espaço de estados contínuo, introduzido ao contexto SLAM pelos trabalhos seminais de Smith, Self & Cheeseman (1990) e Leonard, Durrant-Whyte & Cox (1990), e inspirando uma grande parte dos trabalhos neste campo. O EKF é usado para manter uma estimativa da localização atual do robô e as localizações de algumas características geométricas, na maioria das vezes pontos. No entanto, o uso do EKF para resolver o problema da estimação em SLAM levanta algumas limitações. Apesar de alguns deles terem sido superados ou atenuados por modificações posteriores, outros são intrínsecos à representação utilizado pelo EKF e, portanto, requer alterações mais profundas para ser resolvido. Exemplos de tais limitações são o pressuposto de que as incógnitas são bem modeladas por variáveis gaussianas, a aproximação satisfatória do movimento

do robô e modelos de observação em suas versões linearizadas, e a falta de uma solução integrada para o problema da associação de dados.

Todas as limitações citadas podem ser pensadas como problemas de representação. Para superar essas limitações, vários algoritmos SLAM foram criados com base em amostragem, que corresponde a uma representação mais geral para uma distribuição de probabilidade. *DP-SLAM* (ELIAZAR; PARR, 2003; ELIAZAR; PARR, 2004) utiliza um filtro de partículas para manter a crença sobre a localização do robô e um mapa de grade de evidências, com o auxílio de uma estrutura especial denominada árvore de ancestralidade. *FastSLAM* (MONTEMERLO et al., 2002) usa um filtro de partículas *Rao-Blackwellizadas* para manter as amostras durante toda a trajetória do robô. De fato, cada amostra representa mais que uma trajetória, mas também pequenos EKFs associados a cada trajetória. Em ambos os casos, cada amostra assume uma solução diferente para a associação de dados, de modo que estes métodos apresentam uma forma integrada para resolver o problema de associação de dados.

No entanto, filtros de partículas também apresentam inconvenientes. Mesmo se não forem utilizados na localização e mapeamento em tempo real, eles podem sofrer a degeneração das partículas, um problema que impede o filtro de amostrar corretamente o suporte da distribuição de probabilidade alvo. Se for utilizado para aplicações em tempo real, mantendo o número suficiente de amostras, pode consumir mais tempo computacional por etapa de atualização do que o disponível. Além disso, as técnicas utilizadas para evitar o problema da degeneração das amostras exigem um tempo adicional de processamento.

Outra importante limitação das técnicas de filtragem em geral, é que nenhuma revisão das decisões anteriores é dada. Por definição, técnicas de filtragem inferem o estado atual de um HMM a partir dos dados disponíveis (CAPPÉ; MOULINES; RYDÉN, 2005), e usam apenas novos dados para atualizar a crença sobre o estado atual da localização do robô e do mapa. Portanto, a solução ou soluções assumidas para a associação de dados, armazenadas em amostras mantidas por técnicas de filtragem, são só atualizadas, mas não podem ser revistas. Uma maneira de superar esta limitação é inferir toda a história da localização do robô e do mapa de uma só vez, realizando otimização por lote para encontrar a melhor a solução ótima ou quase ótima. Técnicas que estimam a história dos estados de um HMM são chamadas de técnicas de smoothing (CAPPÉ; MOULINES; RYDÉN, 2005). Quando o problema SLAM é tratado como um problema de otimização por lote, ele é comumente designado por SLAM

completo, SLAM *off-line* ou SAM, acrônimo para *Smoothing and Mapping* (DELLAERT, 2005), e constitui um problema distinto da maioria do SLAM *on-line*, mais comumente tratado na literatura. Embora os métodos de SLAM *off-line* não possam ser aplicados para a localização e mapeamento em tempo real, em geral, ele permite o ajustamento em lote do mapa e trajetória, gerando resultados mais precisos.

Exemplos de técnicas SLAM *off-line* são o trabalho de Lu & Milios (1997), que propôs uma solução específica para o alinhamento de varredura de laser, estendido pelo trabalho de Gutmann & Konolige (1999), que propôs uma técnica para fechar grandes voltas no mapa. Folkesson & Christensen (2004) propôs a técnica *Graph-SLAM*, que realiza otimização do tipo descida pelo gradiente no espaço conjunto das posturas do robô, localização das características e soluções da associação de dados. Dellaert (2005) propôs a técnica $\sqrt{\text{SAM}}$, um algoritmo de estimação eficiente para o problema de SLAM completo, cuja versão *on-line* pode substituir os algoritmos de filtragem baseados no EKF, mais comumente utilizado, com grande vantagem (KAESS; RANGANATHAN; DELLAERT, 2007).

De fato, a formulação original do EKF coloca sérias questões quando aplicada em estimativa no problema de SLAM. Uma vez que a complexidade computacional da formulação original do EKF cresce com o quadrado do número de características no mapa, algumas variações têm sido propostas considerando explicitamente um modelo gráfico do problema de SLAM e atualizando-o em tempo constante, assumindo algumas aproximações realistas (PASKIN, 2003; LIU; THRUN, 2003). Por sua vez, SAM incremental (iSAM), a versão on-line do algoritmo de $\sqrt{\text{SAM}}$, é capaz de atualizar o mapa de forma exata em tempo constante, porém ele precisa executar $\sqrt{\text{SAM}}$ de tempos em tempos, o que é executado, na prática, com complexidade $O(n^2)$ no número de características (KAESS; RANGANATHAN; DELLAERT, 2007). No entanto, as experiências mostram que esta abordagem é perfeitamente adequada para processar uma grande quantidade de dados em tempo de execução (KAESS, 2008).

2.2.3 Problema de associação de dados

Muitas técnicas foram propostas para resolver o problema da associação de dados em SLAM. O principal fator que muito influencia no sucesso de uma determinada técnica é a densidade do mapa sob construção. Em geral, uma distinção entre os problemas de associação de dados local e global é adequada para a construção de mapas métricos ou de características densamente povoadas, tanto em SLAM *on-line*

quanto em SLAM *off-line*. Neste caso, a associação de dados local corresponde a manter o correto alinhamento de consecutivas leituras do sensor, e a associação global de dados corresponde a corrigir erros grosseiros e fechar grandes voltas na trajetória, geralmente desencadeado pela detecção inconsistente dos sensores. No caso da construção de mapas de características esparsas e mapas topológicos, normalmente não é feita nenhuma distinção. Uma notável exceção é o DP-SLAM (ELIAZAR; PARR, 2003; ELIAZAR; PARR, 2004), que constrói mapas de grades de ocupação usando varreduras de laser e não faz distinção entre associação local e global de dados.

Técnicas comuns para o alinhamento local dos dados dos sensores são o método Iterative Closest Point (ICP) (ZHANG, 1994), o método de correspondência por varredura de laser proposto por Lu & Milios (1997), e o uso do método *Random Sample Consensus* (RANSAC) (FISCHLER; BOLLES, 1981) para corresponder pontos característicos detectados em imagens consecutivas pelo modelo de câmera (SE; LOWE; LITTLE, 2002). No caso da associação global de dados, especialmente na construção de mapas métricos, explorar o espaço de estado da localização do robô (GUTMANN; KONOLIGE, 1999) pode resolver inconsistências do sensor; no entanto, estruturas auxiliares mais abstratas são utilizadas para melhorar a eficiência do fechamento de volta (HÄHNEL et al., 2003; ELIAZAR; PARR, 2003; BLANCO; FERNANDEZ-MADRIGAL; GONZALEZ, 2008).

Quando se trata de mapeamento de características e mapeamento topológico, a associação de dados assume a forma de um problema de busca. Em SLAM *on-line*, este problema consiste em encontrar a melhor característica mapeada para associá-lo a uma nova medição dada para: quando o robô faz uma nova medição, ele tem que decidir se a medida foi gerada por uma característica no mapa ou por uma não mapeada, ou mesmo se ele corresponde a uma medição espúria. Em SLAM *off-line*, a associação de dados consiste em agrupar as medições disponíveis em conjuntos, de modo que as medições pertencentes ao mesmo conjunto são consideradas como tomadas a partir da mesma característica.

Muitos dos métodos de SLAM *on-line* que constroem mapas de característica consideram abordagens heurísticas para o problema da associação de dados. Os mais comuns são o *Nearest Neighbor* (NN) e *Maximum Likelihood* (ML). Na abordagem NN, cada característica no mapa é projetado no espaço das medições e comparada com a nova medição tomada pelo robô. Em seguida, a característica projetada que mais se aproxima da nova medição é associada a ela. Se nenhuma característica aproxima-se

da nova medição suficientemente bem, esta é usada para mapear uma nova característica ou é considerada uma medição espúria. Na abordagem ML, a nova medição é associada a cada característica mapeada, calculando a verossimilhança global do mapa para cada associação testada. A associação que produziu a maior verossimilhança é assumida como verdadeira. Se a verossimilhança permanece abaixo de um certo limiar, a medição é considerada como sendo gerada por uma nova característica (DISSANAYAKE et al., 2001). *Joint-Compatibility Branch and Bound* (JCBB) estende a abordagem ML testando grupos de novas medições com os grupos de características no mapa, melhorando o alinhamento das medições realizadas sucessivamente (NEIRA; TARDOS, 2001).

Esses métodos heurísticos estão sujeitos a erros quando a incerteza sobre a localização do robô aumenta, pois, neste caso, a principal informação usada para associar novas medições e características mapeadas torna-se pouco confiável. Adicionalmente, estes métodos lidam com o problema da associação de dados de um modo local: a associação é decidida com base na análise dos pares candidatos, mas nenhuma informação adicional é extraída a partir do resto do mapa.

Por outro lado, as técnicas de SLAM *off-line* permitem a reparação de associações anteriores, uma vez que ele processa todos os dados num modo de lote. No entanto, existem muito poucos trabalhos na literatura que realizam a revisão destes erros utilizando a fórmula exata para o problema da associação de dados, exigindo a exploração do espaço das suas possíveis soluções, que cresce em complexidade de forma exponencial com o número de medições. As únicas soluções práticas que executam esta exploração de forma consistente, são baseados em métodos MCMC.

Dellaert (2001) introduziu técnicas MCMC ao problema da associação de dados restrito em *Structure From Motion* (SFM), a versão da visão computacional para o problema de SLAM. Nesse trabalho, um objeto complexo de modelo, descrito pelas posições de pontos característicos é reconstruído com base em várias imagens processadas a fim de detectar tais pontos. Em seguida, o MCMC é utilizado para amostrar as possíveis associações entre os pontos detectados e os reconstruídos, produzindo várias reconstruções do objeto, cada uma obtida com uma dada amostra da solução da associação de dados. Então, os objetos obtidos são combinados para formar uma reconstrução marginal do objeto, independente de uma dada solução da associação de dados.

Por sua vez, Ranganathan & Dellaert (2004) e Kaess & Dellaert (2005) utilizaram

o MCMC para tratar o problema da associação de dados não restritos, mais comum em SLAM robótico, e mais difícil de resolver. O problema da associação de dados não restritos corresponde ao particionamento do conjunto de medidas obtidas pelo robô, em subgrupos, onde cada subgrupo corresponde a uma certa característica física detectada pelo robô. No problema da associação de dados não restritos, o número de características é desconhecida inicialmente, e o espaço das possíveis soluções é ainda maior. A partir da teoria dos conjuntos, o número de partições possíveis em um conjunto com K elementos é dado pelo número de Bell $\mathfrak{B}(K) = \sum_{N=1}^K S(K, N)$, onde $S(K, N)$ é o número de Stirling do segundo tipo, correspondente ao número das possíveis formas de particionar M elementos em N sub-conjuntos (ROTA, 1964). Portanto, o número de possíveis soluções da associação de dados aumenta exponencialmente com o número de medições a associar.

Aqui, o problema da associação de dados não restritos é tratado, seguindo a abordagem proposta por Ranganathan & Dellaert (2004), Kaess & Dellaert (2005). No entanto, este trabalho estende a utilização da amostragem MCMC para resolver o problema SLAM baseado em objeto, o qual envolve a resolução do reconhecimento do objeto, a fim de construir mapas baseados em objetos.

3 O-Map

Este capítulo introduz o mapa de objetos proposto nesta tese, a saber, o O-Map. Mesmo que os conceitos envolvidos na definição do mapa sejam simples, a proposta de representação ambiente é nova no contexto da robótica m. Na verdade, a proposta de mapas baseados em objetos que misturam informações geométricas e semânticas e técnicas para a sua construção é um tema recente, motivado pelo desenvolvimento de algoritmos práticos para o reconhecimento de objetos em imagens e segmentação semântica de mapas métricos.

O mapa proposto codifica descrições geométricas e semânticas dos objetos no ambiente, bem como a incerteza crença sobre a existência desses objetos no mundo real. A Figura 3.1 ilustra a estrutura do mapa proposto. O O-Map é composto por um conjunto de objetos, descritos por rótulos semânticos associados a simples volumes geométricos, como esferas, cilindros ou pirâmides. O uso de elementos geométricos simples têm como objetivo representar as dimensões dos objetos de forma vaga, em analogia à representação que o próprio ser humano faz dos objetos. Por exemplo, quando alguém descreve as dimensões de um vaso, descreve sua altura e seu diâmetro, mas não se preocupa em detalhar a sua forma tridimensional. Cada objeto também possui um rótulo semântico, expresso em linguagem natural, para descrever a que classe o objeto pertence, isto é, se se trata de um relógio, uma mesa, uma cadeira, etc. Cada objeto tem também uma probabilidade de detecção associada, o que indica o grau de certeza de que ele realmente existe no ambiente, em vez de ter sido construído a partir de medições espúrias.

Matematicamente, o O-Map é definido por $M \triangleq \{o_j\}_{j=1}^N$, um conjunto de N objetos o_j . Cada objeto $o_j \triangleq (l_j, g_j, c_j)$ é descrito por sua localização 3D $l_j \triangleq (x_j, y_j, z_j)$, geometria g_j e rótulo semântico de classe c_j . A geometria g_j do objeto descreve os parâmetros do volume geométrico que o representa. Por exemplo, no caso de os objetos serem representados por paralelepípedos, g_j são vetores tridimensionais compreendendo comprimentos, larguras e alturas dos objetos. O rótulo de classe c_j

Figura 3.1: Exemplo de O-Map, tal como proposto nesta tese. Neste exemplo, os objetos são representados por paralelepípedos, descritos por sua largura, altura e espessura. O plano verde representa o chão, que não faz parte do mapa. Transparência na representação do objeto indica incerteza quanto à sua existência.

deve corresponder a um elemento tirado de um conjunto predefinido de possíveis rótulos. Cada objeto no mapa também possui uma probabilidade de detecção $\pi_j \in (0, 1]$ associada, indicando uma crença sobre o objeto efetivamente existir no ambiente.

Embora a proposta de representação do mapa não seja adequada para descrever todos os possíveis objetos que o robô pode encontrar, ele contém os elementos necessários para caracterizar muitos dos objetos mais comuns nos espaços internos. Um O-Map mais geral permitiria que cada objeto fosse representado como um volume geométrico adequado e as posições dos objetos seria substituídas por posturas. Posturas codificam, além da localização dos objetos, a seus ângulos de atitude (rolagem, ataque e guinada). No entanto, as pessoas descrevem muitos objetos de interesse pelos parâmetros de volumes eretos, como prismas tendo suas bases paralelas ao plano do chão. Além disso, considerando os propósitos do mapa proposto, que excluem, por exemplo, a renderização de imagens de textura dos objetos, as suas orientações em relação ao plano do chão podem ser omitidas.

Neste trabalho, os objetos são modelados por cilindros. Assim, as medidas resultantes da observação de objetos pelo robô não dependem do ângulo a partir do qual eles são observados, simplificando o processo de mapeamento. O inconveniente deste pressuposto é que objetos alongados como portas, mesas, sofás são, portanto, descritos por uma altura e uma largura média, não refletindo a sua forma plana. No entanto, a representação cilíndrica ainda reflete os tamanhos objeto suficientemente bem para os fins apresentados por este trabalho, melhor explicado na Seção 3.1.

3.1 Aplicações de O-Maps

O-Maps codificam diferentes aspectos dos objetos que representam, sendo útil para diferentes aplicações. Como mapas de navegação para robôs, eles podem ser usados para ajudar a localizá-lo em um ambiente e planejar um caminho em que os objetos são pontos de referência. Por outro lado, porque o mapa possui conteúdo semântico, também pode ajudar na interação e comunicação com seres humanos. Em algumas aplicações, como robôs de serviço e AR, um sistema automático deve se comunicar e interagir com um usuário humano. No caso dos conceitos envolvidos nessa comunicação serem objetos, a proposta de um mapa com base em objetos pode ser útil. Assim, O-Maps podem ser aplicados em tarefas mais complexas, onde navegação ou reconhecimento de objetos é apenas parte do problema.

Para ilustrar o caráter multi-funcional de O-Maps, um exemplo de como eles poderiam ser usados em robótica é dado. Considere a situação em que um usuário solicita a um robô de serviço pegar um copo na mesa da cozinha. O robô deve ir para a cozinha, explorá-la, identificar a mesa ali e possivelmente os vários copos sobre ela, e perguntar ao usuário que copo deve pegar através de uma área de trabalho remota. Um O-Map poderia ser utilizado em todas as tarefas envolvidas.

Antes de tudo, o robô de serviço do exemplo tem de saber como chegar à cozinha. Se ele tem um O-Map da casa, a cozinha pode ser reconhecida pelos objetos que ela contém, usando uma técnica de modelagem espacial como a proposta por Ranganathan & Dellaert (2007). Então, entre os objetos considerados como pertencentes à cozinha, a mesa é selecionada como o alvo para onde o robô deve ir. Se um algoritmo reativo de navegação e desvio de obstáculos for utilizado, como o descrito por Selvatici & Costa (2007b), não é necessário fornecer qualquer outro mapa para o robô. Esse algoritmo só precisa de um ponto objetivo e um mapa para localização, enquanto navega evitando obstáculos usando sensores de sonar.

Quando o robô detecta a mesa da cozinha e o copo ou copos que estão sobre ela, o mesmo tem de associá-los aos copos registrados no mapa. Se o procedimento de associação de dados falhar em casar os copos detectados com os mapeados, então novos copos são mapeados através da técnica O-SLAM, descrita no Capítulo 4. Então, o robô envia uma imagem dos copos para a área de trabalho remota através da qual o usuário se comunica com o robô, destacando os copos detectadas, como demonstrado na Figura 3.2. Naquele momento, existe uma associação direta entre

Figura 3.2: Ilustração de como um O-Map pode ser usado para criar um vocabulário comum para a interação homem-robô. Este exemplo é meramente ilustrativo, e não foi realizado experimentalmente. (a) Vista superior de parte de uma cozinha representada por seus objetos. (b) Amostra de uma imagem usada para a construção do mapa, mostrando os copos detectados. A associação entre cada detecção e o objeto correspondente no mapa é utilizada pelo usuário humano para especificar o copo desejado.

cada copo detectado na imagem e os correspondentes no mapa, de modo que quando o usuário indica o copo que ele, o robô sabe exatamente qual escolher, e é capaz de manobrar a garra em direção a ele, mesmo que o objeto alvo saia do campo de visão. Finalmente, o robô usa o O-Map para navegar de volta para o usuário, trazendo o desejada copo.

Este simples exemplo mostra que o mapa proposto nesta tese é adequado para tarefas relacionadas com navegação de robôs e também como uma base para a comunicação entre robôs e seres humanos, indo de encontro a algumas das necessidades da pesquisa em robótica atual, que também está preocupada com questões relativas a interação homem-robô. Neste trabalho, no entanto, só a tarefa de navegação é tratada; especificamente, o problema da localização do robô e de construção do mapa baseado em objetos.

3.2 Arcabouço para O-SLAM

Embora a estrutura de dados utilizada no O-Map é nova no contexto de navegação de robôs, a principal contribuição desta tese é a técnica probabilística que o constrói. Uma vez que a técnica proposta não depende de saber a localização do robô, que deve ser inferida conjuntamente com os parâmetros dos objetos detectados, ela é

Figura 3.3: Ilustração do cenário considerado pela técnica O-SLAM. O contorno do objeto detectado no exemplo é simplificado como um retângulo, que corresponde à projeção da seção longitudinal de um cilindro.

chamada de *O-SLAM*.

A técnica proposta para a resolução do problema de SLAM considera que o robô explora o ambiente com objetos estáticos, enquanto registra os dados de odometria e captura imagens através de uma câmara de vídeo ligada a ele, como mostrado na Figura 3.3. As imagens capturadas são processadas por algoritmos de detecção de objetos especializados em detectar e segmentar instâncias de certas classes de objetos. Cada algoritmo de detecção reconhece objetos pertencentes a uma classe específica nas imagens. Uma vez que o mapa baseado em objetos os representam como formas volumétricas simples, a forma de cada objeto segmentado na imagem é transformada em uma figura 2D simples, correspondente a uma projeção simplificada da forma volumétrica assumida no plano da imagem. Os dados gerados por este processo consistem nas posições aparentes dos objetos nas imagens, nos parâmetros que descrevem o contorno simplificado dos objetos nas imagens, e nos rótulos de classe atribuídos pelos correspondentes detectores de objetos.

Para simplificar a notação, todos os dados dos sensores são considerados como adquiridos simultaneamente, a cada passo de aquisição de dados. Considere um cenário onde o robô captura T imagens em diferentes instantes de tempo, enquanto ele perfaz uma trajetória descrita por $X = \{x_i\}_{i=1}^T$, onde x_i é a postura do robô no instante i . Os dados de odometria são expressos por $V = \{v_i\}_{i=2}^T$, correspondendo à

distância percorrida e à diferença de orientação do robô em dois instantes de tempo consecutivos, medidos por *encoders* na roda do robô. Os dados adquiridos por visão computacional compreendem K_i detecções de objetos em cada instante i , sendo representados por $Z = \{z_i\}_{i=1}^T$, onde $z_i = \{z_{i,k}\}_{k=1}^{K_i}$. Considerando os pressupostos sobre os dados adquiridos pelos algoritmos de detecção de objetos, como ilustrado na Figura 3.3, temos

$$z_{i,k} = (u_{i,k}, s_{i,k}, a_{i,k}), \quad (3.1)$$

onde cada medição $z_{i,k}$ fornece a posição 2D do objeto na imagem, $u_{i,k}$, os respectivos parâmetros do seu contorno, $s_{i,k}$, e a classe detectada $a_{i,k} \in \mathcal{C}$.

3.2.1 Hipóteses sobre o sistema de detecção de objetos

A forma como os dados adquiridos pelos sensores são processados a fim de obter o O-Map depende da acurácia do algoritmo de visão computacional, o que influencia diretamente a precisão do mapeamento e o custo computacional.

No caso mais simples, assume-se que os algoritmos de detecção de objetos são capazes de perfeitamente detectar e reconhecer os objetos individualmente no ambiente. Isso significa que a identidade de cada objeto detectado é conhecida, e todas as medições tomadas de um mesmo objeto físico são marcadas com a mesma identidade. Assim, a solução da associação de dados consiste em associar diretamente todas as medições com a mesma identidade. Tanto a solução para a associação de dados quanto a identidade dos objetos são expressas pela *função de correspondência* $J : \{i \times \{1, \dots, K_i\}, i = 1, \dots, T\} \rightarrow \mathbb{N}^+$, que é um mapeamento de índices de tempo i e índices de detecções k em índices de objetos no mapa j , de tal forma que o_j , com $j = J(i, k)$, é o objeto detectado na imagem obtida no instante i dando origem à medição $z_{i,k}$.

A função de correspondência J também pode ser vista como uma forma de particionar o conjunto de índices de detecção, conforme ilustrado na Figura 3.4. As medições $z_{i,k}$ cujos índices (i, k) são mapeados para o mesmo objeto j formam um subconjunto das medições Z . Analogamente a uma partição de conjunto, uma solução válida para a associação de dados atribui pelo menos uma medição a cada objeto, e não podem associar a mesma medição com diferentes objetos. Na Figura 3.4, as medições são representadas por linhas conectando círculos, representando as sucessivas poses do robô, com estrelas, que representam os objetos detectados pelas medições. Círculos

Figura 3.4: Ilustração de como a função de correspondência J particiona o conjunto de medições em N subconjuntos (aqui, $N = 4$). Na figura, círculos representam a postura do robô em diferentes instantes $i = 1, 2, \dots, 8$, e as estrelas representam os objetos detectados com índices $j = 1, 2, 3, 4$. As linhas representam as medições disponíveis, indexadas pelos pares (i, k) , que são mostrados apenas em parte, para não carregar a figura. As linhas são distribuídas entre os objetos por J , de modo que $J(i, k) = j$. Por exemplo, temos $J(2, 1) = 1$ e $J(4, 2) = 4$ na figura.

são anotados com índices de postura i , variando de 1 a 8, enquanto as estrelas são anotados com índices de objeto j , variando de 1 a 4 na figura. A função de correspondência J agrupa as linhas, anotadas com índices de detecção (i, k) , em diferentes objetos, particionamento o conjunto das linhas.

Os algoritmos de detecção de objetos existente não são perfeitos, e assim um algoritmo de SLAM baseado em objetos com utilidade prática deve considerar erros nas identidades e classes do objeto. Supondo que $C = \{c_j\}_{j=1}^N$ seja o conjunto de todas as classes dos objetos mapeados, há quatro configurações possíveis do problemas a considerar:

1. As classes C e a função de correspondência J são conhecidos;
2. As classes C são conhecidas, mas a função de correspondência J é desconhecida;
3. A função de correspondência J é conhecida, mas as classes C são desconhecidas, e
4. As classes C e a função de correspondência J são desconhecidas.

Embora a configuração 4 é a que melhor descreve a maioria das situações práticas, outras configurações têm algum interesse especial. A configuração 1 pode ser assumida se os objetos têm características especiais que são facilmente detectadas, como uma etiqueta com código de barras, ou se o reconhecimento é feito à mão, com a ajuda de um operador humano. A configuração número 2 corresponde a situações em que os objetos são detectados corretamente, mas o número de objetos de cada classe é desconhecido. O caso de única categoria se enquadra nesta configuração, onde não é feita distinção entre as classes de objetos. Finalmente, a configuração número 3 ocorre quando os objetos são identificados perfeitamente, mas as suas classes são desconhecidas. Este caso, porém, não tem qualquer interesse para o arcabouço proposto, uma vez que assume-se que as classes de objetos são facilmente inferidas caso suas identidades possam ser perfeitamente reconhecidas.

3.2.2 Acoplamento das incógnitas através de modelagem probabilística

A abordagem proposta para recuperar o mapa e a trajetória do robô é modelar o comportamento dos odômetros do mesmo e dos detectores de objetos utilizando modelos probabilísticos e, em seguida, deduzir os parâmetros desejados. Em todos os casos descritos na Seção 3.2.1, as leituras dos odômetros são consideradas como corrompidas por ruído branco, bem como os parâmetros de posição e tamanho aparentes dos objetos nas imagens.

No caso da configuração 1, onde as classes dos objetos e suas identidades são todas conhecidas, as únicas variáveis são a trajetória do robô e a geometria e localizações dos objetos. Uma vez que só os parâmetros geométricos precisam ser inferidos neste caso, o problema de SLAM a partir de objetos assumindo-se a configuração 1 é aqui denominado SLAM geométrico, cuja solução é proposta aqui de forma inédita. Nenhum trabalho conhecido considera o problema de se modelar objetos através de volumes geométricos usando seqüências de imagens capturadas por um robô.

Por outro lado, no caso da configuração 2, temos também de resolver o problema de associação de dados, de modo que as identidades dos objetos são consideradas desconhecidas. Este trabalho considera o pior caso para o problema de associação de dados, onde os dados Z não contêm medidas diretas das identidades dos objetos. Assim, se houver duas ou mais detecções de sofás, não há nenhuma medida que indica se os dados devidos a essas detecções correspondem ao mesmo sofá ou a

diferentes sofás. Aqui, um modelo probabilístico Bayesiano é assumido para a função de correspondência, a ser melhor explicado na Seção 4.2.

No caso de assumirmos a configuração 4, que considera imperfeitos os algoritmos de visão computacional os erros na detecção dos objetos são modelados como dois modos diferentes de erro: confusão entre as classes ou detecção espúria. O primeiro modo de erro ocorre quando um detector de objetos identifica um objeto pertencente a uma classe diferente, semelhante à que o detector é especializado. O último modo corresponde à atribuição errônea de uma determinada classe de objeto classe para um pedaço aleatório da imagem. Ambos os erros podem ocorrer devido a mudanças na luminosidade do ambiente, imagem ruidosa, ou outros fatores. Portanto, diferentemente de todas as outras abordagens de mapeamento de objetos presentes na literatura robótica, que confia cegamente no sistema de reconhecimento do objeto, a técnica de O-SLAM infere a trajetória do robô em conjunto com as geometrias e classes de objetos, detectando possíveis objetos espúrios.

O modelo probabilístico utilizado por O-SLAM também considera um conhecimento prévio sobre o tamanho dos objetos e, possivelmente, a sua localização, dependendo de suas classes. A idéia principal por trás disso é que, se se conhece uma estimativa aproximada do tamanho real dos objetos pertencentes a uma determinada classe, o tamanho aparente de uma instância na imagem leva a uma estimativa grosseira da distância do robô para o objeto. A diferença interessante entre SLAM visual com base em pontos (DAVISON, 2003) é de que o tamanho aparente agora indica a distância dos objetos mesmo com uma única observação. Após várias detecções, as dimensões e posições dos objetos serão fortemente determinadas por triangulação, tornando o conhecimento a priori obsoleto e vinculando as classes do objeto com os parâmetros geométricos. A Figura 3.5 ilustra este modelo. Esta idéia foi também utilizada por Hoiem, Efros & Hebert (2006) para recuperar a posição da câmera e determinar a posição dos objetos no plano do chão a partir de uma única imagem, enquanto executa reconhecimento de objetos usando uma técnica de terceiros, vinculando fortemente a classe do objeto com as regiões em que aparecem na imagem. No entanto, aqui as restrições sobre as localizações, tamanhos e classes dos objetos são mais relaxadas, a fim de recuperar modelos de maior escala a partir de várias imagens.

Figura 3.5: Ilustração de como o conhecimento sobre o tamanho e posição do objeto podem ser utilizados para estimar a distância relativa da câmara. (a) A diminuição do tamanho da caixa de madeira nas imagens combinada com o conhecimento prévio sobre o tamanho das caixas informa sobre a sua crescente distância para a câmara. (b) modelo probabilístico gráficos que ilustram as dependências entre posição e tamanho dos objetos, e postura da câmara do robô (incógnitas são representados por elipses) dada a saída do algoritmo de detecção de objetos (retângulos).

3.2.3 Algoritmos de visão computacional compatíveis com O-SLAM

Este trabalho pressupõe que o sistema de reconhecimento de objetos que processa as imagens captadas pelo robô é capaz de detectar e segmentar instâncias das classes de interesse. Assim, os algoritmo de classificação de imagem, que rotulam imagens de acordo com o objeto mais proeminente na mesma, são incompatíveis com a abordagem aqui proposta. Existem duas classes de algoritmo que podem ser usadas com O-SLAM: métodos de rotulação de pixel e detectores de objeto.

Métodos de rotulação de pixel atribuem um rótulo a cada pixel da imagem. Estes métodos são geralmente baseados em abordagens probabilísticas discriminativas que consideram o valor de cada pixel e seus vizinhos para inferir o melhor marcador para ele. Eles têm recebido grande impulso durante os últimos anos, depois que Conditional Random Fields (CRFs) (LAFFERTY; MCCALLUM; PEREIRA, 2001) e suas variantes

foram adaptadas do problema de rotulação de seqüências de estados para rotulação de pixels em um reticulado (QUATTONI; COLLINS; DARREL, 2004; CRIMINISI et al., 2006; WINN; SHOTTON, 2006; SHOTTON; JOHNSON; CIPOLLA, 2008). Contudo, se considera-se portá-los para o problema de SLAM a partir de objetos, estes algoritmos devem ser usados com cuidado. Uma vez que eles dependem de uma fase de treinamento onde imagens completamente rotuladas são utilizadas como amostras, o modelo aprendido rotula os pixels pertencentes ao objetos tendo os *pixels da borda* em consideração. Isso representa um problema quando o mesmo objeto deve ser detectado a partir de diferentes pontos de vista, uma vez que o fundo muda de uma imagem para outra, mudando também o valor dos pixels de borda do objeto.

Detectores de objeto são mais adequados para utilização com O-SLAM. Essas técnicas utilizam características locais e globais para detectar e segmentar objetos nas imagens, independentemente do fundo. Características locais correspondem a elementos de informação que descrevem pontos isolados na imagem, como cor do pixel, gradiente de cor, detecção de de quinas, etc. Características globais são informações que representam partes de uma imagem, ou mesmo toda ela, tais como a média de cor, histogramas, etc.

Os detectores de objeto mais simples utilizam apenas informações sobre cores, tais como histogramas de cores (SWAIN; BALLARD, 1990) e *blobs* de cores (BRUCE; BALCH; VELOSO, 2000). A primeira técnica tem sido utilizada com sucesso em SLAM a partir de objetos (BRUCE; BALCH; VELOSO, 2000), conjugada com o uso de descritores SIFT locais como um passo de refinamento para filtrar falsas detecções. SIFT é o acrônimo para *Scale-Invariant Feature Transform*, uma técnica que detecta características locais em imagens amplamente utilizada nos algoritmos modernos de reconhecimento de objetos (LOWE, 1999). Outros detectores de objeto utilizados em mapeamento de objetos utilizam descritores SIFT com grande sucesso (LOWE, 1999). Recentemente, técnicas promissoras de detecção robusta de múltiplas classes de objeto em imagens foram propostas (MIKOLAJCZYK; LEIBE; SCHIELE, 2006; THOMAS et al., 2006), sendo muito adequadas para a integração com O-SLAM. Apesar do seu potencialmente elevado custo computacional, devido à realização de operações de pesquisa sobre toda a imagem, essas técnicas podem ser otimizados para detectar os mesmos objetos em imagens consecutivas, minimizando a necessidade de pesquisa extensiva. Além disso, a atual forma de O-SLAM corresponde a um algoritmo de processamento em lote, não tendo requisitos apertados de tempo de processamento.

3.3 Comparação com outros mapas baseados em objetos

Esta tese sustenta a introdução de um novo mapa baseado em objetos para navegação de robôs e o primeiro algoritmo de SLAM que trata os erros de reconhecimento de objeto concomitantemente com o mapeamento 3D de objetos e a localização do robô. Assim, poucos aspectos do presente trabalho podem ser comparados com trabalhos relacionados.

Estritamente falando, nenhum outro trabalho resolve o problema de SLAM com base apenas nos objetos detectados nas imagens. A maioria dos trabalhos dependem de uma técnica tradicional de SLAM para localização e mapeamento (FORSEN et al., 2008; EKVALL; KRAGIC; JENSFELT, 2007), utilizando exclusivamente odômetros e sensor de laser para isso. A localização dos objetos detectados é, então, registrada no mapa, mas ela não interfere na localização do robô. Portanto, esses trabalhos realizam mapeamento de objetos, mas não SLAM. O único trabalho relacionado que usa apenas detecções de objetos para construir o mapa localiza o robô apenas por *dead reckoning*, e portanto não constituem uma técnica SLAM (VASUDEVAN et al., 2007). Embora usar apenas odômetros e detecção de objetos não possui vantagem em termos da precisão da localização do robô, o arcabouço para SLAM com base em objetos desenvolvidos nesta tese pode ser facilmente integrado a qualquer outro algoritmo probabilístico de SLAM, melhorando a precisão do mapa e da localização e ajudando a fechar grandes voltas na trajetória.

Este trabalho também difere das abordagens de SLAM visual, que só mapeiam alguns pontos de interesse do ambiente (DAVISON, 2003; KLEIN; MURRAY, 2007; WILLIAMS; KLEIN; REID, 2007), sem fornecer a representação dos objetos. Ligeiramente mais ligada com a abordagem atual, (CASTLE et al., 2007) também incorpora algumas entidades mais abstratas no mapa, correspondendo a segmentos planos identificados nas imagens. No entanto, essa abordagem só indica que certos pontos do mapa pertencem a uma estrutura específica, fazendo suas associações mais confiável devido a esse sistema adicional de verificação. O presente trabalho contribui ao usar as informações de tamanho e a classe dos objetos, e não apenas as posições de certos pontos.

Embora o trabalho de (RANGANATHAN; DELLAERT, 2007) não esteja preocupado com mapeamento objetos detectados em várias imagens captadas por um robô móvel,

mas sim em extrair uma representação 3D dos objetos presentes em uma única imagem, ela possui algumas semelhanças com o presente trabalho. Em vez de depender de um algoritmo de detecção de objetos pré-existente, o trabalho integra diferentes aspectos geométricos e visuais de uma imagem captada por um par estereográfico — proporcionando informações de cor e profundidade — para mapear a localização 3D, a dimensão e a classe dos objetos retratados.

O-SLAM e do método de modelagem lugar de Ranganathan & Dellaert (2007) são, então abordagens complementares. Enquanto aquele trabalho desempenha segmentação e classificação probabilística de objetos, a abordagem aqui proposta limita-se a executar classificação de objetos, baseando-se no segmentação determinística gerada pelo algoritmo de visão computacional. No entanto, O-SLAM é capaz de encontrar o mesmo objeto físico em diferentes imagens e recuperar a trajetória da câmera, enquanto que a outra abordagem não é. Apesar das diferenças de ambas os trabalhos, eles apontam para o mesmo objetivo final: a integração de aspectos geométricos e visuais, a fim de reconhecer e mapear os objetos do ambiente.

4 Inferência em O-SLAM

Este capítulo apresenta o algoritmo proposto para a obtenção mapas de objetos, a saber, o algoritmo O-SLAM. De acordo com a decisão de se utilizar uma abordagem probabilística para resolver problema de SLAM e classificação de objetos, neste capítulo, o modelo probabilístico do problema é desenvolvido, juntamente com a técnica de inferência utilizada para estimar os parâmetros do mapa. Embora o algoritmo proposto se destine a resolver o problema sem que a solução para o problema de associação de dados seja conhecida e considerando imperfeita a detecção de objetos, a solução para as configurações 1,2, e 4 explicadas na seção 3.2.1 é apresentada em sequência, uma vez que a solução para uma configuração mais difícil depende da solução da mais fácil.

A primeira configuração a ser considerada é a configuração 1, o que implica o problema sozinho de inferência geométrica, sendo referido como *SLAM geométrico* neste trabalho. Este problema, cuja solução é descrita na Seção 4.1, consiste na construção de um O-Map considerando que a detecção e reconhecimento de objetos são perfeitos, de modo que as classes e identidades corretas dos objetos são conhecidas.

A Seção 4.2 apresenta a solução proposta para a configuração 2, que considera o o algoritmo de detecção de objetos perfeito, mas assume desconhecida a solução para o problema de associação de dados. Uma vez que o objetivo é inferir O-Maps e a trajetória do robô, as variáveis de interesse são obtidas através da sua expectativa em relação às possíveis soluções para a associação de dados.

A Seção 4.3 estende a solução proposta para a configuração 2 para a configuração 4, onde os algoritmos utilizados são assumidos imperfeitos, e podem ter a saída errada. Para além do ruído nas aparentes posições e tamanhos dos objetos, tratados na proposta de solução para o problema de SLAM geométrico (configuração 1), dois modos de erro são admitidos: falsas detecções, o que corresponde a falsos positivos e medições de objetos parcialmente ocluídos, e classes erroneamente detectadas.

A idéia de fundo para a solução do problema de SLAM a partir de objetos é usar uma amostragem do tipo *Rao-Blackwell*, que representa uma distribuição de probabilidade combinando amostragem e representação paramétrica analítica. Primeiramente, as variáveis são divididas em dois grupos: o primeiro, das que possuem inferência exata computacionalmente intratável, e o segundo grupo, das que possuem inferência tratável conhecendo-se valores das variáveis do primeiro grupo. Então, a amostragem do tipo Rao-Blackwell usa alguma técnica para amostrar as variáveis do primeiro grupo, e, para cada amostra obtida, obtém também a forma analítica da crença sobre as variáveis do segundo grupo.

No presente trabalho, as variáveis do primeiro grupo são a solução da associação de dados J e as classes dos objetos C , e as variáveis do segundo grupo são aquelas envolvidas no problema de SLAM geométrico. Para a amostragem das variáveis do primeiro grupo, emprega-se, aqui um método MCMC.

4.1 Solução eficiente para o problema de SLAM geométrico

Nesta seção, mostra-se como resolver o problema geométrico de SLAM, assumindo o caso de perfeita detecção e reconhecimento de objetos. A solução proposta aqui é a realização de inferência do tipo *Maximum a Posteriori* (MAP) utilizando dados das medições Z fornecidas pelo sistema de detecção de objetos e da função de correspondência J fornecida pelo conhecimento das identidades dos objetos, segundo declarado na configuração 1 na Seção 3.2.1. A estimativa MAP (X^*, M^*) é definida por

$$(X^*, M^*) \triangleq \arg \max_{(X, M)} P(X, M | J, Z). \quad (4.1)$$

A abordagem probabilística implica que as classes de objetos e suas localizações e geometria associadas são inferidas juntamente com as posturas do robô, conectando essas variáveis. Retomando a definição do mapa dada no do Capítulo 3, em que $M \triangleq \{o_j\}_{j=1}^N$, onde cada objeto $o_j \triangleq (l_j, g_j, c_j)$ é descrito por sua localização l_j , geometria g_j , e rótulo de classe c_j . O número de objetos no mapa, N , depende das diferentes identidades dos objetos detectados pelo reconhecimento de objetos.

Assumindo a configuração 1, onde as identidades e classes são conhecidas, uma abordagem semelhante à utilizada no tradicional problema de SLAM é usada, com

a exceção de que o mapa agora inclui geometrias e classes de objetos. Sendo as classes determinadas diretamente a partir daquelas perfeitamente detectadas $a_{i,k} \in Z$, e as identidades objeto dadas por J , a crença a posteriori (4.1) é expressa por

$$P(X, M | C, J, Z) \propto P(Z | X, M, C, J) P(X, M | C, J) = P(X, L, G | C, J) P(Z | X, L, G, C, J), \quad (4.2)$$

onde $P(X, L, G | C, J)$ é uma densidade a priori sobre a trajetória e a parte geométrica do mapa, condicionada nos rótulos de classe $C = \{c_j\}_{j=1}^N$ and e função de correspondência J . $P(Z | X, M, J)$ é a verossimilhança conjunta das projeções dos objetos detectados nas imagens. Uma vez que, por enquanto, as variáveis sujeitas à inferência são a trajetória do robô X , as localizações dos objetos $L = \{l_j\}_{j=1}^N$ and suas geometria $G = \{g_j\}_{j=1}^N$, define-se aqui o vetor de variáveis $\theta \triangleq (X, L, G)$.

Para descrever mais precisamente o modelo adotado, as relações entre as variáveis e as medições são mostradas na Figura 4.1. Além do modelo de medição, que consiste no modelo de projeção de objetos no plano da imagem, assumimos um modelo prévio sobre os tamanhos dos objeto, em função das suas classes. A densidade prévia da crença sobre a trajetória do robô e os modelos geométricos dos objetos pode ser escrita como

$$\begin{aligned} P(\theta | C, J) &= P(X) P(L | C) P(G | C) \\ &= P(X) \prod_{j=1}^N \{P(l_j | c_j) P(g_j | c_j)\}, \end{aligned} \quad (4.3)$$

e a informação da odometria V impõe uma densidade a priori da forma

$$P(X) = P(x_1) \prod_{i=2}^T P(x_i | x_{i-1}, v_i). \quad (4.4)$$

Uma vez que não é utilizado nenhum sensor de localização absoluta, o mapa obtido pode ter qualquer coordenada de referência. Uma escolha comum para essa definição é escolher x_1 como uma constante com qualquer valor, por vezes igual à origem, e fazer todas as outras variáveis estimadas em relação a ela.

A verossimilhança das medições é modeladas considerando que as medições são estatisticamente independentes mutuamente, e que as posições dos objetos na imagem dependem do deslocamento relativo entre o robô e os objetos, e também da orientação do robô. O contorno do objeto na imagem é assumida independente da sua posição. Finalmente, consideramos que os rótulos de classe C perfeitamente geram

Figura 4.1: Fragmento da Rede Bayesiana que representa o modelo probabilístico adotado por O-SLAM. Os retângulos representam variáveis conhecidas, e os círculos representam as incógnitas. Estamos basicamente assumindo que a classe do objeto c_j dá uma idéia aproximada sobre a sua geometria g_j , e pode dar pistas sobre sua localização l_j . Em SLAM geométrico, a classe detectada $a_{i,k}$ é considerada igual à real, $c_{J(i,k)}$, de forma que $P(a_{i,k}|c_{J(i,k)}) = \delta(a_{i,k}, c_{J(i,k)})$. Este modelo não admite oclusão entre os objetos, nem objetos em movimento.

os detectados, de modo que:

$$P(Z|X, M, C, J) = \prod_{i=1}^T \prod_{k=1}^{K_i} \{P(u_{i,k}|x_i, l_{J(i,k)})P(s_{i,k}|x_i, l_{J(i,k)}, g_{J(i,k)})\}. \quad (4.5)$$

Como resultado, a crença a posteriori expressa em (4.1) é dada pelo modelo generativo

$$\begin{aligned} P(\theta|C, J, Z) &\propto P(x_1) \prod_{i=2}^T P(x_i|x_{i-1}, v_i) \prod_{j=1}^N \{P(l_j|c_j)P(g_j|c_j)\} \\ &\times \prod_{i=1}^T \prod_{k=1}^{K_i} \{P(u_{i,k}|x_i, l_{J(i,k)})P(s_{i,k}|x_i, l_{J(i,k)}, g_{J(i,k)})\}. \end{aligned} \quad (4.6)$$

Enquanto (4.6) dá a forma geral da distribuição posterior em (4.1), é necessário especificar as formas das densidades envolvidas e chegar a um modelo prático a partir do qual podemos inferir os desejados parâmetros geométricos. Para tal, o arcabouço $\sqrt{SAM}$ é utilizado. De acordo com esse arcabouço, a crença a posteriori em (4.6) é fatorada como produto de probabilidades gaussianas, o que naturalmente leva (4.1) a ser formulado como um problema LS linearizado. Uma vez que estamos a tratar um problema de SLAM *off-line*, resolver o problema linearizado é parte de uma estraté-

gia iterativa de otimização não-linear, como Levenberg-Marquardt (LM) (KELLY, 1999). Nesta Seção, apenas a parte linear da solução é discutida.

4.1.1 Usando modelos gaussianos linearizados

Para assegurar a densidade a posteriori (4.6) é expressa como um produto de densidades gaussianas, o modelo probabilístico geométrico é definido supondo-se que a incerteza relacionada a todas as medições e ao conhecimento prévio sobre as variáveis são razoavelmente bem modeladas assumindo que essas quantidades foram modificadas por um erro aleatório, com distribuição Normal. Assim, a crença a priori sobre as localizações e tamanhos dos objetos é dada por

$$l_j = \gamma(c_j) + e_j^l, \quad e_j^l \sim \mathcal{N}(0, \Gamma(c_j)) \text{ and} \quad (4.7)$$

$$g_j = \varsigma(c_j) + e_j^g, \quad e_j^g \sim \mathcal{N}(0, \Sigma(c_j)), \quad (4.8)$$

onde e_j^l e e_j^g são os erros na localização e tamanho prévios do objeto, respectivamente. As leituras de odometria e medições na imagem também são perturbados por ruído branco, por isso consideramos

$$x_i = f(x_{i-1}, v_i) + e_i^x, \quad e_i^x \sim \mathcal{N}(0, Q_i), \quad (4.9)$$

$$u_{i,k} = h^u(x_i, l_{J(i,k)}) + e_{i,k}^u, \quad e_{i,k}^u \sim \mathcal{N}(0, R_{i,k}), \text{ and} \quad (4.10)$$

$$s_{i,k} = h^s(x_i, l_{J(i,k)}, g_{J(i,k)}) + e_{i,k}^s, \quad e_{i,k}^s \sim \mathcal{N}(0, W_{i,k}), \quad (4.11)$$

onde e_i^x , e_{i,k_i}^u e e_{i,k_i}^s são, respectivamente, os erros de odometria e os erros de posição e tamanho do objeto na imagem. A função determinista de movimento $f(.,.)$ e as covariâncias Q_i definem o modelo probabilístico de movimento assumido, que codifica a nossa crença em onde o robô pode estar depois de ter sido imposta um comando de deslocamento a partir de x_{i-1} representado por v_i .

Analogamente, as projeções deterministas $h^u(.,.)$ e $h^s(.,.)$, e as respectivas matrizes de covariância $R_{i,k}$ and $W_{i,k}$ definem o modelo probabilístico de observação, que codifica a nossa convicção sobre a posição e tamanho do objeto medido na imagem, dado que o robô possui a postura x_i e a localização e tamanho do objeto no mundo real são $l_{J(i,k)}$ e $g_{J(i,k)}$, respectivamente.

Dado que as funções f , h^u e h^s são, em geral, não-lineares, suas versões linearizadas são usadas para garantir uma densidade posterior gaussiana sobre os parâmetros geométricos do mapa, L e G , e a trajetória do robô, X . Substituindo a versão linear

das densidades (4.7-4.11) em (4.6) produz a densidade posterior:

$$\begin{aligned}
 P(\theta|C, J, Z) \propto & \frac{1}{\sqrt{|2\pi\mathbb{P}|}} \exp - \frac{1}{2} \left\{ \sum_{j=1}^N \left[\|\delta l_j + l_j^0 - \gamma(c_j)\|_{\Gamma(c_j)}^2 + \right. \right. \\
 & \left. \|\delta g_j + g_j^0 - \varsigma(c_j)\|_{\Sigma(c_j)}^2 \right] + \sum_{i=2}^T \|F_i^x \delta x_{i-1} - \delta x_i + f(x_{i-1}^0, v_i) - x_i^0\|_{Q_i}^2 \\
 & + \sum_{i=1}^T \sum_{k=1}^{K_i} \left[\|U_{i,k}^x \delta x_i + U_{i,k}^l \delta l_{J(i,k)} + h^u(x_i^0, l_{J(i,k)}^0) - u_{i,k}\|_{R_{i,k}}^2 + \right. \\
 & \left. \|S_{i,k}^x \delta x_i + S_{i,k}^l \delta l_{J(i,k)} + S_{i,k}^g \delta g_{J(i,k)} + h^s(x_i^0, l_{J(i,k)}^0, g_{J(i,k)}^0) - s_{i,k}\|_{W_{i,k}}^2 \right] \Big\}, \quad (4.12)
 \end{aligned}$$

sendo que

$$|\mathbb{P}| = \prod_{j=1}^N (|\Gamma(c_j)| \cdot |\varsigma(c_j)|) \prod_{i=2}^T |Q_i| \prod_{i=1}^T \prod_{k=1}^{K_i} (|R_{i,k}| \cdot |W_{i,k}|), \quad (4.13)$$

onde o índice ⁰ indica o ponto linearização das respectivas variáveis, e $\delta\alpha$ significa a variação da variável α em torno do seu ponto de linearização, de modo que $\alpha = \alpha^0 + \delta\alpha$. As letras maiúsculas decoradas representam os Jacobianos das funções deterministas: F_i^x é o Jacobiano de $f(x_{i-1}, v_i)$ com respeito a x_{i-1} ; $U_{i,k}^x$ e $U_{i,k}^l$ são, respectivamente, os Jacobianos de $h^u(x_i, l_{J(i,k)})$ com respeito a x_i e $l_{J(i,k)}$; $S_{i,k}^x$, $S_{i,k}^l$, e $S_{i,k}^g$ são os Jacobianos de $h^s(x_i, l_{J(i,k)}, g_{J(i,k)})$ com respeito a x_i , $l_{J(i,k)}$, e $g_{J(i,k)}$, respectivamente. A notação $\|\alpha\|_{\Sigma}^2$ é usada para indicar a norma de Mahalanobis ao quadrado de α com respeito a Σ , dado por $\alpha^T \Sigma^{-1} \alpha$. A função $\text{diag}(\mathcal{S})$ toma um conjunto de matrizes quadradas $\mathcal{S}$ como argumento e constrói uma matriz de blocos na diagonal onde os blocos são os elementos do argumento. Note que, dados os pontos de linearização $\{x_i^0\}_{i=1}^T$ e $\{l_j^0, g_j^0\}_{j=1}^N$, as variáveis sujeitas a inferência tornam-se $\delta\theta \triangleq (\delta X, \delta L, \delta G)$, onde $\delta X = \{\delta x_i\}_{i=2}^T$, $\delta L = \{\delta l_j\}_{j=1}^N$, e $\delta G = \{\delta g_j\}_{j=1}^N$. Desde que a postura inicial é considerada conhecida, nós temos $\delta x_1 = (0, 0, 0)$.

4.1.2 Inferência usando decomposição QR

A inferência MAP sobre o posterior (4.14) pode ser transformada em um problema de LS, que pode ser resolvido de forma eficiente usando fatoração QR. Re-escrevendo o posterior (4.12) na forma matricial implica

$$P(\theta|C, J, Z) \propto \frac{1}{\sqrt{|2\pi\mathbb{P}|}} \exp \left\{ -\frac{1}{2} \|A\delta\theta - b\|_{\mathbb{P}}^2 \right\}, \quad (4.14)$$

onde cada bloco de linhas em A e b corresponde a um dos termos no somatório em (4.12), e $\mathbb{P}$ é a matriz com blocos na diagonal contendo as covariâncias que ponderam as parcelas. Por exemplo, são fragmentos dessas matrizes e vetores:

$$A = \begin{bmatrix} F_2^x & -I_{3 \times 3} & \cdots & 0 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \cdots \\ 0 & U_{2,1}^x & \cdots & U_{2,1}^l & 0 & \cdots \\ 0 & S_{2,1}^x & \cdots & S_{2,1}^l & S_{2,1}^g & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad \delta\theta = \begin{bmatrix} \delta x_2 \\ \vdots \\ \delta l_1 \\ \delta g_1 \\ \vdots \end{bmatrix}, \quad b = \begin{bmatrix} x_2^0 - f_2(x_1^0, v_2) \\ \vdots \\ u_{2,1} - h^u(x_2^0, l_1^0) \\ s_{2,1} - h^s(x_2^0, l_1^0, g_1^0) \\ \vdots \end{bmatrix},$$

$$\text{and } \mathbb{P} = \begin{bmatrix} Q_2 & \cdots & 0 & 0 & \cdots \\ \vdots & \ddots & \vdots & \vdots & \cdots \\ 0 & \cdots & R_{2,1} & 0 & \cdots \\ 0 & \cdots & 0 & W_{2,1} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix},$$

onde $I_{d \times d}$ é a matriz identidade com d elementos na diagonal.

Maximizando o posterior (4.14) corresponde a encontrar

$$\delta\theta^* = \arg \min_{\delta\theta} \|A\delta\theta - b\|_{\mathbb{P}}^2, \quad (4.15)$$

o que também a média a posteriori dos parâmetros, cuja covariância é expressa por $C_\theta = (A^T \mathbb{P}^{-1} A)^{-1}$.

Devido à esparsidão de A , a fatoração QR é uma maneira eficiente de resolver (4.15) (DELLAERT; KAESSE, 2006). Fatoração QR representa uma matriz $m \times n$ A , com $m \geq n$, como a multiplicação de outras duas matrizes (GOLUB; LOAN, 1996), $A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$, onde Q é uma matriz ortonormal $m \times m$, e R é o fator de Cholesky super-triangular $n \times n$ de $A^T A$. Reescrevendo (4.15) como a minimização de uma distância euclidiana, incorporando a matriz de covariância nos outros termos, tem-se:

$$\|A\delta\theta - b\|_{\mathbb{P}}^2 = \|\bar{A}\delta\theta - \bar{b}\|^2, \quad (4.16)$$

onde $\bar{A} = \mathbb{P}^{-\frac{1}{2}} A$, $\bar{b} = \mathbb{P}^{-\frac{1}{2}} b$, e $\mathbb{P}^{-\frac{1}{2}}$ é o fator de Cholesky $\mathbb{P}^{-1}$. A fatoração QR de $\bar{A}$

permite-nos re-escrever (4.15) na forma

$$\begin{aligned}\delta\theta^* &= \arg \min_{\theta} \left\| Q \begin{bmatrix} R \\ 0 \end{bmatrix} \delta\theta - \bar{b} \right\|^2 \\ &= \arg \min_{\theta} \left\| Q \left(\begin{bmatrix} R \\ 0 \end{bmatrix} \delta\theta - \begin{bmatrix} c \\ r \end{bmatrix} \right) \right\|^2,\end{aligned}\quad (4.17)$$

onde $\begin{bmatrix} c \\ r \end{bmatrix} = Q^T \bar{b}$.

Sendo R triangular, quadrada e não-singular, desde que $\bar{A}$ represente um sistema super-definido, a solução do problema é evidente: é dada pela solução de $R\delta\theta^* = c$, deixando $\|r\|^2$ como o resíduo quadrático total. Se a covariância posterior for pedida, ela pode ser calculada a partir do fator de Cholesky R :

$$C_{\theta} = (A^T P^{-1} A)^{-1} = (R^T R)^{-1} = R^{-1} (R^{-1})^T. \quad (4.18)$$

A fatoração QR da matriz $\bar{A}$ possui íntima ligação com a representação em árvore da estrutura gráfica do problema de SLAM nela representado (veja o trabalho de Dellaert & Kaess (2006) para mais informações). Dessa forma, a ordem e o agrupamento das variáveis contam muito na eficiência do processo de obtenção dos fatores Q e R e na solução do problema $R\delta\theta^* = c$. Uma vez que cada coluna de $\bar{A}$ representa uma variável do problema, recomenda-se a reordenação das colunas dessa matriz de acordo com alguma heurística. Neste trabalho foi usada a heurística COLAMD, já implementada em Matlab.

Agora, temos todas as ferramentas matemáticas para resolver o problema geométrico de SLAM. O Algoritmo 1 descreve o algoritmo para a configuração 1, quando a função de correspondência e classes dos objetos J e C são dadas. Na próxima Seção, será tratado o caso em que o J correto é desconhecido, mas sabe-se como amostrar a distribuição possíveis casos de J . O Capítulo 5 é dedicado a descrever a forma de executar essa amostragem.

4.2 Associação de dados probabilística

Tipicamente, as identidades dos objetos não estão disponíveis, e então a solução da associação de dados (incluindo declarações sobre as identidades dos objetos

Algorithm 1 Algoritmo O-SLAM considerando detecção e reconhecimento de objetos perfeito.

Entrada: dados Z , a função de correspondência J , os modelos de movimento e observação definidos pelas funções f , h^u , e h^s e as covariâncias Q_i , $R_{i,k}$, and $W_{i,k}$

Saída: a trajetória ótima X^* , o mapa ótimo M^* , e opcionalmente a matriz de covariância das variáveis C_θ

1. Determine as classes $C = \{c_j\}_{j=1}^N$ de forma que $\forall z_{i,k} \in Z, c_{J(i,k)} = a_{i,k}$
 2. Escolha os pontos de linearização iniciais
 3. Execute um algoritmo iterativo de otimização não-linear tal como LM, de forma que, em cada passo:
 - (a) Construa as matrizes A e $\mathbb{P}$ e o vetor b usando os modelos de movimento e observação linearizados
 - (b) Resolva o sistema linear definido por A , $\mathbb{P}$ e b usando fatoração QR
 - (c) Determine os novos pontos de linearização de acordo com as regras da técnica de otimização escolhida
 - (d) Se a mudança nos pontos de linearização for menor do que um certo limiar, pare a iteração e retorne os valores atuais de A , b , e $\mathbb{P}$ e os pontos de linearização
 4. Resolva o sistema formado por A , $\mathbb{P}$ e b usando a fatoração QR, obtendo o vetor de variáveis $\delta\theta$
 5. Encontre $(X^*, L^*, G^*) = \delta\theta + \theta^0$, onde θ^0 são os pontos de linearização das variáveis, e $M^* = (L^*, G^*, C)$
 6. Se necessário, calcule C_θ de acordo com (4.18)
-

feitas na Seção 3.2.1) deve ser inferida conjuntamente com as variáveis de ordem geométrica. Nesta seção, o algoritmo O-SLAM para a configuração 2 é discutido, considerando-se o caso de construção de O-Maps sem assumir a solução correta para associação de dados, embora ainda considerando conhecidas as classes dos objetos.

Uma vez que J está sujeito à inferência, o vetor de variáveis θ deve inclui-lo. Para tanto, é redefinido como $\theta \triangleq (J, \theta_J)$, onde $\theta_J \triangleq (X, L_J, G_J)$ são os parâmetros geométricos sob inferência dada a função de correspondência J . No entanto, os parâmetros que se deseja estimar permanecem os mesmos, especificamente a trajetória do robô X , as localizações dos objetos L e tamanhos G dos objetos, que não dependem de uma solução específica para a associação de dados.

Caso todos os vetores de parâmetros θ_J tivessem a mesma dimensionalidade e natureza, ou seja, cada posição em θ_J correspondesse à mesma variável física independentemente do valor de J , as incógnitas seriam estimadas levando-se em conta todas as possíveis soluções da associação de dados:

$$\hat{\theta}_J \triangleq E[\theta_J|Z] = \sum_J P(J|Z) \int_{\theta_J} \theta_J P(\theta_J|C, J, Z) d\theta_J, \quad (4.19)$$

Em visão computacional, esta abordagem é conhecida como *structure-from-motion* sem correspondência (DELLAERT et al., 2000). Ela pode ser usada quando a essência do vetor de variáveis é conhecida a priori, ou seja, θ tem um tamanho fixo e cada um dos seus componentes corresponde a uma variável específica do problema, configurando o problema de associação de dados restrito descrito na Seção 2.2.3. A vantagem da abordagem *structure-from-motion* sem correspondência reside em tirar partido de todas as informações que possam ser recolhidas a partir dos dados Z para inferir as variáveis de interesse, obtendo ótimos resultados, mesmo que os dados disponíveis não sejam suficientes para determinar uma solução única para associação de dados.

A abordagem de *structure-from-motion* sem correspondência pode ser usada para inferir a trajetória do robô X . Em primeiro lugar, a trajetória X está sempre presente em θ_J para qualquer instância de J , e sempre indica a seqüência de posturas do robô nos tempos $i = 1$ até $i = T$. Assim, podemos definir a trajetória desejada como a

expectativa

$$\hat{X} \triangleq E[X|Z] = \sum_J P(J|Z) \int_{\theta_J} X P(\theta_J|J, Z) d\theta_J \quad (4.20)$$

Porém, quando se consideram os parâmetros geométricos do mapa L e G , a dimensionalidade de θ_J torna-se desconhecida a priori, e depende dos dados observados Z e as assumidas identidades dos objetos dadas por J , configurando o problema irrestrito de associação de dados.

Por exemplo, considere o caso onde os dados Z contém algumas detecções do objeto classe "relógio". Se todas as detecções estão associados a um único objeto, as variáveis em θ estão relacionados com um único relógio, por outro lado, se as detecções de "relógio" estão associadas a dois objetos diferentes, existem dois conjuntos de variáveis relacionadas com relógios em θ : um conjunto descreve um relógio, e o outro conjunto descreve o outro relógio. Além disso, mesmo se duas diferentes hipóteses sobre a associação de dados levar a vetores com o mesmo número de objetos de cada classe, não há um modo natural de combinar todos os objetos em ambos os vetores. Essa incompatibilidade de vetores de variáveis dados obtidos considerando diferentes soluções de associação é expressa assumindo-se que, se $P(\theta_1|J_1, Z) > 0$, então $P(\theta_1|J_n, Z) = 0$, para todos $n \neq 1$.

Para resolver o problema de estimar os parâmetros do mapa sem saber o número correto de objetos, a presente tese propõe que cada objeto tenha seus parâmetros calculados separadamente, ou seja, calculando-se a esperança dos parâmetros de cada objeto individualmente em vez de calcular a esperança do mapa como um todo. Para tanto, é necessário desenvolver um critério para identificar o mesmo objeto físico representado nos diferentes vetores θ_J , associando a mesma *identidade* de objeto a eles.

4.2.1 Identidades de objetos

As identidades dos objetos são a solução proposta para o problema de se calcular a esperança dos parâmetros de um mapa com número desconhecido de objetos. Para se calculara esperança dos parâmetros de apenas um objeto separadamente, deve-se somar seus valores esperados assumindo-se diferentes valores de J compatíveis com a existência daquele objeto. Seja ID a variável que indica a identidade de certo

Figura 4.2: Representação de como casar a mesma identidade do objeto em dois mapas gerados assumindo diferentes soluções para a associação de dados. Objetos 1, 3 e 4 têm a mesma identidade em ambas as funções de correspondência representadas em (a) e (b), enquanto que o objeto 2 em (a) e os objetos 2 e 5, em (b), não podem ser igualados se utilizada a atual definição de identidade de objeto.

objeto, o que não depende de uma solução específica da associação de dados. Além disso, seja $J(ID)$ a representação do índice dos parâmetros do objeto o_{ID} no vetor de parâmetros θ_J . Dessa forma, $l_{J(ID)}$ é a localização de o_{ID} representada no vetor θ_J .

Freqüentemente, também é possível fazer o casamento de parte dos objetos em dois vetores θ_J gerados com diferentes funções de correspondência. Para tal, é necessário desenvolver um critério para coincidir o mesmo objeto físico representado nesses vetores de variáveis, atribuindo a mesma identidade para eles. Seguindo a lógica da Seção 3.2.1, que faz a correspondência entre a solução da associação de dados e as identidades dos objetos, este trabalho propõe que a identidade de um único objeto dada uma certa função de correspondência seja definida pela medições associadas a ele. Em outras palavras, se dois casos de J concordam que um certo grupo de medições correspondem a um único objeto, o objeto é exatamente o mesmo em ambos os vetores. Por exemplo, objetos 1, 3 e 4 têm a mesma identidade em ambas as funções de correspondência representadas nas Figuras 4.2a e 4.2b.

A variável de identidade de objeto é definida por $ID \triangleq \{(i,k)_1, \dots, (i,k)_m\}$, um conjunto de pares de índices de medições (i,k) . A função de correspondência J é dita *geradora* de uma identidade de objeto ID se e somente se: $\forall (i,k) \in ID, J(i,k) = j$ e $\forall (i,k) \notin ID, J(i,k) \neq j$, onde $j \in \mathbb{N}^+$ pode ser qualquer índice de objeto. Define-se também $J(ID) \triangleq j | \forall (i,k) \in ID, J(i,k) = j$ como o índice de objeto j que corresponde à identidade ID na função de correspondência J . Note que a definição proposta para identidade de objeto é restritiva: apesar de objetos com a mesma identificação em-

bora em diferentes vetores de variáveis, por certo, referentes ao mesmo objeto físico, os objetos com diferentes ID também podem corresponder a um mesmo objeto físico. Por exemplo, considere objeto 2 na Figura 4.2a e o objeto 2 na Figura 4.2b. Enquanto poderiam ser referido como o mesmo objeto físico utilizando um critério mais liberais, eles não apresentam a mesma identificação, ou seja, as medições são formadas por diferentes subgrupos. Este trabalho não tenciona desenvolver uma definição de identidade que corresponda perfeitamente ao mesmo objeto físico em todos os vetores de variáveis, mas a definição proposta mostrou-se bastante satisfatório para o objetivo desta tese.

Os valores dos parâmetros de certo objeto o_{ID} são definidos por esperanças condicionais, que consideram apenas os valores de J que consideram a existência do objeto com identidade ID :

$$\hat{l}_{ID} \triangleq E[l_{ID}|ID, Z] = \sum_{J \text{ gera } ID} P(J|ID, Z) \int_{\theta_J} l_{J(ID)} P(\theta_J|C, J, Z) d\theta_J \quad (4.21)$$

$$\hat{g}_{ID} \triangleq E[g_{ID}|ID, Z] = \sum_{J \text{ gera } ID} P(J|ID, Z) \int_{\theta_J} g_{J(ID)} P(\theta_J|C, J, Z) d\theta_J \quad (4.22)$$

onde $P(J|ID, Z)$ é a probabilidade condicional de certa função correspondência J considerando-se apenas aquelas que geram a identidade ID . As classes dos objetos c_{ID} são diretamente determinadas a partir das medições $\{a_{i,k} | (i, k) \in ID\}$, que devem todas coincidir sobre a mesma classe detectada sob a suposição de detecção perfeita de objetos.

Because an object identity is yielded by some values of J , but not by others, the belief over the possible values of J implies a certain trust level π_{ID} in that a certain ID actually represent an object in the environment. Thus, we can assign a probability to the event of a certain ID occur in the environment:

Uma vez que uma identidade de objeto é gerada por apenas alguns valores de J , mas não por todos, a crença sobre os possíveis valores de J implica um certo nível de confiança π_{ID} em que uma certa identidade ID de fato represente um objeto no ambiente:

$$\pi_{ID} \triangleq P(ID|Z) = \sum_J P(ID|J, Z) P(J|Z), \quad (4.23)$$

onde $P(ID|J, Z) = \mathbf{1}_J(ID)$ é a função que indica se J gera ID ou não.

Em sentido estrito, (4.23) é a probabilidade marginal de que as medições em ID

de fato correspondam ao mesmo objeto. A interpretação dada aqui é que (4.23) corresponde à probabilidade de que o objeto gerado pelas medições $\{z_{i,k} | (i,k) \in ID\}$ realmente existir no ambiente, e não ser gerado por um grupo de medições espúrias.

4.2.2 Aproximando a crença sobre a associação de dados

Enumerar o espaço de associação de dados não é prático. Uma vez que o número de possíveis associações cresce exponencialmente com o número de medições, como é explicado na Seção 3.2.1, calcular exatamente as somas (4.20)-(4.22) não é viável.

Todavia, a maioria das possíveis soluções para a associação de dados são bastante improváveis, especialmente aquelas que exigem a trajetória do robô ser uma versão muito distorcida daquela obtida utilizando odômetros apenas. Assim, a distribuição de probabilidade das soluções de associação de dados pode ser bem aproximada por uma versão amostrada, concentrada em poucos casos. Então, propõe-se aproximar $P(J|Z) \approx \frac{1}{N_s} \sum_n \delta(J, J_n)$, com N_s sendo o número de amostras de J , e $P(J|ID, Z) \approx \frac{1}{N_s(ID)} \sum_n \delta(J, J_n^{ID})$, onde $N_s(ID)$ é o número de amostras em que a identidade ID ocorre e J_n^{ID} é uma amostra de J que gera a identidade ID . Desse modo, tem-se

$$\pi_{ID} \approx \frac{1}{N_s} \sum_{n=1}^{N_s} \mathbf{1}_{J_n}(ID) \quad (4.24)$$

$$\hat{X} \approx \frac{1}{N_s} \sum_{n=1}^{N_s} \hat{X}_n \quad (4.25)$$

$$\hat{l}_{ID} \approx \frac{1}{N_s(ID)} \sum_{n=1}^{N_s(ID)} \hat{l}_{ID}^n \quad (4.26)$$

$$\hat{g}_{ID} \approx \frac{1}{N_s(ID)} \sum_{n=1}^{N_s(ID)} \hat{g}_{ID}^n \quad (4.27)$$

com

$$\hat{X}_n \triangleq \int_{\theta_{J_n}} X P(\theta_{J_n} | J_n, Z) d\theta_{J_n} \quad (4.28)$$

$$\hat{l}_{ID}^n \triangleq \int_{\theta_{J_n}} l_{J_n(ID)} P(\theta_{J_n} | J_n^{ID}, Z) d\theta_{J_n} \quad (4.29)$$

$$\hat{g}_{ID}^n \triangleq \int_{\theta_{J_n}} g_{J_n(ID)} P(\theta_{J_n} | J_n^{ID}, Z) d\theta_{J_n} \quad (4.30)$$

Algorithm 2 Construção de um O-Mapa a partir de amostras de soluções para associação de dados J_n e medições Z

Entrada: dados Z , e ao conjunto de amostras $\{J_n\}_{n=1}^{Ns}$

Saída: a trajetória estimada $\hat{X}$ e o mapa $\hat{M}$

1. Atribua $ID_list \leftarrow \{\}$
 2. Para n variando de 1 a Ns
 - (a) Calcular o posterior $P(\theta_n|J_n, Z)$, com média $\theta_n^* = (\hat{X}_n, \hat{L}_n, \hat{G}_n)$ e covariância C_θ^n , de acordo com Algoritmo 1
 - (b) Para j variando de 1 a N_n , onde N_n é o número de objetos gerados por J_n
 - i. Determinar ID de forma que $J_n(ID) = j$.
 - ii. Determinar $\hat{l}_{ID}^n$ e $\hat{g}_{ID}^n$ acessando diretamente os valores desses parâmetros em θ_n^*
 - iii. Faça $IDlist \leftarrow ID_list \cup \{ID\}$
 3. Para cada $ID \in ID_list$, encontre π_{ID} , $\hat{l}_{ID}$, e $\hat{g}_{ID}$ usando (4.24), (4.26), e (4.27), respectivamente
 4. Encontre $\hat{X}$ usando (4.25), e $\hat{M} = \{(\hat{l}_{ID}, \hat{g}_{ID}, c_{ID}) | ID \in ID_list\}$
-

Sendo a densidade $P(\theta_{J_n}|J_n, Z)$ assumida como gaussiana, as esperanças $\hat{X}_n$, $\hat{l}_{ID}^n$, e $\hat{g}_{ID}^n$ são simplesmente os parâmetros correspondentes no vetor de variáveis $\theta_n^* \triangleq \arg \max_{\theta_{J_n}} P(\theta_{J_n}|J_n, Z)$, calculado de acordo com a Seção 4.1. O O-Map resultante é definido por $\hat{M} \triangleq \{\hat{o}_{ID} | \pi_{ID} > 0\}$, onde $\hat{o}_{ID} \triangleq (\hat{l}_{ID}, \hat{g}_{ID}, c_{ID})$. O algoritmo para construir um mapa de objetos a partir das amostras J_n considerando a classes dos objetos conhecidas está descrito no Algoritmo 2.

4.3 Tratando detecção imperfeita de objetos

Embora os algoritmos de reconhecimento de objetos têm visivelmente melhorado muito nos últimos anos, ainda não há nenhum algoritmo relatado na literatura que nunca falha. Assim, uma solução prática para SLAM baseado em objeto deve considerar erros do sistema de detecção de objetos e corrigi-los sempre que possível. Este é o caso da configuração 4 do problema, que considera imperfeita a detecção de objetos e a associação de dados, o que corresponde à maior parte dos casos práticos. Neste trabalho, os erros de visão computacional são tratados em um arcabouço probabilístico, sendo solucionados por meio da estimação do mapa provavelmente mais

correto através de inferência probabilística no modelo proposto.

Dois tipos de erros são aqui considerados. O primeiro é a confusão entre duas classes, que pode ocorrer devido à diferença na imagem do objeto causada por alterações na luz ambiente, ruído na imagem, ou outras razões. A outra é a presença de falsas medições devido a falsos positivos na detecção dos objetos. O modelo probabilístico desenvolvido para ambos os erros estão descritos nas seções 4.3.1 e 4.3.2.

4.3.1 Modelo de confusão entre as classes

Até agora, o objeto detectado por um detector especializado em uma determinada classe foi considerado como pertencente a essa classe com probabilidade 1. No entanto, poderá haver classes com exemplares semelhantes, de modo que os objetos pertencentes a uma determinada classe podem ser detectada por algoritmos especializados em objetos de outra classe. Por exemplo, um detector projetado para encontrar cadeiras em imagens poderia, por vezes, detectar uma porta no lugar de uma cadeira. Isso configura o erro de confusão entre as classes.

Mantendo a abordagem adotada na Seção 4.1, segundo a qual a probabilidade de detecção de uma determinada classe de objeto depende apenas de sua própria classe, a verossimilhança dos dados (4.5) é agora dada por

$$P(Z|X, L, G, C, J) = \prod_{i=1}^T \prod_{k=1}^{K_i} \{P(u_{i,k}|x_i, l_{J(i,k)})P(s_{i,k}|x_i, l_{J(i,k)}, g_{J(i,k)})P(a_{i,k}|c_{J(i,k)})\}, \quad (4.31)$$

que agora é uma distribuição multi-modal, onde cada modo corresponde a uma certa combinação de classes de objetos C . No entanto, para um valor específico de C , a forma da densidade posterior (4.6) não é alterada, sendo apenas multiplicada pelo fator de penalidade

$$\prod_{i=1}^T \prod_{k=1}^{K_i} P(a_{i,k}|c_{J(i,k)}). \quad (4.32)$$

Porque não se pode combinar vetores de variáveis θ_J gerados por diferentes soluções J para a associação de dados, como discutido na Seção 4.2, é necessário definir a estimativa de cada parâmetro em face do novo modelo de incerteza. Embora os parâmetros geométricos ainda sejam dados pela esperança, a classe de cada objeto é dada pela estimativa MAP, de acordo com

$$\hat{X} = \sum_J \sum_C P(C, J|Z) \int_{\theta_J} X P(\theta_J|C, J, Z) d\theta_J \quad (4.33)$$

$$(4.34)$$

$$\hat{l}_{ID} \triangleq \sum_{J \text{ gera } ID} \sum_C P(C, J|ID, Z) \int_{\theta_J} l_{J(ID)} P(\theta_J|C, J, Z) d\theta_J \quad (4.35)$$

$$\hat{g}_{ID} \triangleq \sum_{J \text{ gera } ID} \sum_C P(C, J|ID, Z) \int_{\theta_J} g_{J(ID)} P(\theta_J|C, J, Z) d\theta_J \quad (4.36)$$

$$\hat{c}_{ID} \triangleq \arg \max_c \sum_{J \text{ gera } ID} \sum_{C - \{c_{J(ID)}\}} P(C, J|c_{J(ID)} = c, ID, Z), \quad (4.37)$$

onde $C - \{c_{J(ID)}\}$ é o conjunto de variáveis representando todas as classes de objetos com exceção de $c_{J(ID)}$, a classe do objeto $o_{J(ID)}$.

Analogamente ao espaço de soluções de associação de dados, o espaço de classes de objeto também cresce exponencialmente com o número de objetos no mapa. O número de possíveis valores para C dada a função de correspondência J é $|\mathcal{C}|^{N_J}$, onde $|\mathcal{C}|$ é o número de possíveis classes a que podem pertencer um objeto. Assim, o somatório sobre esse espaço é também inviável, e uma versão amostrada de $P(C, J|Z) = P(C|J, Z)P(J|Z)$ é considerada. Uma vez que se espera que as classes dos objetos em questão não sejam tão semelhantes, e os seus exemplares não sejam confundidos com tanta frequência, bem como a distribuição $P(C|J, Z)$ seja também concentrada em torno de alguns valores, de modo que uma versão amostrada deve representá-la bem.

Considerando a aproximação $P(C, J|Z) \approx \frac{1}{N_S} \sum_n \delta(C, C_n) \delta(J, J_n)$, e $P(C, J|ID, Z) \approx \frac{1}{N_S} \sum_n \mathbf{1}_{J_n}(ID) \delta(C, C_n)$, as estimativas (4.33-4.36) ainda são aproximadas por (4.25-4.27), mas agora

$$\hat{X}_n \triangleq \int_{\theta_{J_n}} X P(\theta_{J_n}|C_n, J_n, Z) d\theta_{J_n} \quad (4.38)$$

$$\hat{l}_{ID}^n \triangleq \int_{\theta_{J_n}} l_{J_n(ID)} P(\theta_{J_n}|C_n, J_n, Z) d\theta_{J_n} \quad (4.39)$$

$$\hat{g}_{ID}^n \triangleq \int_{\theta_{J_n}} g_{J_n(ID)} P(\theta_{J_n}|C_n, J_n, Z) d\theta_{J_n} \quad (4.40)$$

Novamente, uma vez que assume-se a densidade $P(\theta_{J_n}|C_n, J_n, Z)$ como gaussiana, as expectativas $\hat{X}_n$, $\hat{l}_{ID}^n$, e $\hat{g}_{ID}^n$ são simplesmente os parâmetros correspondentes no

vetor estimado $\theta_n^* = \arg \max_{\theta_{J_n}} P(\theta_{J_n} | C_n, J_n, Z)$, calculado de acordo com o procedimento descrito na Seção 4.1. A estimativa de c_{ID} é aproximada por

$$\hat{c}_{ID} \approx \arg \max_c \sum_{n=1}^{Ns(ID)} \delta(c, c_{J(ID)}),$$

o que corresponde à classe mais amostrada para certo ID de objeto.

4.3.2 Filtragem de objetos espúrios

Não vale a pena considerar objetos espúrios como parte do mapa se eles puderem ser detectados. Dado que não existem detectores perfeitos de objetos, alguns objetos podem não ser detectados e pode haver algumas falsas medições nos dados Z . Embora o efeito de ignorar objetos não é importante para a nossa abordagem, falsas detecções geram objetos no mapa que não deveriam ser mapeados. Uma vez que estes objetos são geralmente detectados poucas vezes, as suas posições seriam muito incertas, especialmente se não houver conhecimento prévio considerado. Na verdade, o conhecimento prévio sobre os tamanhos de objetos espúrios devem ser sempre desconsiderados, já que diferem consideravelmente do tamanho real do objeto, já que esses objetos, na verdade, não pertencem às classes que geram tais estimativas prévias.

Sendo assim, este trabalho propõe uma forma de identificar objetos espúrios probabilisticamente, baseando-se na incerteza sobre as suas posições no mapa depois de remover o efeito do conhecimento prévio. Se a incerteza é grande para um determinado objeto, a chance de esse objeto ser espúrio é grande. A razão é que os objetos que são freqüentemente detectados, especialmente por diferentes pontos de vista, tem seus parâmetros inferidos com maior grau de certeza. Essa idéia pode ser modelada atribuindo-se uma variável binária de veracidade ξ_j para cada objeto o_j , de forma que $P(\xi_j | C, J, Z)$ depende da incerteza sobre l_j ignorando a crença prévia $P(l_j, g_j | c_j)$ sobre os parâmetros geométricos do objeto.

Seja $\theta_j = (l_j, g_j)$ os parâmetros geométricos do objeto o_j em um certo vetor de variáveis θ assumindo que a solução da associação de dados é dada pela função de correspondência J . Considerando a densidade prévia $P(\theta | C, J) = \prod_{j=1}^{N(J)} P(\theta_j | c_j, J)$ um produto de densidades independentes, a crença posterior sobre um determinado θ_{j_0}

ignorando a prévia é

$$\begin{aligned}
 \bar{P}(\theta_{j^0}|C, J, Z) &= \frac{1}{P(Z|C, J)} \int_{\theta - \{\theta_{j^0}\}} P(Z|\theta, J) \prod_{j=1}^{j^0-1} P(\theta_j|c_j, J) \prod_{j=j^0+1}^{N(J)} P(\theta_j|c_j, J) \\
 &= \frac{1}{P(Z|C, J)} \int_{\theta - \{\theta_{j^0}\}} \frac{P(Z|\theta, J)}{P(\theta_{j^0}|c_{j^0}, J)} \prod_{j=1}^{N(J)} P(\theta_j|c_j, J) \\
 &= P(\theta_{j^0}|C, J, Z) / P(\theta_{j^0}|c_{j^0}, J),
 \end{aligned} \tag{4.41}$$

onde $P(\theta_{j^0}|C, J, Z)$ é a crença a posteriori em θ_{j^0} considerando a prévia, e $\theta - \{\theta_{j^0}\}$ é o vetor contendo as variáveis do mapa e da trajetória, excluindo a posição e geometria do objeto θ_{j^0} . Uma vez que ambas as crenças prévia e posterior θ_{j^0} são assumidas gaussianas, a matriz de covariância de θ_{j^0} ignorando a prévia é dada por

$$\bar{C}_{\theta_{j^0}}^{-1} = C_{\theta_{j^0}}^{-1} - \text{diag}(\{\Gamma(c_{j^0})^{-1}, \Sigma(c_{j^0})^{-1}\}) \tag{4.42}$$

onde $C_{\theta_{j^0}}$ é a covariância a posteriori de θ_{j^0} .

Aqui, a probabilidade de veracidade do objeto é definida como

$$P(\xi_{j^0}|C, J, Z) \triangleq S(\|\bar{C}_{l_{j^0}}\|_2), \tag{4.43}$$

onde $\bar{C}_{l_{j^0}}$ é a covariância posterior de l_{j^0} desconsiderando-se a crença prévia sobre θ_{j^0} , e $\|\cdot\|_2$ retorna a norma-2 do argumento, dada pelo modulo do seu maior autovalor. A função $S(\cdot)$ é a função sigmoidal espelhada, definida como $S: \mathbb{R} \rightarrow (0, 1)$, $S(t) = (1 + \exp\{(t - t_0)/\tau\})^{-1}$, onde τ e x_0 são parâmetros da função. A Figura 4.3 ilustra a forma dessa função.

Para implementar a filtragem de objetos espúrios no arcabouço para a obtenção de mapas de objetos, a variável de não-espuriedade é tida em conta no cálculo da probabilidade π de existência do objeto, de forma que a redefinimos como

$$\pi_{ID} \triangleq \sum_{J \text{ gera } ID} \sum_C P(\xi_{J(ID)}|C, J, Z) P(J|ID, Z), \text{ or} \tag{4.44}$$

$$\pi_{ID} \approx \sum_{n=1}^{Ns(ID)} P(\xi_{J_n(ID)}|C_n, J_n^{ID}, Z), \tag{4.45}$$

no caso de a distribuição em J e C serem aproximadas por uma versão amostrada. O algoritmo para construir um mapa de objetos a partir de amostras $\{(C_n, J_n)\}_{n=1}^{Ns}$ é descrita no Algoritmo 3.

Algorithm 3 Construção de um mapa de objetos a partir de amostras (C_n, J_n)

Entrada: dados Z , e o conjunto de amostras $\{(C_n, J_n)\}_{n=1}^{N_s}$

Saída: a trajetória estimada $\hat{X}$ e o mapa $\hat{M}$

1. Atribua $ID_list \leftarrow \{\}$
 2. Para n variando de 1 a N_s :
 - (a) Calcular a crença posterior $P(\theta|C_n, J_n, Z)$, com média θ_n^* e covariância C_θ^n
 - (b) Para j variando de 1 a N_n
 - i. Determinar ID de forma que $J_n(ID) = j$
 - ii. Determinar $\hat{l}_{ID}^n$ e $\hat{g}_{ID}^n$ acessando diretamente os valores dos parâmetros correspondentes em θ_n^*
 - iii. Se o histograma de classes assumidas por c_{ID} não existe, criá-lo e inicializar todas as entradas com 0
 - iv. Incrementar a entrada correspondente a c_{ID}^n no histograma de classes assumidas por c_{ID}
 - v. Atribua $P(ID|J_n, Z) \leftarrow 1$
 - vi. Atribua $ID_list \leftarrow ID_list \cup \{ID\}$
 3. Para cada $ID \in ID_list$, encontrar:
 - (a) π_{ID} usando (4.45);
 - (b) $\hat{l}_{ID}$ usando (4.26) e (4.39);
 - (c) $\hat{g}_{ID}$ usando (4.27) e (4.40); e
 - (d) $\hat{c}_{ID}$, que é a classe mais freqüente no histograma de classes assumidas por c_{ID}
 4. Encontre $\hat{X}$ usando (4.25) e (4.38), e $\hat{M} = \{(\hat{l}_{ID}, \hat{g}_{ID}, \hat{c}_{ID}, \pi_{ID}) | ID \in ID_list\}$
-

Figura 4.3: Exemplo gráfico de uma função sigmóide espelhada com parâmetros $\tau = 0.1$ e $t_0 = 0.3$.

4.4 Distribuição de probabilidade sobre as soluções da associação de dados e classes dos objetos

Neste trabalho, a distribuição conjunta $P(C, J|Z)$ é expressa usando a lei de Bayes. Considerando que a função de correspondência J e as classes dos objetos C definem um modelo com parâmetros $\theta_J \triangleq (X, L, G)$, a probabilidade do modelo definido por J e C é dada através da marginalização dos seus parâmetros (HOETING et al., 1999)

$$\begin{aligned} P(C, J|Z) &= \int_{\theta_J} P(\theta_J, C, J|Z) \\ &\propto \int_{\theta_J} P(Z|\theta_J, C, J)P(\theta_J|C, J)P(C, J) d\theta_J \end{aligned} \quad (4.46)$$

O termo $P(Z|C, J)P(C, J) = \int_{\theta_J} P(Z|\theta_J, C, J)P(\theta_J|C, J)P(C, J) d\theta_J$ corresponde à crença posterior não normalizada e é chamada *evidência* do modelo definido por J e C (MINKA, 2000). Em esquemas de Bayesian Model Selection (BMS), a evidência é utilizada para selecionar o melhor modelo que se ajusta aos dados disponíveis junto a um pool de candidatos. Se um número suficiente de dados for utilizado, uma versão simplificada para a evidência é utilizada pelo Bayesian Information Criterion (BIC) para selecionar o modelo com complexidade mais adequada (SCHWARZ, 1978).

Neste trabalho, escolher uma única hipótese para a associação de dados não é o caso; em vez disso, a abordagem adotada aqui é o Bayesian Model Averaging (BMA), uma vez que os valores do parâmetro procurados são calculados pela marginalização

de todas as possíveis hipóteses sobre a solução da associação de dados e classes dos objetos (ver (4.33-4.36)). Na abordagem de BMA, a evidência é utilizada como a forma da distribuição alvo $P(C, J|Z)$, utilizada para calcular a expectativa de um determinado parâmetro. Esta abordagem é apoiada por resultados práticos e teóricos, entre os quais aquele que mais motiva a sua utilização é fato de que a evidência do modelo correto tende a tornar-se infinitamente maior do que as outras evidências quando o tamanho do conjunto de dados cresce infinitamente (BERGER; PERICCHI, 1995).

4.4.1 Calculando a evidência

A evidência é calculada por

$$P(Z|C, J) = \int_{\theta_J} P(Z|\theta_J, C, J)P(\theta_J|C, J) d\theta_J, \quad (4.47)$$

que é o mesmo que a integração da da crença posterior não-normalizada (4.6), exceto que agora, porque o objeto detectado pode assumir diferentes classes, ela é penalizada por um fator constante, como discutido na Seção 4.3.1. Apesar de não alterar o procedimento de inferência de SLAM geométrico caso C e J sejam dados, a penalidade devem ser tida em conta para o cálculo da evidência.

Na Seção 4.1.2, o integrando é assumido como sendo proporcional a uma densidade gaussiana, de modo que consideramos a integração de uma função com forma gaussiana. Seja $g(x) = \beta \exp \left\{ -\frac{1}{2} \|x - \mu_x\|_{\Sigma_x}^2 \right\}$ a forma geral de uma função gaussiana multi-variada, onde β é uma constante:

$$\int_x g(x) dx = \beta \sqrt{|2\pi\Sigma_x|} \quad (4.48)$$

Para $\beta = 1$, (4.48) corresponde ao fator de normalização da densidade gaussiana multi-variada. Podemos também isolar a constante β fazendo $x = \mu_x$, resultando em $g(\mu_x) = \beta$.

No nosso caso a função gaussiana a ser integrada é

$$g(\theta_J; C, J) = P(Z|\theta_J, C, J)P(\theta_J|C, J) \quad (4.49)$$

Para calcular a integral desejada, é necessário determinar o valor do pico da distribuição $g(\theta_J^*; C, J)$ e o determinante do parâmetro Σ_θ correspondente. A representação

matricial de (4.49) pode ser re-escrita em termos da fatoração QR $\mathbb{P}^{-\frac{1}{2}}A = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$, de forma que

$$g(\theta_J; C, J) = \frac{\prod_{i=1}^T \prod_{k=1}^{K_i} P(a_{ik} | c_{J(i,k)})}{\sqrt{|2\pi\mathbb{P}|}} e^{-\frac{1}{2}(\|R\delta\theta_J - c\|^2 + \|r\|^2)} \quad (4.50)$$

O valor de pico $g(\theta_J^*; C, J)$ ocorre quando o log negativo de (4.50) é minimizado, o que, como afirmado na Seção 4.1.2, ocorre quando $R\delta\theta_J - c = 0$, deixando $\|r\|^2$ como o resíduo quadrático total. Então,

$$g(\theta_J^*; C, J) = \frac{\prod_{i=1}^T \prod_{k=1}^{K_i} P(a_{ik} | c_{J(i,k)})}{\sqrt{|2\pi\mathbb{P}|}} e^{-\frac{1}{2}(\|r\|^2)} \quad (4.51)$$

que é imediatamente computado se os parâmetros ótimos θ_J^* são determinados via fatoração QR do problema LS linear associado. Ainda mais, uma vez que $g(\theta_J; C, J) \propto P(\theta_J | C, J)$, que é um posterior gaussiano, o parâmetro Σ_θ corresponde à covariância a posteriori, dado por (4.18). Sendo o fator de Cholesky R uma matriz quadrada, o determinante é facilmente calculado se tal fator está disponível:

$$|\Sigma_\theta| = |R|^{-2} = \prod_i r_{ii}^{-2} \quad (4.52)$$

onde r_{ii} é o i -ésimo elemento da diagonal de R .

Para ilustrar o efeito da formulação escolhida para $P(C, J | Z)$, um exemplo simples é proposto. Por razões de simplicidade, o exemplo preocupa-se apenas com a probabilidade de se assumir diferentes soluções para a associação de dados, uma vez que este é o fator que influencia a complexidade do modelo, ou seja, o número de objetos assumidos no ambiente. Considere o caso em que um robô estático tem um único feixe de sonar, sendo capaz de detectar a distância de um obstáculo em frente dele, como demonstrado na Figura 4.4. O robô é colocado na origem das coordenadas, orientado segundo o eixo x . Seu sensor detecta a distância de obstáculos frontais com uma incerteza representada pelo desvio padrão σ_r . Após fazer duas leituras em diferentes instantes de tempo, r_1 e r_2 , o problema de associação de dados corresponde a determinar se as duas leituras correspondem à mesma posição do obstáculo (J_1), ou se o obstáculo se moveu (J_2). Assim, de acordo com a notação definida até agora, $J_1(1, 1) = 1$ e $J_1(2, 1) = 1$, enquanto $J_2(1, 1) = 1$ e $J_2(2, 1) = 2$.

No caso de J_1 corresponder à correta solução da associação de dados, o vetor de variáveis compreende apenas a posição do obstáculo detectado x_o . Assumindo uma

Figura 4.4: Exemplo ilustrativo que representa a probabilidade de associação de dados.

crença a priori gaussiana em x_o , $P(x_o) \sim \mathcal{N}(0, \sigma_o^2)$, sua densidade a posterior é

$$\begin{aligned}
 P(x_o | r_1, r_2, J_1) &= \frac{1}{\sqrt{|2\pi\Sigma_1|}} \exp \left\{ -\frac{1}{2} \|x_o - x_o^*\|_{\Sigma_1}^2 \right\} \\
 &\propto \frac{1}{\sqrt{(2\pi)^3 \sigma_r^4 \sigma_o^2}} \exp \left\{ -\frac{1}{2} \left\| \begin{bmatrix} x_o / \sigma_o \\ (x_o - r_1) / \sigma_r \\ (x_o - r_2) / \sigma_r \end{bmatrix} \right\|^2 \right\} \quad (4.53)
 \end{aligned}$$

com

$$x_o^* = \frac{\sigma_o^2 \sigma_r^2 (r_1 + r_2)}{2\sigma_o^2 \sigma_r^2 + \sigma_r^4} \text{ e } \Sigma_1 = \frac{\sigma_o^2 \sigma_r^4}{2\sigma_o^2 \sigma_r^2 + \sigma_r^4} \quad (4.54)$$

Se, em vez disso, J_2 for a associação de dados correta, $\theta_{J_2} = (x_{o1}, x_{o2})$, compreendendo as posições do obstáculo em dois instantes diferentes. Assumindo independentes crenças a priori em x_{o1} e x_{o2} , com $P(x_{o1}) \sim \mathcal{N}(0, \sigma_o^2)$ e $P(x_{o2}) \sim \mathcal{N}(0, \sigma_o^2)$, a densidade posterior é

$$\begin{aligned}
 P(\theta | r_1, r_2, J_2) &= \frac{1}{\sqrt{|2\pi\Sigma_2|}} \exp \left\{ -\frac{1}{2} \|\theta_{J_2} - \theta_{J_2}^*\|_{\Sigma_2}^2 \right\} \\
 &\propto \frac{1}{\sqrt{(2\pi)^4 \sigma_r^4 \sigma_o^4}} \exp \left\{ -\frac{1}{2} \left\| \begin{bmatrix} x_{o1} / \sigma_o \\ x_{o2} / \sigma_o \\ (x_{o1} - r_1) / \sigma_r \\ (x_{o2} - r_2) / \sigma_r \end{bmatrix} \right\|^2 \right\} \quad (4.55)
 \end{aligned}$$

com

$$x_{o1}^* = \frac{\sigma_r^2 r_1}{\sigma_r^2 + \sigma_o^2}, x_{o2}^* = \frac{\sigma_r^2 r_2}{\sigma_r^2 + \sigma_o^2}, \text{ and } \Sigma_2 = I_{2 \times 2} \frac{\sigma_r^4}{\sigma_r^2 + \sigma_o^2} \quad (4.56)$$

Figura 4.5: Gráfico dos resíduos res_1 e res_2 para os valores de σ_o e σ_r indicados.

Supondo que a distribuição de probabilidade a priori em J é uniforme ($P(J_1) = P(J_2) = 0.5$), as evidências para ambas associações de dados são dadas por

$$P(r_1, r_2 | J_1) = \sqrt{|2\pi\Sigma_1|} P(r_1, r_2 | x_o^*, J_1) P(x_o^*) P(J_1) = \frac{0.5}{2\pi} \frac{1}{\sqrt{2\sigma_o^2\sigma_r^2 + \sigma_r^4}} e^{-\frac{1}{2}res_1^2} \quad (4.57)$$

$$P(r_1, r_2 | J_2) = \sqrt{|2\pi\Sigma_2|} P(r_1, r_2 | x_{o_1}^*, x_{o_2}^*, J_2) P(x_{o_1}^*, x_{o_2}^*) P(J_2) = \frac{0.5}{2\pi} \frac{\sigma_r^2/\sigma_o^2}{\sigma_r^2 + \sigma_o^2} e^{-\frac{1}{2}res_2^2}, \quad (4.58)$$

onde res_1^2 e res_2^2 são os resíduos das normas quadráticas em (4.53) e (4.55), respectivamente.

Embora espera-se que a densidade a priori nos parâmetros $P(\theta | J)$ não influencie a evidência da associação de dados tanto quanto a verossimilhança dos dados $P(Z | \theta, J)$, um prior totalmente desinformativo não pode ser usado. Algebricamente, a razão $P(r_1, r_2 | J_1) / P(r_1, r_2, | J)$ depende diretamente da razão das crenças a priori

$$P(x_o^*) / P(x_{o_1}^*, x_{o_2}^*) = \sqrt{2\pi\sigma_o} \exp \left\{ -\frac{1}{2} \left[(x_o^*)^2 - (x_{o_1}^*)^2 - (x_{o_2}^*)^2 \right] \right\} \quad (4.59)$$

A expressão para os resíduos res_1 e res_2 é muito complexa, e não é mostrada por motivo simplicidade. No entanto, a Figura 4.5 compara os gráficos de ambos os resíduos como função de r_2 , considerando $r_1 = 1m$ e valores razoáveis para σ_r e σ_o . Pode-se observar que o resíduo no caso de obstáculo estático é mínimo quando $r_2 = r_1$. Além disso, assumir J_1 sempre implica um maior resíduo que assumir J_2 , a não ser em uma pequena janela em torno de $r_2 = r_1$. J_1 produz maior resíduo porque gera um modelo mais rígido por restringir ambas as leituras à mesma posição do obstáculo, enquanto J_2 relaxa o modelo assumindo movimento livre do obstáculo.

Para analisar o comportamento da evidência, a Figure 4.6 compara os gráficos das evidências devidas a J_1 e J_2 considerando valores diferentes de σ_o e σ_r . Em todos os gráficos há uma janela em torno de $r_2 = r_1$ onde $P(r_1, r_2 | J_1) > P(r_1, r_2 | J_2)$. De fato,

Figura 4.6: Comparação gráfica do modelo de evidência considerando as soluções da associação de dados J_1 e J_2 , com diferentes desvios padrão.

quanto mais perto r_1 e r_2 estão, mais plausível a hipótese de obstáculo estático é. Essa observação é corroborada pela explicação de Minka (2000) sobre como BMS funciona: quando os dados disponíveis podem ter sido gerados tanto por um modelo mais simples —mais restrito— ou mais complexo —mais relaxado—, o mais simples tem maior evidência. Caso o modelo mais simples não aparenta ter gerado os dados, então o mais complexo é preferido. Então, BMS implementa uma navalha de Occam automática ao selecionar o modelo de maior evidência.

Os desvios padrão σ_r e σ_o também influenciam as regiões onde um modelo tem maior evidência do que outro. Se a incerteza das medições é elevada, torna-se difícil distinguir entre as diferentes posições possíveis do obstáculo, analogamente ao que acontece quando uma pessoa míope tentar distinguir diferentes objetos distantes. Por outro lado, quando as medidas são acuradas, é mais fácil distinguir se o obstáculo mudou de posição, e o modelo assumindo J_2 é preferido a J_1 a menos que r_1 e r_2 sejam bem próximos.

Por sua vez, a influência do desvio padrão do conhecimento a priori sobre a posição do objeto, σ_o , é mais tênue. Normalmente, os valores para σ_o são elevados, uma vez que os dados deverem ser mais relevantes para estimar os parâmetros do que a crença prévia sobre eles. Assim, o fator a priori prévia (4.59) tende a penalizar os modelos mais complexos produzidos assumindo-se J_2 . No limite, quando uma crença totalmente desinformada é utilizada, com $\sigma_o \rightarrow \infty$ e o modelo mais simples, gerado por J_1 , tem probabilidade 1 posterior $P(J_1|Z) = 1$. Portanto, embora o prior sobre os parâmetros não afete muito a inferência dos valores dos parâmetros a partir dos dados, é importante para a seleção do modelo (BERGER; PERICCHI, 1995). Como consequência, poucos priors padrão podem ser utilizados, e alguns cuidados devem ser tomados para definir a densidade a priori. Apesar dos métodos propostos por alguns autores para definir priors padrão objetivos (ver o artigo de Kass & Wasserman (1996) para uma revisão sobre o tema), este trabalho adota a perspectiva subjetiva que $P(\theta_J)$ deve refletir a crença do projetista. Além da relativa estabilidade das evidências com relação a variações sobre σ_o , conforme representado na Figura 4.6, aqui a crença a priori sobre θ_J implica uma escala em que o modelo é observado. Se o conhecimento é pouco informado, a escala de observação é muito grande e a diferença entre os modelos com uma ou duas posições para o obstáculo tornam-se irrelevantes, fazendo a navalha de Occam decidir pelo modelo mais simples. Por outro lado, se a prévia enfoca uma área limitada onde o obstáculo pode estar, as diferenças entre ambos os modelos são mais perceptíveis, e J_2 passa a ter maior probabilidade caso ambas as posições dos obstáculos coincidam dentro dessa área.

5 Amostragem por MCMC

Este capítulo é dedicado a mostrar como amostrar a distribuição de probabilidades sobre as soluções de associação de dados e classes de objetos, definida na Seção 4.4. Neste trabalho foi adotada a abordagem MCMC, uma vez que existem razões teóricas e práticas para acreditar que é um método promissor para realizar inferência aproximada, especialmente no espaço combinatório de associação de dados (DELLAERT, 2001; RANGANATHAN, 2008).

Métodos MCMC funcionam simulando uma cadeia de Markov sobre o espaço de estados a ser amostrado, com a propriedade de convergir por fim à distribuição de interesse. Cadeias de Markov são processos estocásticos ao longo do tempo, representadas por uma seqüência de variáveis de estado aleatórias $\{S_0, S_1, \dots\}$ que assumem valores obedecendo a propriedade de Markov. Esta propriedade indica que a probabilidade de valores de estado futuros não dependem de estados passados. Assim, as propriedades de uma cadeia Markov são descritas pelas suas probabilidades de transição $P(S_{t+1}|S_t)$, que são geralmente as mesmas, independentemente do tempo t .

Algumas classes de cadeias de Markov têm a propriedade de convergir para uma distribuição estacionária. Isto significa que, para essas cadeias, a distribuição marginal sobre o valor de estado atual, $P(S_t)$, não depende do tempo se t for suficientemente grande, independentemente dos seus estados iniciais. Neste caso, se o espaço de estados corresponde àquele que pretendemos amostrar, utilizar os estados da cadeia de Markov como amostras garante que elas serão sorteadas da distribuição alvo. Métodos MCMC utilizam esta propriedade definindo uma cadeia de Markov que fornece amostras de uma certa distribuição de interesse. Não é preciso se preocupar em definir explicitamente os parâmetros da cadeia de Markov para utilizar técnicas MCMC.

5.1 O algoritmo de Metropolis-Hastings

Para entender como a abordagem MCMC funciona, vamos introduzir o algoritmo Metropolis-Hastings (MH) (HASTINGS, 1970), que é a base de todos os métodos MCMC (GILKS; RICHARDSON; SPIEGELHALTER, 1996a). O algoritmo MH implementa uma cadeia de Markov, propondo novos estados e aceitando-os ou não de acordo com um critério estatístico. O estado atual da cadeia é retornado como uma amostra a cada transição de estados, geralmente após um período de *burn-in*, no qual se considera que a cadeia está um regime transitório.

Considere uma cadeia Markov cujo espaço de estados corresponde àquele de que desejamos obter amostras. Dado o estado atual da cadeia S_t , o algoritmo MH funciona aceitando ou rejeitando um novo estado proposto S' gerado por uma função propositora, que implementa um algoritmo (normalmente simples) para amostrar de uma distribuição de propostas sobre os estados, $q(S_t \rightarrow S')$. O estado proposto é aceito com probabilidade $\alpha = \min\left(1, \frac{P(S')q(S' \rightarrow S_t)}{P(S_t)q(S_t \rightarrow S')}\right)$, onde $P(S)$ é a distribuição alvo da qual desejamos obter amostras. Note que, além da garantia teórica da amostragem da distribuição de probabilidades correta quando t é suficientemente grande, o algoritmo MH requer apenas que se possa computar uma versão não normalizada de $P(S)$, uma clara exigência para o problema atacado nesta tese (veja a formulação da distribuição de probabilidade alvo (4.46)).

Para garantir que a cadeia de Markov definida pelo algoritmo MH convirja para a distribuição estacionária desejada sobre os estados, é necessário observar alguns requisitos da distribuição proposta (ROBERTS, 1996). O primeiro é que as transições de estado propostas devem ser reversíveis, i.e. $q(S_t \rightarrow S') > 0 \Rightarrow q(S' \rightarrow S_t) > 0$. O outro é a possibilidade de se atingir qualquer estado com probabilidade 1 após um número finito de transições, independentemente do estado inicial S_0 , tornando a cadeia irredutível. Em outras palavras, a distribuição de propostas deve permitir passar de qualquer estado a qualquer outro por uma sequência finita de estados propostos.

Para implementar o algoritmo MH, é necessário definir duas distribuições de probabilidade: a distribuição alvo — apenas proporcionalmente —, e a de propostas, que descreve como a função propositora seleciona estados para propor. Embora distribuição alvo seja específica para o problema estudado, não existem restrições à distribuição de propostas além da condição de reversibilidade e a garantia de irredutibilidade da cadeia. No entanto, embora a frequência das amostras extraídas da cadeia de

Algorithm 4 O algoritmo Metropolis-Hastings para amostrar $P(C, J|Z)$

1. Iniciar a cadeia com um estado válido (C_0, J_0) e definir o índice de amostra $n \leftarrow 0$
2. Para t variando de 1 a $Nb + Ns$, onde Ns é o número desejado de amostras e Nb é o período de burn-in, fazer:
 - (a) Propor um novo estado (C', J') de acordo com uma distribuição de propostas adequada $q(C_t, J_t \rightarrow C', J')$
 - (b) Calcular a razão de aceitação

$$\alpha = \min \left(1, \frac{P(Z|C', J')P(C', J')q(C', J' \rightarrow C_t, J_t)}{P(Z|C_t, J_t)P(C_t, J_t)q(C_t, J_t \rightarrow C', J')} \right) \quad (5.1)$$

- (c) Com probabilidade α , aceitar (C', J') e definir $C_{t+1} \leftarrow C'$ e $J_{t+1} \leftarrow J'$, caracterizando uma auto-transição na cadeia
- (d) Se $t \geq Nb$, definir o índice amostra $n \leftarrow n + 1$ e atribuir a amostra $(C_n, J_n) \leftarrow (C_{t+1}, J_{t+1})$
- (e) Atribua $t \leftarrow t + 1$

estado consecutivas obedecerá $P(S)$ quase certamente apenas quando o número de amostras vai ao infinito, queremos que esta distribuição seja bem representada pelo menor número possível amostras, já que determiná-las exige grande esforço computacional.

Distribuições de propostas bem-projetadas podem ajudar a reduzir o número necessário de amostras ao propor transições de estado que tenham mais chance de serem aceitas, ao explorar o espaço de estados visitando os estados mais prováveis mais freqüentemente. Assim, boas distribuições de propostas são projetadas na tentativa de imitar a distribuição alvo, tornando a razão de aceitação (5.1) mais próxima de 1 do que as de propósito geral. Idealmente a distribuição de propostas seria a própria distribuição alvo; entretanto, se amostragem direta fosse uma opção a abordagem MCMC se tornaria desnecessária. Normalmente, o cálculo da distribuição alvo é computacionalmente proibitivo, especialmente quando ela é conhecida apenas aquém de um fator de escala, pois o cálculo deste fator exige uma soma sobre todo o espaço de variáveis. Por outro lado, se distribuições de propostas muito simples e computacionalmente baratas forem utilizadas, como a distribuição uniforme utilizada no algoritmo de *random-walk* (METROPOLIS et al., 1953), podem ser necessárias tantas amostras que a abordagem MCMC se torna inviável.

Como os métodos MCMC são baseados em executar uma cadeia de Markov so-

bre o espaço a ser amostrado, alguma noção de vizinhança de estados deve ser definida, ao menos implicitamente, para implementar a função propositora. Assim, dado o estado atual da cadeia de estados, os estados proponíveis definem tal vizinhança. Há evidências de que as soluções associação de dados plausíveis — aquelas com probabilidade não desprezível — são similares entre si, e são colocadas em uma vizinhança relativamente pequena se um certo tipo de função propositora for utilizado (RANGANATHAN, 2008). Isto é uma motivação extra para utilizar a abordagem MCMC para tomar amostras do espaço de soluções de associação de dados e classes de objetos.

5.2 Distribuição de propostas

A função de correspondência J , que define a solução para o problema de associação de dados, pode ser vista como uma partição do conjunto de detecções de objeto, onde cada partição dá origem a um objeto. A mesma idéia foi usada para definir a identidade do objeto em diferentes funções de correspondência na Seção 4.2.1, partindo do princípio de que a igualdade de partições sempre corresponde ao mesmo objeto físico. Daí que a definição do problema de associação de dados neste trabalho, e o problema de SLAM em geral, esteja intimamente relacionada com o problema de mapeamento topológico, onde um conjunto de detecções de marcos tem que ser particionado em distintas localidades. Nos mapas topológicos, marcos são locais especiais onde o robô já passou, ligados entre si por uma relação de traversibilidade. Eles são mais comumente representados por um grafo onde os nós são os marcos e as arestas representam um caminho de um marco para o outro. Os locais específicos dos marcos não são geralmente muito importantes, de modo a que o objectivo final do mapeamento topológico é encontrar a topologia do grafo.

Devido ao seu sucesso em amostrar o espaço de mapas topológicos, a distribuição de propostas utilizada no presente trabalho é inspirada pelo trabalho em Probabilistic Topological Maps (PTMs) (RANGANATHAN, 2008). Nesse trabalho, um método MCMC amostra o espaço de topologias, rendendo um histograma de probabilidades de possíveis mapas, motivado pela possível falta de dados de entrada, que pode ser insuficiente para determinar uma única boa topologia.

5.2.1 Transição de estados proposta

No trabalho sobre PTMs, uma nova topologia é proposta de acordo com uma simples transformação da atual, chamada de *movimento*. Há dois movimentos possíveis: o movimento de fusão, que funde dois subconjuntos de marcos detectados em uma única localidade, ou o movimento de divisão, que divide o subconjunto de marcos observados correspondente a uma determinada localidade em dois subconjuntos disjuntos. A mesma estratégia é adotada neste trabalho, que funde ou divide subconjuntos de detecções de objeto. Por razões de simplicidade, a referência à fusão ou divisão de objetos é usado para expressar o ato de fusão ou divisão de subconjuntos de detecções de objetos, que dão origem a diferentes objetos em um mapa.

Embora o tipo de transição de estados utilizada aqui seja semelhante ao proposto por Ranganathan (2008), este trabalho também considera que o caso de classes dos objetos desconhecidas, o que exige a mostragem de mais esse espaço também. Além disso, aqui uma maneira diferente de escolher os objetos a serem divididos ou fundidos é proposta, mesmo no caso de detecção perfeita de objetos. A presente tese preocupa-se apenas com o uso de distribuições de propostas direcionadas pelos dados, o que significa que os dados de entrada são sempre usados para propor novos movimentos. Veja o trabalho de (RANGANATHAN, 2008) para um estudo comparativo de como distribuições direcionadas e não direcionadas por dados influenciam o desempenho dos métodos de MCMC na aproximação da distribuição de interesse.

A função propositora escolhem um de três movimentos possíveis: *fusão*, *divisão* ou *troca de classes*, com probabilidades P_{merge} , P_{split} , e P_{switch} , respectivamente. Cada movimento é projetado para tentar imitar o comportamento da distribuição alvo, $P(C, J|Z)$, discutida na Seção 4.4.1. Como regra geral, a função propositora tenta propor amostras boas (de alta probabilidade) a maior parte do tempo, permitindo que outras não tão boas sejam propostas algumas vezes. No entanto, controlar quando uma proposta é boa ou não é difícil em geral. Construir heurísticas que façam isso muito bem podem ser quase tão custosas quanto amostrar diretamente da distribuição desejada, então aqui os movimentos propostos aplicam somente algumas regras intuitivas que não podem propor boas e más amostras a seu bel prazer.

A regra intuitiva para o movimento de fusão é que objetos próximos devem poder ser fundidos sem o aumento significativo do resíduo das medidas, ao mesmo tempo em que simplifica o modelo, provavelmente gerando bons modelos. O movimento de fusão também favorece a fusão de objetos pertencentes à mesma classe, uma vez

que misturar detecções de diferentes classes de objetos geram modelos altamente penalizados, de acordo com a formulação da verossimilhança (4.31). O procedimento adotado para propor movimentos de fusão está descrito no Algoritmo 5.

O resíduo das medições refletem a diferença entre as medições reais Z e as preditas dado o mapa inferido e a trajetória. É mais ou menos equivalente ao resíduo do sistema super-determinado (4.15) usado para resolver o problema geométrico de SLAM. Embora, o resíduo do sistema é também influenciado pelas diferenças entre a crença prévia sobre os parâmetros geométricos do objeto e aqueles inferido a porção dos resíduos devidos a essas diferenças é usualmente irrelevante quando comparada com a porção causada pelas diferenças nas medições.

O movimento de divisão é projetado com base na idéia de que objetos responsáveis por uma grande parcela do resíduo estão provavelmente forçando muito o modelo, e dividi-los provavelmente gerará boas amostras. O resíduo por objeto é calculado tirando a norma do resíduo de todas as equações do sistema que descrevem as medidas de um certo objeto, dado por

$$\mathbf{r}^2(j) = \sum_{i=1}^T \sum_{\substack{k_i=1 \\ J_n(i, k_i) = j}}^{K_i} \left[\|U_{i,k_i}^x \delta x_i^* + U_{i,k_i}^l \delta l_j^* + h^u(x_i^0, l_j^0) - u_{i,k_i}\|_{R_{i,k_i}}^2 + \|S_{i,k_i}^x \delta x_i^* + S_{i,k_i}^l \delta l_j^* + S_{i,k_i}^g \delta g_j^* + h^s(x_i^0, l_j^0, g_j^0) - s_{i,k_i}\|_{W_{i,k_i}}^2 \right], \quad (5.2)$$

onde δx_i^* , δl_j^* , e δg_j^* são as variações da trajetória e dos parâmetros geométricos dos objetos em torno do seu ponto de linearização dada a função de correspondência J e os dados. Contudo, o objeto é dividido aleatoriamente. O Algoritmo 6 explica como realizar um movimento de divisão.

Propor um movimento de troca de classes, que explora o espaço de classes dos objetos é mais fácil. Para uma dada função de correspondência, J , o comportamento da distribuição alvo $P(C, J|Z)$ (4.46) é fortemente regido pelo fator de penalidade devido à confusão entre classes (4.32), que é factível de ser calculada. Além do mais, objetos fortemente penalizados são mais passíveis de pertencer à classe errada do que os fracamente penalizados. Assim, o movimento de troca de classes tende a escolher um objeto altamente penalizado e mudar sua classe para uma mais conveniente, como descrito no Algoritmo 7.

Algorithm 5 Proposição de um movimento de fusão

1. Seja $(X_t^*, M_t^*) = \theta_t^* = \arg \max_{\theta} P(\theta | C_t, J_t, Z)$ o mapa e trajetória obtidas usando o estado atual (C_t, J_t) .
2. Determinar $\mathbf{d}_t = \{d_j\}_{j=1}^{N_t}$, onde N_t é o número de objetos em M_t^* e d_j é o número de objetos da mesma classe que $o_j \in M_t^*$ que são mais próximos que um certo limiar de distância (por ex., 50 cm).
3. Transformar $\mathbf{d}_t$ em uma distribuição de probabilidade através de normalização e amostrar um índice de objeto j_1 .
4. Determinar $\mathbf{e}_t = \{\exp(-\frac{1}{2}e_j)\}_{j=1, j \neq j_1}^{N_t}$, onde $e_j = \frac{\|l_j - l_{j_1}\|^2}{\sigma_E^2}$, com o comprimento σ_E escolhido apropriadamente. Se $c_j \neq c_{j_1}$, multiplique e_j por um fator de penalidade (e. g., 3).
5. Se $\mathbf{e}_t$ é vazio, re-proponha $J' = J_t$ e $C' = C_t$, e retornar.
6. Transformar $\mathbf{e}_t$ em uma distribuição de probabilidade através de normalização e amostrar um índice de objeto j_2 .
7. Proponha uma nova função de correspondência J' através da fusão dos objetos o_{j_1} e o_{j_2} em o_{j_1} fazendo, para todos $(i, k_i) \in \{i \times \mathcal{K}, i = 1, \dots, T\}$:
 - Se $J_t(i, k_i) < \max(j_1, j_2)$, $J'(i, k_i) = J_t(i, k_i)$
 - Se $J_t(i, k_i) = \max(j_1, j_2)$, $J'(i, k_i) = \min(j_1, j_2)$
 - Se $J_t(i, k_i) > \max(j_1, j_2)$, $J'(i, k_i) = J_t(i, k_i) - 1$
8. Calcular a distribuição de propostas de classes

$$q(c_{j_1} | J', Z) \propto \prod_{i=1}^T \prod_{\substack{k_i=1 \\ J'(i, k_i) = j_1}}^{K_i} P(a_{i, k_i} | c_{j_1}).$$
9. Amostrar a nova classe c'_{j_1} de acordo com $q(c_{j_1} | J', Z)$.
10. Calcular a probabilidade da transição proposta: $q(C_t, J_t \rightarrow C', J') = P_{\text{merge}} \frac{\mathbf{d}_t(j_1)}{\sum_{j=1}^{N_t} \mathbf{d}_t(j)} \frac{\mathbf{e}_t(j_2)}{\sum_{j=1}^{N_t} \mathbf{e}_t(j)} q(c'_{j_1} | J', Z)$.
11. Uma vez que a transição reversa só pode ser obtida através de um movimento de divisão, calcular sua probabilidade $q(C', J' \rightarrow C_t, J_t)$ perfazendo os passos 1, 2, 6, e 8, assumindo $j_{\text{split}} = \min(j_1, j_2)$.

Algorithm 6 Proposição de um movimento de fusão

1. Seja $(X_t^*, M_t^*) = \theta_t^* = \arg \max_{\theta} P(\theta | C_t, J_t, Z)$ o mapa e a trajetória obtidos usando o estado atual (C_t, J_t) .
2. Calcular o resíduo total por objeto $\mathbf{r}_t^2$ usando (5.2)
3. Transformar $\mathbf{r}_t^2$ em uma distribuição de probabilidade através da sua normalização, e use-a para amostrar o índice j_{split} .
4. Seja $\Psi \leftarrow \{(i, k_i) | J_t(i, k_i) = j_{split}\}$ o subconjunto de medições que estão atribuídas ao objeto o_j . Particione Ψ em dois conjuntos disjuntos Ψ_1 e Ψ_2 de forma que:
 - O índice da medição $(i_{min}, 1)$ é atribuído ao subconjunto Ψ_1 , com $i_{min} = \min\{i | \exists k_i \in \mathcal{K}_i, J_t(i, k_i) = j_{split}\}$
 - Uma medição aleatoriamente escolhida $(i_2, k_{i_2}) \in \Psi - \{(i_{min}, 1)\}$ é atribuída ao subconjunto Ψ_2
 - Os elementos restantes $(i, k_i) \in \Psi - \{(i_{min}, 1), (i_2, k_{i_2})\}$ são aleatoriamente divididos entre Ψ_1 e Ψ_2
5. Propor uma nova solução para a associação de dados J' dividindo os subconjuntos de medições atribuídas a $o_{j_{split}}$ fazendo, para todo $(i, k_i) \in \{i \times \mathcal{K}_i, i = 1, \dots, T\}$:
 - Se $(i, k_i) \notin \Psi$, $J'(i, k_i) = J_t(i, k_i)$
 - Se $(i, k_i) \in \Psi_1$, $J'(i, k_i) = j_{split}$
 - Se $(i, k_i) \in \Psi_2$, $J'(i, k_i) = N_t + 1$
6. Calcular a distribuição de propostas de classes

$$q(c_{j_{split}} | J', Z) \propto \prod_{i=1}^T \prod_{\substack{k_i=1 \\ J'(i, k_i) = j_{split}}}^{K_i} P(a_{i, k_i} | c_{j_{split}}) \quad \text{e}$$

$$q(c_{N'} | J', Z) \propto \prod_{i=1}^T \prod_{\substack{k_i=1 \\ J'(i, k_i) = N'}}^{K_i} P(a_{i, k_i} | c_{N'}), \quad \text{onde } N' \text{ é o número de objetos no}$$
 mapa obtido usando a função de correspondência proposta J' .
7. Amostre as novas classes $c'_{j_{split}}$ e $c'_{N'}$ de acordo com $q(c_{j_{split}} | J', Z)$ e $Pq(c_{N'} | J', Z)$, respectivamente.
8. Calcular a probabilidade da transição proposta: $q(C_t, J_t \rightarrow C', J') = P_{split} \frac{\mathbf{r}_t^2(j_{split})}{\sum_{j=1}^{N_t} \mathbf{r}_t^2(j)} \frac{1}{2^{|\Psi|-1}} q(c'_{j_{split}} | J', Z) Pq(c'_{N'} | J', Z)$.
9. Uma vez que o movimento reverso só pode ser obtido através de um movimento de fusão, calcule sua probabilidade $q(C', J' \rightarrow C_n, J_n)$ executando os passos 1, 2, 4, 8, e 10 do Algoritmo 5 duas vezes: uma vez considerando $j_1 = j_{split}$ e $j_2 = N'$, resultando $q_1(C', J' \rightarrow C_n, J_n)$, e outra vez considerando $j_1 = N'$ e $j_2 = j_{split}$, resultando $q_2(C', J' \rightarrow C_t, J_t)$. Então some ambas as probabilidades $q(C', J' \rightarrow C_t, J_t) = q_1(C', J' \rightarrow C_t, J_t) + q_2(C', J' \rightarrow C_t, J_t)$.

Algorithm 7 Proposição do movimento de troca de classes

1. Seja $(X_t^*, M_t^*) = \theta_t^* = \arg \max_{\theta} P(\theta | C_t, J_t, Z)$ o mapa e a trajetória obtidos usando o estado atual (C_t, J_t) .
 2. Propor $J' = J_t$.
 3. Para j variando de 1 a N_t , calcular o vetor $\mathbf{p}_t(j) = 1 - \prod_{i=1}^T \prod_{\substack{k_i=1 \\ J_t(i, k_i)=j}}^{K_i} P(a_{i, k_i} | c_j^t)$
 4. Transformar $\mathbf{p}_t$ em uma distribuição de probabilidade através de sua normalização e amostrar um índice j_{switch} .
 5. Calcular a distribuição de propostas de classes $q(c_{j_{switch}} | J', Z) \propto \prod_{i=1}^T \prod_{\substack{k_i=1 \\ J'(i, k_i)=j_{split}}}^{K_i} P(a_{i, k_i} | c_{j_{switch}})$
 6. Amostrar a nova classe $c'_{j_{switch}}$ de acordo com $q(c_{j_{switch}} | J', Z)$
 7. Calcular a probabilidade da transição proposta: $q(C_t, J_t \rightarrow C', J') = P_{switch} \frac{\mathbf{p}_t(j_{switch})}{\sum_{j=1}^{N_t} \mathbf{p}_t(j)} q(c'_{j_{switch}} | J', Z)$.
 8. Uma vez que o movimento reverso só pode ser obtido através de outro movimento de troca de classes, calcular sua probabilidade $q(C', J' \rightarrow C_t, J_t)$ efetuando os passos 1,3,5, e 7, assumindo o mesmo j_{switch} .
-

5.3 Questões práticas sobre o algoritmo de amostragem proposto

Devido à natureza geral dos métodos de MCMC, seu desempenho depende fortemente da aplicação específica no qual será utilizado. Mais especificamente, depende do comportamento da cadeia de Markov subjacente e do custo computacional de executar as transições de estados. Uma questão importante para a eficiência da implementação do MCMC é a definição da distribuição de propostas com base nos dados, discutida na Seção 5.1, o qual incentiva a cadeia a explorar os estados mais prováveis com probabilidade maior.

Uma consequência importante da concepção de uma boa função propositora é a melhora da *mixagem* da cadeia. A taxa de mixagem é uma medida de quão rápido a cadeia escolhe entre os estados não negligenciáveis (GILKS; RICHARDSON; SPIEGELHALTER, 1996a) — veja o artigo de Tierney (1996) para definições formais. Indiretamente, ela indica o quão rápido a distribuição resultante, que consiste no histograma de amostras normalizado, aproxima a distribuição alvo quando o número de amostras cresce. Assim, uma cadeia com uma boa mixagem alterna entre os estados de maior probabilidade na maior parte do tempo, e os histogramas de amostras se parecem com a distribuição alvo de forma muito rápida.

A eficiência da geração e da avaliação das amostras e a condução da cadeia para um estado de convergência são outros aspectos importantes na implementação dos métodos MCMC. A cadeia é considerada convergida se a probabilidade marginal de amostragem de um determinado estado não depende mais do estado inicial. No caso do espaço de estados incluir o espaço das soluções de associação dos dados, a convergência da cadeia está associada com a vizinhança onde a maioria das soluções são não negligenciáveis. No entanto, mesmo depois da cadeia convergir, infinitas amostras são necessárias para imitar perfeitamente a distribuição. Assim, um grande número de amostras é necessário para aproximar a distribuição posterior alvo bem o suficiente, de modo que métodos eficientes para gerar e avaliar amostras são necessários para tornar o processo de amostragem computacionalmente viável.

Por último, como qualquer método de amostragem, técnicas de MCMC exigem a definição de um número de amostras a serem obtidas. A regra geral consiste em garantir que o histograma de amostras normalizado se aproxime da distribuição alvo bem o suficiente, o que implica que o significado de "bem o suficiente" também deve ser

especificado (GILKS; RICHARDSON; SPIEGELHALTER, 1996a). A forma mais óbvia, mas errada seria pensar em definir quando o MCMC deve parar de amostrar comparando o histograma de amostras normalizado diretamente com a distribuição alvo normalizada calculada nos estados amostrados e normalizada. No entanto, ambas as funções podem ser muito similares no início do processo de amostragem – por exemplo, quando apenas um ou dois estados diferentes foram amostrados – e divergirem mais tarde, convergindo novamente com muitas mais amostras. Um procedimento correto e objetivo corresponde à execução de várias cadeias em paralelo, partindo de diferentes estados, e comparando seus histogramas de amostras resultantes. Quando todos os histogramas forem similares uns aos outros, o processo de amostragem deve parar (GILKS; RICHARDSON; SPIEGELHALTER, 1996a). Porém, esta abordagem não se encaixa bem no problema abordado neste trabalho, uma vez que calcular e avaliar transições de estados é computacionalmente custoso. Neste trabalho, nenhum critério fortemente embasado é usado para determinar o número de amostras, o qual é escolhido definindo-se que quando o espaço de estados é maior, mais amostras são necessárias para representar a distribuição posterior.

5.3.1 Melhorando a mixagem com o algoritmo Metropolis-Coupled

Embora técnicas de MCMC construam cadeias Markov com a propriedade de terem sua convergência assegurada, conseguir boas taxas de mixagem é bastante difícil, especialmente quando a distribuição alvo tem grandes picos e vales, como mostrado na Figura 5.1 (RANGANATHAN, 2008). Se o projetista sabe a priori quais estados estão nos picos de probabilidade, conceber distribuições de proposta que amostrassem esses picos iria produzir cadeias com boa mixagem. No entanto, técnicas de MCMC são frequentemente utilizadas como "último recurso", motivado em parte pela ignorância sobre a forma da distribuição, — de outro modo seria possível amostrá-la diretamente — de modo que obter uma cadeia com boa mixagem uma missão desafiadora.

Uma forma intuitiva para melhorar a mixagem, neste caso, é atenuar o efeito dos picos e vales na distribuição alvo, permitindo ao amostrador mover-se por seu suporte mais livremente. Isso pode ser feito por "aquecimento" da distribuição, que consiste na definição de uma nova distribuição alvo $P_{heat}(S) = P(S)^{\frac{1}{T}}$, onde T é uma temperatura relativa. Quanto mais alta a temperatura T é, mais atenuados os picos e vales são. Este processo é referido como *têmpera simulada*, por analogia ao processo utilizado

Figura 5.1: Ilustração de como o algoritmo MH amostra distribuições com picos de modo a gerar cadeias com mixagem ruim. (a) Densidade alvo, com picos centrados em $x = 4$ e $x = 12$. (b) Gráfico das amostras extraídas pelo algoritmo MH iniciando em $x = 4$ e propondo novos estados dentro de uma janela fixa em torno do estado atual. Os estados pertencendo ao pico à direita não são amostrados, indicando uma mixagem ruim. As figuras foram extraídas de (RANGANATHAN, 2008).

em Mecânica dos Materiais para tornar o aço mais rígido através de aquecimento intenso e arrefecimento rápido dele. A amostragem da distribuição aquecida melhora a mixagem da cadeia.

No entanto, não é da distribuição aquecida que se quer amostras. Para tirar partido das suas propriedades de melhor mixagem, podemos usar o algoritmo Metropolis-Coupled MCMC (também conhecido como MCMCMC ou MC^3) (GEYER, 1991; GILKS; ROBERTS, 1996). Ele funciona através da execução de um certo número de cadeias em paralelo, eventualmente trocando estados entre eles. Esse algoritmo é totalmente derivado do Metropolis-Hastings, e tem o propósito de tirar partido das propriedades de diferentes cadeias para amostrar a distribuição desejada, incluindo o uso de cadeias resultantes da aplicação do algoritmo MH a versões aquecidas da distribuição alvo como caso particular. O algoritmo MC^3 aplicado a tempera simulada está descrito no Algoritmo 8, enquanto a Figura 5.2 ilustra seu funcionamento.

Note apenas que cadeias geradas por distribuições com grandes diferenças de temperatura entre si podem dificilmente trocar seus estados, de forma que a versão tipo tempera simulada do algoritmo de MC^3 funciona melhor caso pequenas diferenças de temperatura sejam usadas (GILKS; ROBERTS, 1996). Esse efeito pode ser entendido de modo intuitivo: cadeias mais frias tendem a se concentrar em estados com maior

Figura 5.2: Ilustração de como MC^3 com cadeias aquecidas ajudam a melhorar a mixagem quando distribuições com picos são amostradas. (a) Densidade alvo em temperaturas diferentes: $T = 1$ (linha preta), representando a densidade original, a mesma da Figura 5.1, $T = 11$ (azul), e $T = 21$ (vermelha). (b) Gráfico das amostras extraídas de pelo algoritmo MC^3 , usando o mesmo estado inicial e densidade de propostas do exemplo da Figura 5.1. Neste caso, o amostrador rapidamente alterna entre ambos os picos, indicando boa mixagem da cadeia. As figuras foram extraídas de (RANGANATHAN, 2008).

probabilidade, enquanto as mais quentes movem-se sobre o espaço de estados mais livremente. de forma que a troca de estados tende a ser pouco provável. Se essa tendência for exacerbada, o movimento de troca de estados pode ficar extremamente difícil de ocorrer.

5.3.2 Melhorando a eficiência da transição de estados

Para propor e avaliar novas amostras, o problema geométrico do SLAM deve ser resolvido por duas vezes: uma vez usando a solução de associação de dados a classes de objetos do estado atual, a outra vez usando os mesmos parâmetros do estado proposto. A primeira solução é utilizada para calcular as estruturas auxiliares necessárias para propor uma nova amostra, de acordo com os Algoritmos 5, 6, e 7, e para avaliar o estado atual. A segunda solução é usada para calcular as mesmas estruturas auxiliares, necessárias para calcular a probabilidade do movimento inverso e para avaliar o estado proposto.

No entanto, resolver o problema geométrico de SLAM duas vezes para cada estado de transição é uma abordagem ingênua. Sempre que um novo estado é aceito,

Algorithm 8 O algoritmo Metropolis-Coupled MCMC aplicado à amostragem de $P(C, J|Z)$

1. Inicie N_c cadeias com um estado válido (C_0, J_0) , e inicie o índice de amostras $n \leftarrow 0$
2. Para t variando de 1 a $N_b + N_s$, onde N_s é o número de amostras desejadas e N_b é o número de amostras de *burn-in*, faça:
 - (a) Para cada cadeia i variando de 1 a N_c , faça
 - i. Calcular o expoente de aquecimento $\beta_i = \frac{1}{1 + \Delta T \cdot (i-1)}$, onde ΔT é um incremento de temperatura fixo
 - ii. Execute um passo do algoritmo de MH, correspondendo aos passos a–c no Algoritmo 4, considerando a distribuição alvo não normalizada como $(P(Z|C, J)P(C, J))^{\beta_i}$ em vez $P(Z|C, J)P(C, J)$.
 - (b) Incremente $t \leftarrow t + 1$
 - (c) Para cada par de cadeias $(i, i-1)$, começando com $i = N_c$, troque os estados com probabilidade

$$\alpha_{\text{swap}} = \min \left(1, \frac{(P(Z|C_t^i, J_t^i)P(C_t^i, J_t^i))^{\beta_{i-1}} (P(Z|C_t^{i-1}, J_t^{i-1})P(C_t^{i-1}, J_t^{i-1}))^{\beta_i}}{(P(Z|C_t^i, J_t^i)P(C_t^i, J_t^i))^{\beta_i} (P(Z|C_t^{i-1}, J_t^{i-1})P(C_t^{i-1}, J_t^{i-1}))^{\beta_{i-1}}} \right) \quad (5.3)$$

onde (C_t^i, J_t^i) é o estado da cadeia i no tempo t

- (d) Se $t > N_b$, incremente o índice de amostras $n \leftarrow n + 1$ e defina a amostra $(C_n, J_n) \leftarrow (C_t^1, J_t^1)$
-

tanto o mapa e a trajetória obtidos a partir dos parâmetros (C, J) daquele estado quanto as estruturas auxiliares calculadas $\mathbf{p}$, $\mathbf{d}$, e $\mathbf{e}$ podem ser armazenadas, tal que propor novos estados requer resolver o problema geométrico do SLAM apenas uma vez, considerando-se apenas as classes de objetos e a solução de associação dos dados propostos.

Outra questão sobre eficiência envolvendo a solução do problema geométrico do SLAM é seu tempo de processamento. Mesmo utilizando uma abordagem muito eficiente para o problema, a solução poderia demorar, uma vez que muitas iterações podem ser necessárias para a otimização não-linear convergir, e o cálculo das equações lineares podem requerer modelos de processamento muito complexo. Além disso, uma vez que o custo de resolver o problema geométrico de SLAM aumenta com o número medidas utilizadas, o uso de dados consistindo de muitas detecções de objetos torna mais lento o processo de amostragem.

Neste trabalho, algumas ações são propostas para reduzir o tempo de processamento necessário para resolver o problema geométrico de SLAM para cada transição de estado. A primeira delas é a tentativa de eliminar as medidas espúrias dos dados usados para calcular a evidência dos estados. Para tal, um algoritmo muito restritivo de associação dos dados é executado sobre os dados brutos, associando medidas similares em imagens consecutivas, o que produz pequenos conjuntos de medidas. Então, apenas as medidas pertencentes aos conjuntos com um número mínimo de elementos são mantidos; as outras medidas são então descartadas. A lógica por detrás desta escolha é que as medidas geradas pelas detecções dos objetos reais atual devem ser mais consistentes em imagens consecutivas. Quanto mais medidas estiverem presentes nos dados brutos, maior podem ser o número mínimo de medidas por conjunto, tornando o processo mais selectivo contra medidas espúrias.

Os conjuntos de medidas gerados pelo passo preliminar da associação de dados são também utilizados como conjuntos atômicos, os quais não podem ser divididos no processo probabilístico de associação de dados. Esses conjuntos atômicos representam rastros dos objetos, uma vez que as medidas contidas são rastreadas em imagens consecutivas, e trabalham da mesma forma que o rastro de características utilizado por Kaess & Dellaert (2005). Considerando que resolver o problema da associação dos dados consiste em encontrar subconjuntos de rastros de objetos, os quais estão em muito menor quantidade do que as medidas individuais, o pré-agrupamento delas em rastros tem a vantagem de reduzir significativamente a dimensão do estados,

aumentando a probabilidade de se propor bons estados e assim melhorar a mixagem e convergência resultantes.

A outra ação para reduzir o tempo necessário para resolver o problema geométrico de SLAM consiste em evitar as múltiplas iterações do processo de otimização não-linear. Após o período de *burn-in*, a cadeia é considerada como pertencendo a um estado de convergência, o qual, no problema aqui abordado, pertence a uma restrita vizinhança de estados. Assim, as soluções de associação de dados pertencentes a esta vizinhança são similares entre si, e gerarão mapas cujos valores de parâmetros comuns mudam muito pouco quando comparados uns com os outros. Afim de tirar vantagem desta funcionalidade, os pontos de linearização da variável são apenas copiados do estado atual da solução de SLAM para o proposto, de modo que o problema geométrico de SLAM é aproximadamente resolvido pela aplicação do passo de otimização linear, descrito na Seção 4.1 apenas uma vez.

No entanto, quando um movimento de fusão ou divisão é proposto, o número de objetos no mapa muda, e nem todos os objetos resultantes da solução atual da associação de dados correspondem aos objetos resultantes da solução proposta para a associação de dados. Para lidar com o número diferente de objetos, assumimos que um movimento de divisão ou fusão entre dois estados não negligenciáveis ou fundem objetos próximos e similares, ou dividem um objeto em dois similares. Este pressuposto é também utilizado pela função de proposição descrita na Seção 5.2.1. Assim, quando o movimento de fusão é proposto, o ponto de linearização do novo objeto assume o valor do ponto de linearização do objeto originário do mapa atual visto primeiro. O ponto de linearização um objeto refere-se aos valores de linearização de suas variáveis geométricas, especificamente a localização l_j e o tamanho g_j . Os objetos originários são aqueles cujos conjuntos de medidas foram unidos para formar o novo objeto e o objeto originário visto primeiro é o que gerou a medida com menor índice de tempo i dentre as medidas que se referem ao objeto fundido no mapa proposto. Por outro lado, se um movimento de divisão é proposto, os pontos de linearização de ambos os novos objetos serão iguais aos pontos de linearização do objeto originário.

Como consequência, apenas uma pequena parte do sistema de equações lineares usadas para resolver o problema geométrico de SLAM precisam ser recalculadas a cada movimento, aumentando a eficiência do processo de amostragem. Na verdade, quando um movimento de divisão é proposto, nenhum coeficiente de equação

precisa ser recalculado: as equações referentes às medições dos objetos originários mantêm os mesmos coeficientes, os quais devem ser apenas deslocados na matriz A do sistema. Quando um movimento de fusão é proposto, no entanto, as equações referentes ao objeto que foi eliminado do mapa, ou seja, aquele não utilizado para gerar o ponto de linearização do novo objeto, devem ser recalculadas.

Porque agora apenas uma parte do conjunto de equações devem ser recalculado quando uma nova solução de associação de dados solução é proposta, um método para atualizar incrementalmente o fator de Cholesky R do sistema linear foi investigada, mas com resultados desanimadores. Da mesma forma que o trabalho de Khan, Balch & Dellaert (2006) sempre que um movimento foi proposto, rotações de Givens foram usadas para atualizar o fator Cholesky resultante do acréscimo de novas equações no sistema, e rotações hiperbólicas foram usadas para atualizar R quando da eliminação de antigas equações do sistema (GOLUB; LOAN, 1996). Mesmo que a aplicação de rotações hiperbólicas para atualização do fator de Cholesky seja numericamente instável, o pior problema enfrentado quando se usa a atualização de Cholesky foi a *diminuição do desempenho*. Embora a atualização de Cholesky de uma simples equação possua complexidade temporal de $O(n)$ no número de variáveis, estes procedimentos tornam-se menos eficiente do que a realização da fatorização QR completa em matrizes esparsas usando reflexões de Householder e reordenando as colunas quando o número de equações a serem atualizadas vão aumentado. No caso deste trabalho, devido a detecção do mesmo objeto diversas vezes em diferentes imagens, cada movimento proposto requer muitas equações a serem atualizadas no fator de Cholesky, causando a degradação do desempenho. No problema abordado por Khan, Balch & Dellaert (2005), apenas poucas equações são atualizadas de cada vez, mantendo o processo eficiente.

5.3.3 Guiando a convergência da cadeia

Ao explorar o espaço de funções de correspondência, as cadeias Markov resultantes da aplicação do algoritmo de MCMC têm lenta convergência e mixagem. Uma vez que a convergência da cadeia requer seu estado estar entre aqueles com probabilidade não desprezível, o período de *burn-in* corresponde ao tempo estimado para a cadeia explorar o espaço do estado, guiada pela distribuição de propostas, e alcançar um estado de "convergência". Este processo pode ser muito lento em distribuições com picos esparsos, mesmo quando boas distribuições de propostas são utilizadas,

devido ao caráter exploratório dos métodos MCMC.

Enquanto determinar um estado inicial não é uma questão importante em aplicações MCMC em geral (GILKS; RICHARDSON; SPIEGELHALTER, 1996a), casos difíceis que resultem em convergência lenta e em cadeias com mixagem ruim podem se beneficiar da condução do estado inicial para um estado "convergado", ou seja, um estado S pertencente ao suporte de $P(S)$, região onde os estados têm probabilidade não desprezível. Esse processo pode economizar muitas transições de estado, o que é especialmente válido neste trabalho, em que transições de estado são computacionalmente caras e há poucas pistas sobre onde os estados altamente prováveis estão.

Neste trabalho, o conhecimento sobre o problema pode ser utilizado para fazer a cadeia convergir mais rápido. Uma vez que existem dois movimentos básicos que exploram o espaço de soluções de associação de dados, um estado altamente provável pode ser encontrado indo de um estado extremo para um outro: quer começando com todas as divisões de rastros de objetos e tentando fundi-las, quer começando com todos os rastros fundidos e tentando dividi-las. Aqui, a primeira opção é usada para acelerar a convergência da cadeia, pois a função da proposta propõe bons movimentos de fusão com maior probabilidade do que bons movimentos de divisão. Isso significa que movimentos de fusão são mais prováveis de aumentar a probabilidade do estado que movimentos de divisão. A razão para isso é a falta de um critério para dividir um determinado objeto em dois conjuntos de rastros, enquanto a distância entre dois objetos é um critério claro para fundi-los.

A estratégia para guiar a convergência da cadeia é implementada usando a técnica de otimização *hill climbing*. Em primeiro lugar, todos os rastros de objetos são considerados como gerando um objeto cada. Então, a função de proposição descrita na Seção 5.2 é utilizada para propor movimentos no espaço do estado, mas propõe fundir movimentos com muito mais probabilidade. Em seguida, o movimento é aceito se e somente se ele leva a um estado de maior probabilidade. Depois de certo número de etapas, o procedimento de amostragem convencional MCMC é iniciado. No entanto, uma vez que a otimização gulosa por si só não assegura a condução de um estado de alta probabilidade, realizar um pequeno período de *burn-in* em MCMC ainda é uma boa prática, embora menos necessária neste caso. A estratégia global para melhorar a eficiência e eficácia do processo de amostragem é descrito no Algoritmo 9.

Algorithm 9 Estratégia proposta para melhorar o processo de amostragem

1. Encontrar uma solução fundamental de associação de dados J_0 , associando medições semelhantes da mesma classe de objeto em imagens consecutivas, agrupando-as em rastros de objetos.
 2. Eliminar do conjunto de dados as medidas pertencentes aos rastros com menos de um número mínimo de medidas associadas
 3. Utilizando o estado (C_0, J_0) composto pela solução fundamental de associação de dados J_0 e de classes de objetos C_0 determinados a partir de classes aparentes, encontrar o mapa e os parâmetros geométricos da trajetória utilizando otimização não-linear.
 4. Considerando rastros de objetos produzidos pela função de correspondência J_0 como indivisíveis, executar otimização *hill climbing* do espaço do estado (C, J) , propondo novos estados de acordo com o procedimento descrito nas Seções 5.2 and 5.3.2, enfatizando a proposta de movimentos de fusão. Realizar otimização não-linear do mapa geométrico e parâmetros de trajetória após um certo número de passos de *hill climbing*.
 5. Iniciar a amostragem com o algoritmo MC³, propondo novos estados de acordo com o procedimento descrito nas Seções 5.2 and 5.3.2, propondo movimentos de fusão ou divisão com a mesma probabilidade. Considerar os primeiros 10% das amostras como *burn-in*, e descartá-las.
-

6 Resultados Experimentais de O-SLAM

Este capítulo apresenta e discute os resultados experimentais da abordagem proposta para a construção de O-Maps. Eles foram construídos utilizando-se diferentes conjuntos de dados, e três diferentes níveis de conhecimento prévio sobre os objetos foram assumidos, o que corresponde à configuração descrita na Seção 3.2.1.

A abordagem apresentada para O-SLAM foi testada utilizando-se imagens panorâmicas reais capturadas por robôs que carregavam câmeras onidirecionais. A abordagem proposta claramente se beneficia de utilizar as imagens panorâmicas, uma vez que várias visões de um objeto são tomadas sem a preocupação de os focar ativamente na imagem.

6.1 Resultados experimentais de SLAM geométrico

O primeiro conjunto de experimentos é destinado a avaliar o desempenho do O-SLAM quando classes de objetos e identidades são conhecidas. Essa situação corresponde à configuração 1, quando os objetos são perfeitamente detectados e identificados. Embora seja raro o caso na prática, SLAM baseado em objetos assume a configuração 1, que corresponde ao problema de SLAM geométrico, e é a base para resolver todos os outros casos. Dois diferentes conjuntos de imagens captadas por robôs que carregam câmeras onidirecionais foram utilizados para testar esta abordagem.

6.1.1 Conjunto de dados Cogniron

O primeiro conjunto compreende imagens de um conjunto de dados anotados do mundo real, obtidos on-line do site do projeto Cogniron (ZIVKOVIC et al., 2008). A

Figura 6.1: Exemplo de imagem anotada do conjunto de dados Cogniron. Os objetos são marcados com polígonos delimitadores, e receberam rótulos manualmente.

sequência de imagens foi captada por uma câmera onidirecional usando um espelho hiperbólico. A Figura 6.1 mostra um exemplo da imagem anotada do conjunto de dados utilizado. Os objetos nas imagens foram manualmente segmentados utilizando polígonos delimitadores, e tiveram suas classes listadas em um arquivo auxiliar.

Nos experimentos, os objetos da cena são descritos por cilindros, tendo a sua localização representada pela posição 3D do ponto de base, conforme ilustrado na Figura 6.2a. A fim de obter medidas compatíveis com a representação dos objetos adotada, os dados medidos Z são obtidos através da utilização da fatia delimitadora dos polígonos, uma vez que fatias delimitadoras são a projeção simplificada de cilindros em imagens panorâmicas. Figura 6.2b ilustra como as medidas foram tomadas a partir do conjunto de dados anotados.

Os dados adquiridos e odômetros foram utilizados para construir mapas baseados em objetos e recuperar a trajetória robô utilizando a abordagem proposta para SLAM geométrico. Os modelos de movimento e observação, definidos pelas funções $f(x, y)$, $h^u(x, l)$, e $h^s(x, l, g)$ são detalhados no Apêndice A. A implementação da versão geométrica do algoritmo O-SLAM requer também a definição dos parâmetros de incerteza. Ver apêndice A para verificar como as matrizes de covariância Q_i , R_{ik} e W_{ik} são calculadas a partir dos parâmetros de incerteza.

Figura 6.2: Ilustração dos dados medidos e do modelo de objeto usado nos experimentos de SLAM geométrico. (a) Modelo cilíndrico adotado para representar os objetos, descritos por sua altura, largura, e posição do ponto de base. (b) Posições de objetos detectados e tamanhos nas imagens. O tamanho é descrito por uma fatia delimitadora do objeto, extraída do polígono anotado, que consiste no tamanho radial e na largura angular. Os pequenos círculos indicam a projeção dos pontos de base do objeto, usados para descrever suas posições aparentes nas imagens.

Classe	Tamanho a priori	
	Média (diâmetro, altura)	Desvio padrão (diâmetro, altura)
Couch	(1.8m,1.2m)	(1m,1m)
TV	(0.5m,0.5m)	(1m,1m)
Armchair	(0.8m,1.2m)	(1m,1m)
Computer desk	(1.0m,1.5m)	(1m,1m)
Cabinet	(1.0m,1.7m)	(1m,1m)
Livingroom dining table	(1.8m,0.8m)	(1m,1m)
Book case	(1.5m,2.3m)	(1m,1m)
Coat rack	(0.6m,2.0m)	(1m,1m)
Wardrobe	(0.8m,2.2m)	(1m,1m)
Kitchen dining table	(1.8m,0.7m)	(1m,1m)
Bed	(2.0m,0.7m)	(1m,1m)
Kitchen work desk	(2.0m,1.2m)	(1m,1m)
Washing machine	(0.6m,0.8m)	(1m,1m)
Refrigerator	(0.8m,2.0m)	(1m,1m)

Tabela 6.1: Classes detectadas no conjunto de imagens Cogniron e a crença prévia assumida sobre os tamanhos de suas instâncias.

As classes de objetos detectados nas imagens do conjuntos de dados e a crença prévia nos parâmetros de tamanho são descritas na Tabela 6.1. A crença prévia sobre os tamanhos dos objetos corresponde a apenas um palpite baseado nas imagens disponíveis. Para refletir esta incerteza, a variância dos parâmetros prévios é definida por um valor grande.

Os resultados obtidos são mostrados na Figuras 6.3 e 6.4. Como base de comparação, a trajetória obtida com um algoritmo de SLAM tradicional usando varreduras de laser, disponíveis no conjunto de dados, foi usada como trajetória de referência. Os resultados da resolução do problema de SLAM geométrico com a abordagem proposta foram apresentados em um artigo (SELVATICI; COSTA; DELLAERT, 2008), porém sem executar otimização não-linear.

Os resultados mostram que a abordagem proposta efetivamente produz mapas que aproximam objetos reais no ambiente. A Figura 6.4a mostra que o algoritmo proposto melhora a trajetória recuperada com respeito à trajetória obtida a priori por odometria, enquanto a Figura 6.4b mostra que as projeções do objeto mapeado sobre as imagens estão próximos aos correspondentes objetos aparentes reais na maioria dos casos.

Todavia, o modelo de objeto adotado não aproximou muito bem alguns dos objetos mapeados. Existem classes cujas instâncias apresentaram formas achatadas, que

Figura 6.3: Resultados de O-SLAM geométrico obtidos com o conjunto de dados Cognirion. (a) Visão perspectiva do O-Map obtido em conjunto com a trajetória inferida (b) Visão superior do modelo de objetos construído. A linha pontilhada preta representa a trajetória inferida, enquanto que a linha pontilhada verde-escura é a trajetória obtida por SLAM baseado em laser, que consideramos como nossa referência. A planta da casa foi extraída do artigo de Zivkovic, Booij & Kröse (2007).

(a)

(b)

Figura 6.4: Mais resultados obtidos com o conjunto de dados Cogniron, ilustrando o desempenho da abordagem proposta para SLAM geométrico. (a) Comparação entre as trajetórias. A trajetória inferida está mais próxima da trajetória de referência. (b) Exemplo de projeção de objetos do mapa no plano da imagem.

não são bem modeladas pelos cilindros. Nestes casos, a largura do objeto mapeado corresponde a uma "largura intermediária", obtida pela obtenção da média do comprimento do objeto e sua profundidade, de acordo com os pontos de vista dos quais o robô tenha detectado. Além disso, como uma consequência da fragilidade do modelo, a correção da trajetória pelo mecanismo de inferência não foi boa o suficiente para manter a sua qualidade quando o robô se move para longe da posição de partida, causando também a degradação das posições do objeto.

6.1.2 Conjunto de dados LPA

O algoritmo proposto para resolver o problema de SLAM geométrico foi também testado utilizando imagens processadas por uma técnica detectora de objeto simples, amplamente conhecida na comunidade de robótica. Neste experimento, objetos simples e coloridos foram colocados em torno do meio ambiente, e um robô Pioneer 3-DX (ver Figura 6.5), carregando um sistema de câmera onidirecional, captou imagens panorâmicas enquanto explorava o ambiente. Então, CMVision (BRUCE; BALCH; VELOSO, 2000) foi utilizado para encontrar blobs de cor em imagens panorâmicas, correspondentes a detecções de objetos. Dados de varreduras de laser, odometria e sonares também foram gravados. O nome do conjunto de imagens — conjunto de dados LPA — é devido ao fato de que a aquisição dos dados ocorreu no Laboratório de Percepção Avançada (LPA), na Universidade de São Paulo.

O movimento do robô durante a aquisição de dados foi conduzido manualmente, usando um controle remoto sem fio. Os dados foram adquiridos todos de uma vez em etapas discretas: após o robô ter sido ordenado para se mover ou virar por uma pequena distância, ele parou e então registrou a saída dos sensores. Comandos de translação foram impostos ao robô para se mover para a frente ou para trás 20 centímetros, enquanto comandos de rotação orientaram o robô para virar 10 ° no sentido horário ou no sentido anti-horário.

CMVision detecta blobs de cor em imagens encontrando regiões conectadas cuja cor pertence a um determinado intervalo do espaço YUV. Nos experimentos, os intervalos foram determinados pela escolha de uma região de interesse (ROI) de imagens de objetos a serem identificados, e calculando a média e o desvio-padrão para cada canal. Então, a média da cor é definida como centro de intervalo, enquanto lados delimitadores são definidos considerando três vezes o desvio padrão.

Figura 6.5: Fotos do robô e de um objeto presente no conjunto de imagens do LPA. (a) O robô Pioneer 3-DX carrega um sistema de imagem onidirecional, constituída por uma câmera de vídeo acoplada a um espelho hiperbólico. (b) Exemplo de um objeto "green box" detectado por CMVision nos experimentos. Cada classe de objeto é representada por uma cor diferente.

Uma vez que um algoritmo real de visão computacional foi utilizado para detectar objetos em imagens, identidades de objetos tiveram de ser definidos manualmente, e os erros de detecção de objetos tiveram de ser manualmente consertados. Para tal, um conjunto pequeno e igualmente espaçado de imagens captadas foi utilizado para se obter medidas, descartando-se os demais, e detecções corretas de objetos foram selecionadas manualmente nestas imagens. Tal como no experimento anterior, os objetos da cena são descritos por cilindros, como mostrado na Figura 6.2a, e as medidas geométricas são descritas pelos parâmetros de fatias delimitadoras de objetos aparentes. A Figura 6.6 mostra um exemplo de imagem e os objetos detectados.

As classes dos objetos detectados nas imagens do conjunto de dados e a crença prévia nos seus parâmetros de tamanhos são descritas na Tabela 6.2. Já que os objetos foram realmente vistos, uma boa estimativa visual de tamanho pode ser feita, refletindo as variâncias dos parâmetros de tamanho. Os tamanhos corretos também são informados como base de comparação.

Os resultados obtidos são mostrados nas Figuras 6.7 e 6.8. Como base de comparação, a trajetória obtida com o algoritmo DP-SLAM (ELIAZAR; PARR, 2004) usando

Figura 6.6: Posições de objetos detectados e tamanhos nas imagens pertencentes ao conjunto de dados LPA. Observa-se que o tamanho da "orange door" é detectado imprecisamente.

	Tamanhos prévios		Tamanho correto (comprimento, largura, altura)
	Média (diâmetro, altura)	Desvio padrão (diâmetro, altura)	
green box	(0.6m,0.6m)	(0.3m,0.15m)	(0.49m,0.47m,0.57m)
yellow box	(0.3m,0.3m)	(0.2m,0.15m)	(0.23m,0.14m,0.23m)
pink ball	(0.25m,0.25m)	(0.15m,0.15m)	(0.20m,0.20m,0.20m)
blue ball	(0.25m,0.25m)	(0.15m,0.15m)	(0.20m,0.20m,0.20m)
orange ball	(0.25m,0.25m)	(0.15m,0.15m)	(0.20m,0.20m,0.20m)
Magellan	(0.5m,0.3m)	(0.2m,0.2m)	(0.40m,0.40m,0.25m)
orange door	(1.0m,2.2m)	(0.3m,0.5m)	(1.04m,0.00m,2.17m)

Tabela 6.2: Classes detectadas no conjunto de imagens LPA e crença prévia assumida dos tamanhos de suas instância.

	Tamanho inferido (diâmetro, altura)	Tamanho correto (diâmetro, altura)
green box	(0.64m,0.66m)	(0.68m,0.57m)
yellow box	(0.24m,0.25m)	(0.26,0.23m)
pink ball	(0.24m,0.23m)	(0.20m,0.20m)
blue ball	(0.23m,0.23m)	(0.20m,0.20m)
orange ball	(0.20m,0.19m)	(0.20m,0.20m)
Magellan	(0.43m,0.23m)	(0.40m,0.25m)
orange door	(0.98m,2.12m)	(1.04m,2.17m)

Tabela 6.3: Comparação entre os tamanhos dos objetos inferidos com o algoritmo SLAM geométrico proposto e os tamanhos corretos. Cada objeto pertence a uma classe diferente. Os diâmetros corretos de objetos hexaédricos — volume com seis faces — especificamente, yellow box, green box e orange door, foram calculados utilizando-se a norma euclidiana dos valores correspondentes de comprimento e largura na Tabela 6.2.

varreduras de laser foi utilizada como trajetória de referência. Tabela 6.3 compara os tamanhos de objeto obtidos com a abordagem apresentada por SLAM geométrico e os tamanhos corretos, medidos com uma trena.

Como no experimento da Seção 6.1.1, a abordagem proposta para resolver o problema SLAM geométrico foi efetiva em gerar um mapa que aproxima a posição e tamanhos dos objetos no ambiente, e em reduzir o erro de localização. Neste experimento, no entanto, os tamanhos dos objetos foram reproduzidos com maior fidelidade, uma vez que os objetos utilizados foram melhores aproximados pelo modelo cilíndrico do que os utilizados no conjunto de dados Cogniron. Isto pode ser verificado na Figura 6.8b: enquanto as projeções de objetos em imagens não estavam no local correto algumas vezes, seus tamanhos combinaram com os objetos na imagem muito bem.

Uma exceção notável é o objeto "orange door". Uma vez que é um objeto plano, ele está bem modelado por um cilindro, se visto pela câmera somente do ponto de vista frontal. Se o objeto é visto de um ponto de vista enviesado, a sua largura aparente torna-se menor, sugerindo ao modelo assumido que está mais distante do que ele realmente está para o robô. Ela explica porque orange door está distante da parede na Figura 6.7b, enquanto que a posição de referência deve coincidir com a parede. A trajetória do robô também é afetada pelo deslocamento da porta, também indo de encontro a ela.

Embora o mapa e a trajetória obtidos utilizando a abordagem proposta para resolver o problema de SLAM geométrico não geram mapas tão precisos quanto as soluções baseadas em sensores mais precisos, ele é bem sucedido em recuperar a localização do robô no quadro global. A Figura 6.7 mostra que as posições do objeto

(a)

(b)

Figura 6.7: Resultados de SLAM geométrico obtidos com o conjunto de dados LPA ajustados manualmente. (a) Visão perspectiva do O-Map obtido em conjunto com a trajetória inferida. As cores do objeto são a média delas utilizadas para detectá-las em imagens. (b) Visão superior do modelo de objetos construídos. A linha preta pontilhada representa a trajetória inferida, enquanto a linha verde é a trajetória obtida por SLAM baseado em laser, considerada como referência. O mapa de grade de ocupação foi gerado utilizando-se DP-SLAM. Note que green box sobrepõe a maioria dos objetos detectados por laser.

(a)

(b)

Figura 6.8: Mais resultados obtidos com o conjunto de dados LPA, ilustrando o desempenho do modelo geométrico de O-SLAM. (a) A comparação entre as trajetórias. A inferida trajetória está mais próxima da referência. (b) Exemplo de projecção de objectos do mapa no plano da imagem.

obtidas e a trajetória do robô diferem significativamente da referência, principalmente por causa da incerteza do modelo e das medidas. Por exemplo, a Figura 6.6 mostra um exemplo de medição imprecisa do tamanho de um objeto. Então grande imprecisão não ocorre em detectores de linha ou ponto em imagens, tais como descritores SIFT (LOWE, 1999). Mesmo assim, a abordagem proposta melhora a localização robô quando comparada com odômetros, mostrando que, uma vez que a associação de dados corretos é fornecida, a abordagem proposta é capaz de recuperar a localização do robô aproximada mesmo se os odômetros indicarem uma localização inconsistente.

6.2 Resultados sobre associação de dados probabilística

Neste conjunto de experimentos, O-Maps foram obtidos considerando o caso de reconhecimento perfeito do objeto. O conjunto de dados LPA também foi utilizado para testar o algoritmo O-SLAM nos casos de identidade de objetos desconhecidas. Neste caso, em vez de selecionar manualmente boas medidas, como na Seção 6.1.2, as espúrias foram retiradas automaticamente utilizando o método proposto na Seção 5.3.2. No entanto, detecções de objetos pertencentes às classes que são susceptíveis de serem confundidas com outras foram removidas do conjunto de dados, uma vez que detectores de objetos são assumidos para detectar apenas objetos pertencentes a uma classe específica. Este é o caso do objeto "orange ball", que foi detectado no lugar de "orange door" com frequência.

Para a construção do mapa, a função propositora descrita na Seção 5.2.1 foi utilizada juntamente com o algoritmo MC³ e a estratégia apresentada na Seção 5.3 para amostrar ao longo do espaço das funções de correspondência J . O MC³ foi executado com três cadeias, com um incremento de temperatura $\Delta T = 3.33$ entre eles. Todas as soluções de associação de dados possíveis J , isto é, aqueles que não fundem medidas relatando diferentes classes de objetos ou que não dividem rastros de objetos, são consideradas como tendo a mesma probabilidade prévia, de forma que $P(J) \propto 1$.

Os parâmetros da distribuição de propostas são a probabilidade de propor um movimento de fusão $P_{merge} = 0.5$, implicando em $P_{split} = 0.5$, bem como o limiar de distância para objetos próximos, usados na etapa 2 do Algoritmo 5, sendo 0.5m. Uma vez que as classes de objeto são determinadas diretamente a partir dos detectados, não é feita exploração desse espaço. Portanto, a probabilidade de propor movimentos de classe

trocados é definido como zero ($P_{switch} = 0$), e objetos pertencentes a diferentes classes nunca são fundidas. Isto é feito através da definição do fator de penalidade para e_j , usado na etapa 4 do algoritmo 5 em infinito.

Conforme explicado na Seção 4.4.1, densidades prévias Gaussianas devem ser definidas para todos os parâmetros geométricos desconhecidos. Enquanto a média prévia para todas as posições de objetos são assumidas $\gamma(c_j) = (0, 0, 0)^T$, cada classe de objeto lança uma crença prévia diferente sobre os parâmetros de tamanho. Os mesmos parâmetros de crença prévios listados na Tabela 6.2 foram utilizados neste experimento, exceto para a classe "orange ball", cujas detecções de objetos foram ignoradas a partir do conjunto de dados.

A fim de filtrar as detecções espúrias, medições semelhantes em imagens consecutivas foram previamente associadas entre si em conjuntos de detecção de objetos atômicos. Duas medidas em imagens consecutivas $z_{i,k}$ e $z_{i+1,k}$ eram fundidas se

$$\left\| \begin{bmatrix} u_{i+1,k}^r - u_{i,k}^r \\ u_{i+1,k}^\phi - u_{i,k}^\phi \\ s_{i+1,k}^r - s_{i,k}^r \\ s_{i+1,k}^\phi - s_{i,k}^\phi \end{bmatrix} \right\| < \begin{bmatrix} 30 \text{ pixels} \\ 20^\circ \\ 0.2 \min(s_{i+1,k}^r, s_{i,k}^r) \\ 0.2 \min(s_{i+1,k}^\phi, s_{i,k}^\phi) \end{bmatrix}, \quad (6.1)$$

onde $u_{i,k}^r$ e $s_{i,k}^r$ são, respectivamente, a posição e tamanho radiais medidos do objeto na imagem captada no instante i , e $u_{i,k}^\phi$ e $s_{i,k}^\phi$ são, respectivamente, a posição e largura angular medidos. Rastros foram eliminados se fossem compostos por menos de 2 detecções. A configuração descrita foi utilizada para processar as mesmas imagens do experimento descrito na Seção 6.1.2, rejeitando as medidas detectadas como "orange ball". Foram executados 100 passos do algoritmo de otimização *hill climbing* da cadeia, aplicando a otimização não-linear dos parâmetros geométricos a cada 50 passos, e considerando $P_{merge} = 0.9$ e $P_{split} = 0.1$ nesta fase. Então, o algoritmo MC³ foi utilizado para extrair 667 amostras, rejeitando as primeiras 70 amostras como *burn-in*. O mapa obtido é mostrado na Figura 6.9, onde objetos com probabilidade de existência $\pi < 0.1$ não foram mostrados.

Os resultados mostram que, apesar da presença de muitos objetos espúrios no mapa, a trajetória do robô e os parâmetros geométricos dos objetos corretos não se alteraram significativamente em comparação com o caso em que a solução de referência de associação de dados foi utilizada. Os objetos marcados como espúrios com uma elipse na Figura 6.9 são devidos a medidas espúrias que não foram removidas

Figura 6.9: Resultados de O-SLAM obtidos com o conjunto de dados LPA e classes de objetos conhecidas, embora com solução de associação de dados desconhecida. (a) visão perspectiva do O-Map obtido em conjunto com a trajetória inferida. As cores do objeto são a média daquelas utilizadas para detectá-las em imagens. Objetos espúrios estão circulados.

(b) Visão superior do O-Map construído. A linha pontilhada preta representa a trajetória inferida, enquanto a linha verde é a trajetória obtida pela DP-SLAM. (c) Detalhe da região em torno do “green box” não-espúrio na Figura 6.9b, mostrando três objetos daquela classe (dois deles são perfeitamente sobrepostos). Os triângulos vermelhos indicam o centro. A transparência indicaria a probabilidade de existência do objeto, mas dois objetos são tão transparentes que não podem sequer serem vistos.

com sucesso dos dados brutos com o processo de filtração proposto. Todos os objetos não-espúrios foram gerados de forma correta.

A Figura 6.9c mostra como o mapa resultante expressa as múltiplas hipóteses sobre a solução de associação de dados envolvendo medidas de "green box" não-espúrias. Embora apenas um cilindro possa ser visto na figura, existem três diferentes "green boxes" representados lá. O visível, gerado pela solução mais provável de associação de dados, tem alta probabilidade de existência. Os outros dois, representados com alta transparência no mapa, foram gerados assumindo a solução de associação de dados de menor probabilidade. Apesar de todas as posições de objetos sejam indicadas com um triângulo vermelho, duas delas são tão perfeitamente sobrepostas que parecem uma só na figura. Intuitivamente, eles representam o mesmo objeto físico e podem ser combinados no mapa, apesar de terem *ID* diferentes. Isso sugere que uma melhor definição da identidade do objeto em diferentes soluções para a associação de dados é adequada para o tratamento de casos em que um objeto físico aparece como diversas probabilidades baixas no mapa.

Detecções de objeto espúrios correspondem a um modo de erro frequente de detectores de objeto baseados em CMVision utilizados no experimento. A Seção 4.3.2 propõe um modelo probabilístico para descrever se os objetos mapeados são prováveis de terem sido gerados por medidas espúrias ou por aquelas válidas, incorporando esta crença na probabilidade de existência do objeto. Figura 6.10 mostra o resultado da aplicação deste modelo para o cálculo do mapa final, assumindo os parâmetros da função de probabilidade de espuriedade sigmoideal como $t_0 = (0.4m)^2$ e $\tau = (0.2m)^2$. Resultados semelhantes foram submetidos para uma conferência (SELVATICI; COSTA; DELLAERT, 2009), embora tenha sido usado o algoritmo MH, e os objetos foram classificados estritamente entre espúrio e não-espúrio, que é equivalente a usar $\tau = 0$.

Os resultados apresentados na Figura 6.10 indicam que o modelo assumido para detectar objetos espúrios foi eficaz, apesar de um objeto não esperado, identificado como "green box", ainda ter recebido alta probabilidade de existência. Figura 6.11 aponta uma razão plausível para isso: as medidas referentes ao objeto indesejável foram sistematicamente tomadas a partir de um objeto real no ambiente, especificamente a parte da perna de uma pessoa, mesmo que esse objeto não se destinava a ser mapeado.

Finalmente, para analisar o efeito de melhorar a mixagem de cadeia do amostrador usando o algoritmo MC³, o mesmo conjunto de dados e parâmetros do algoritmo foram

(a)

(b)

Figura 6.10: Vista superior do O-Map obtido considerando o modelo proposto de espuriedade do objeto. (a) Mapa completo e trajetória, em que a transparência indica a probabilidade de existência do objeto. (b) A parcela dos objetos com $\pi_{ID} > 0.5$.

Figura 6.11: Exemplo de projeção dos objetos no mapa da Figura 6.10b sobre o plano da imagem. Observa-se que a perna da pessoa na imagem é detectada como "green box".

usados para amostrar sobre o espaço de associação de dados usando o algoritmo MH. No entanto, para uma comparação mais justa, foram extraídas três vezes mais amostras, uma vez que foram utilizadas três cadeias correndo em paralelo pelo algoritmo MC³. Os histogramas da amostra estão apresentados na Figura 6.12, comparando-os com as respectivas parcelas de evidência normalizadas, que é igual, no caso ideal, ao histograma de amostras normalizado. De fato, o algoritmo MC³ melhorou a mixagem do amostrador, visitando cada estado de acordo com a meta posterior, enquanto que o algoritmo MH visitou estados com probabilidade desprezível com mais frequência.

6.3 Resultados de O-SLAM assumindo confusão entre classes

O mesmo conjunto de imagens LPA foi usado para testar a abordagem proposta para tratar detecção imperfeita de objetos, considerando-se o caso em que objetos de uma determinada classe podem ser detectados como objetos pertencentes a outra classe. Para mostrar o impacto em modelar este modo de erro no mapa, o algoritmo O-SLAM foi executado duas vezes: uma vez assumindo classes de objeto perfeitas, e outra vez, permitindo a confusão entre as classes. Neste experimento, uma melhor sintonia do detector de blob CMVision foi realizada, e foram consideradas detecções

Figura 6.12: Comparação entre histogramas de amostras obtidos pelos algoritmos MC³ e MH. (a) Histogramas obtidos com MC³. (b) Histogramas obtidos utilizando o algoritmo MH processando os mesmos dados.

de objetos de todas as classes, incluindo aqueles relatando “orange ball”. Para eliminar mais detecções espúrias a partir dos dados, apenas faixas de objetos com pelo menos três medidas foram consideradas, removendo as medidas não pertencentes a essas faixas. A probabilidade de confusão, $P(a_{i,k_i}|c_{J(i,k_i)})$, é dada pela distância Mahalanobis entre ambas as classes no espaço de cores. Os mesmos parâmetros dos experimentos anteriores foram utilizados. No caso de considerarmos classes imperfeitas, movimentos de classe trocados são também sugeridos pela função de proposta, e as probabilidades de proposta de movimentos adotadas foram $P_{merge} = 0.4$, $P_{split} = 0.4$ e $P_{switch} = 0.2$. Os mapas de objeto resultantes estão representados nas figuras 6.13 and 6.14.

Ao utilizar o modelo que trata classificação errônea de objetos, o mapa resultante mostra todos os objetos de alta probabilidade com suas classes corretas, incluindo orange ball. Por outro lado, quando classes de objetos perfeitas são assumidas erroneamente, o objeto laranja é mapeado por duas vezes: uma vez como “orange ball” e outra vez como objeto “orange door”. Isto porque o mesmo físico de orange ball foi algumas vezes detectada pelo detector “orange door”, como mostrado na Figura 6.15. Assim, a abordagem proposta tem a capacidade de consertar classes de objetos erra-

(a)

(b)

Figura 6.13: O-Map e a trajetória obtida utilizando o conjunto de dados LPA, com o pressuposto errado de classes de objetos perfeitas. Os objetos exibidos têm probabilidade de existência $\pi_{ID} > 0.1$. (a) Visão superior do mapa, mostrando também a grade de ocupação obtida com DP-SLAM. (b) Visão perspectiva do mapa, mostrando objetos sobrepostos correspondentes ao mesmo físico ("orange ball") detectou como pertencentes a duas classes distintas ("orange ball" e "orange door").

(a)

(b)

Figura 6.14: O-Map e trajetória obtida utilizando o conjunto de dados LPA e permitindo a confusão entre as classes. Os objetos exibidos têm a probabilidade de existência $\pi_{ID} > 0.1$.
 (a) Visão superior do mapa, mostrando também a grade de ocupação obtida com DP-SLAM.
 (b) Visão perspectiva do mapa. Agora, orange ball está corretamente mapeada.

Figura 6.15: Posições de objeto detectadas e tamanhos em imagens pertencentes ao conjunto de dados LPA. Observa-se que orange ball é detectada por duas vezes: uma vez como pertencente à classe correta, e outra vez como objeto "orange door".

das, inferindo-as conjuntamente com a trajetória do robô e os parâmetros geométricos do mapa.

Nos mapas das Figuras 6.13 e 6.14, um novo objeto espúrio aparentemente foi mapeado com probabilidade de existência alta, identificada como "yellow box" à esquerda da trajetória do robô. Este objeto corresponde, de fato, a uma verdadeira yellow box, que foi detectada usando nova sintonia do CMVision.

O algoritmo O-SLAM também foi testado usando um conjunto de imagens grande, também obtidos em LPA, a fim de testar a sua capacidade para fechar grandes voltas na trajetória. Neste experimento, o robô realizou uma longa volta enquanto detectava objetos em imagens. O algoritmo blob finder foi sintonizado para detectar diferentes classes que estão listadas, juntamente com a crença prévia sobre seus parâmetros de tamanhos, na Tabela 6.4. A Figura 6.16 mostra exemplos de imagens captadas pelo robô, bem como os objetos detectados. As Figuras 6.17 e 6.18 ilustram os resultados obtidos pela aplicação do algoritmo proposto sobre as imagens. Porque um maior conjunto de dados foi utilizado, agora 2.000 amostras foram extraídas utilizando o algoritmo MC³, rejeitando 200 como *burn-in*, e realizando 400 passos de otimização hill climbing para guiar a convergência da cadeia.

	Tamanho a priori	
	Média (diâmetro, altura)	Desvio padrão (diâmetro, altura)
green box	(0.6m,0.6m)	(0.3m,0.15m)
yellow box	(0.3m,0.3m)	(0.2m,0.2m)
pink ball	(0.25m,0.25m)	(0.1m,0.1m)
blue ball	(0.25m,0.25m)	(0.1m,0.1m)
orange ball	(0.25m,0.25m)	(0.1m,0.1m)
red bottle	(0.1m,0.3m)	(0.05m,0.1m)
green bucket	(0.3m,0.4m)	(0.2m,0.2m)
blue bucket	(0.3m,0.4m)	(0.2m,0.2m)
orange bucket	(0.3m,0.4m)	(0.2m,0.2m)

Tabela 6.4: Classes detectadas no conjunto de imagens LPA e a crença prévia assumida sobre os tamanhos de suas instância.

(a)

(b)

Figura 6.16: Posições de objetos detectados e tamanhos em imagens pertencentes ao conjunto grande de dados LPA. (a) Imagem mostrando somente detecções não-espúrias.

(b) Imagem mostrando medidas espúrias realizadas a partir da porta e dos objetos vermelhos na parte esquerda da imagem, que não se destinaram a serem detectadas no presente experimento.

(a)

(b)

Figura 6.17: Resultados de O-SLAM obtidos com conjunto grande de dados LPA, mostrando os objetos com $\pi_D > 0.5$. (a) Visão perspectiva de uma parte do O-Map obtido, mostrando os objectos detectados no interior do laboratório. Os objetos espúrios estão circulados. (b) Visão superior do modelo de objetos construído. A linha pontilhada preta representa a trajetória inferida, enquanto a linha verde é a trajetória obtida por SLAM baseado em laser, considerado como referência. O mapa de grade de ocupação foi gerado utilizando DP-SLAM. Note que nenhum objeto do lado de fora da sala é mostrado.

(a)

(b)

Figura 6.18: Mais resultados obtidos com o conjunto grande de dados LPA, ilustrando o desempenho do algoritmo proposto em estimar os parâmetros geométricos corretos. (a) Comparação entre as trajetórias. A trajetória inferida está mais próxima à referência. Nesta figura, são mostrados todos os objetos mapeados, representados por triângulos vermelhos. (b) Exemplos de projeção de objetos com baixa probabilidade de espuriedade no mapa no plano da imagem. A imagem à esquerda enfatiza projeções dos objetos mapeados corretamente, enquanto a imagem à direita enfatiza as projeções de objetos espúrios registrados com uma baixa probabilidade de espuriedade no mapa.

Neste experimento, o algoritmo proposto demonstrou a capacidade de corrigir um grande erro na trajetória, apesar de alguns objetos serem mapeados com baixa probabilidade de espuriedade. Objetos fora da sala não puderam ser detectados consistentemente devido à pouca luz no corredor, para que todos os objetos colocados ali fossem considerados espúrios, mesmo se destinavam a serem mapeados. Por outro lado, além do mapeamento de objetos corretos no interior do laboratório, alguns indesejados apareceram no mapa com alta probabilidade. Conforme pode ser observado na Figura 6.18bb, esses objetos correspondem a objetos vermelhos, perto da pessoa na imagem, ou eram causados por detecções fragmentadas de orange door, exemplificadas na Figura 6.16b. Estas medidas espúrias foram consistentemente tomadas neste conjunto de dados, violando os princípios assumidos pela abordagem proposta.

Por outro lado, mesmo se nenhum objeto foi mapeado com alta probabilidade existência fora do laboratório, estes (considerados) objetos espúrios ajudaram no processo de mapeamento, servindo como marcos para localização do robô. Assim, o robô não perdeu a trilha da sua postura fora do laboratório, e foi capaz de combinar corretamente detecções de objetos realizadas no início do trajeto com aquelas realizadas no final.

7 Conclusão e Trabalhos Futuros

Esta tese introduz o O-Map, um mapa baseado em objetos para a navegação de robô, bem como o algoritmo O-SLAM, uma técnica de SLAM off-line para construí-lo a partir de dados visuais. O mapa contém modelos geométricos simples dos objetos no ambiente, bem como etiquetas semânticas que os descrevem e probabilidades associadas à existência real dos robôs no ambiente. Esta estrutura de mapa é nova para o contexto da robótica móvel, e além de ser útil para tarefas de navegação, pode servir como uma base para a comunicação entre o robô e usuários humanos, onde os objetos no mapa representam o vocabulário comum. O algoritmo proposto constrói um mapa que aproxima com êxito os objetos do ambiente, conjuntamente inferindo a semântica e as partes geométricas utilizando um modelo probabilístico que casa bem essas partes, enquanto recupera a trajetória do robô precisamente, mesmo quando a solução correta para a associação de dados é desconhecida e os detectores de objetos utilizados geram algumas vezes detecções erradas.

As alegações sobre os O-SLAM feita na Introdução (Seção 1.4) foram demonstradas nesta tese. Primeiro, ficou demonstrado que O-Maps consistentes são obtidos se a correta solução de associação de dados e classes de objetos são dados. Além disso, quando as geometrias do objeto são bem modeladas pelo volume geométrico assumido, seus parâmetros geométricos estimados são boas aproximações dos correspondentes no mundo real. A trajetória do robô aproximada também é recuperada, melhorando a estimativa da localização do robô em relação ao obtido pelo odômetro, confirmando assim a eficácia da abordagem proposta para resolver o problema SLAM geométrico. Na verdade, a trajetória do robô é sempre recuperada, ou pelo menos melhora significativamente quando comparada com as baseadas em odômetros, mesmo que a solução correta para a associação de dados e as classes de objetos são desconhecidos e o robô realiza uma grande volta na trajetória.

Foi também demonstrado que inferir conjuntamente as geometria dos objetos e classes é a forma adequada para construir mapas baseados em objetos quando a

solução para a associação de dados é desconhecida e as classes de objetos são imperfeitamente conhecidas. Para que isso seja possível, este trabalho apresentou um modelo probabilístico que liga firmemente a trajetória do robô, as geometrias do objeto e classes, e a solução para associação de dados, bem como um método de inferência para estimar essas incógnitas. Foi demonstrado por um exemplo experimental que fazer dessa forma pode gerar mapas melhores do que considerar a associação de dados e a classificação de objeto problemas separados, admitindo que os mesmos dados e os modelos são usados em ambos os casos.

Os experimentos realizados neste trabalho mostram também que os métodos MCMC são capazes de produzir com sucesso mapas baseados em objeto corretos amostrando sobre o espaço de soluções para associação de dados e as classes de objetos, melhorando os resultados de trabalhos relacionados (DELLAERT, 2001; KAESE; DELLAERT, 2005; RANGANATHAN, 2008). Nestas obras, MCMC é utilizado para inferência no espaço de soluções de associação de dados (ou topologia), mas aqui uma dimensão adicional é adicionado para o problema. No entanto, em qualquer caso, MCMC deve ser utilizado apenas em problemas de tamanho limitado, porque ele escala muito mal com o número de medições ou faixas para ser associado. Embora este trabalho propõe estratégias para melhorar a eficiência MCMC, alargando também os resultados dos trabalhos relacionados, isso só aumenta o tamanho máximo dos problemas que podem ser resolvidos em tempo razoável, mas não resolve o problema do mau dimensionamento inerente à inferência no espaço das soluções da associação de dados.

7.1 Contribuições

Esta tese contribui para a literatura de mapeamento de robô, apresentando uma nova representação de mapa dos objetos do ambiente, bem como a técnica de construí-lo. Além disso, o algoritmo O-SLAM é o primeiro que usa apenas objetos detectados em imagens para construir um mapa e recuperar a trajetória do robô sem necessidade de qualquer outro sensor nem imagens estéreo. Assim, a abordagem proposta pode ser usada em uma ampla gama de situações, desde pequenos a grandes robôs, uma vez que não dependem de lasers pesados e de elevado consumo de energia ou sensores sonar. Se um mapa e trajetória precisos são procurados, o método proposto pode ser facilmente integrado a outras abordagens SLAM probabilística que utilizam

sensores mais precisos, inclusive ajudando-as em fechar voltas na trajetória.

Além de estender os resultados de trabalhos relacionados que usam MCMC para resolver o problema de associação de dados de uma maneira probabilística através da amostragem sobre as classes de objetos além do correspondente função espacial, este trabalho apresenta novas estratégias para tornar prático o uso de MCMC no problema abordado. Elas incluem o pré-processamento dos dados brutos para remover medidas não coerentes e construir os caminhos do objeto, uma otimização inicial hill climbing, e a proposta de distribuição dos dados direcionados. Embora algumas das estratégias foram, em parte, inspirada por outros trabalhos, novos elementos são introduzidos no presente trabalho, especialmente a otimização do hill climbing realizada antes de iniciar a amostragem com MCMC e a parte da proposta de distribuição relacionada com a amostragem sobre o espaço de classes de objetos.

O rastreamento de objetos produzidos na fase de pré processamento foram inspirados pelo rastreamento característico usado por Kaess & Dellaert (2005). No entanto, a ideia de utilizar os rastros para remover provavelmente falsas medições é original, bem como o método de associação estrita dos dados utilizado para construir os caminhos. Porque ele é muito rigoroso, o algoritmo de pré-associação proposto filtra muitas medições não espúrias também, além de dividir os rastros dos objetos que poderiam ser inicialmente fundidos, afetando o desempenho do algoritmo O-SLAM. Por exemplo, se um objeto é detectado consistentemente por vários quadros de imagem, é ocluído em alguns quadros, e é detectada mais tarde, o algoritmo de pré-associação utilizado irá dividir as medições de objeto em dois rastros, enquanto um melhor algoritmo de associação de dados seria capaz de fundir os dois rastros.

A fase de otimização hill climbing inicial, proposta para achar uma cadeia de estados inicial para o classificador MCMC, é uma contribuição original. Embora a definição de um estado inicial para a cadeia não é um problema relevante em geral (GILKS; RICHARDSON; SPIEGELHALTER, 1996b), ela acelera a convergência do amostrador quando explorar um espaço de estado muito grande, com muitos máximos locais. Sem um bom estado inicial, a cadeia pode explorar o espaço de estados por longo tempo antes de chegar a um estado de "convergência".

Por fim, o modelo probabilístico apresentado para objetos espúrios é uma ideia original, que provou funcionar muito bem na prática, embora exija a definição de um limiar suave, que define o quanto conservador é a aproximação, isto é, como provavelmente vai atribuir uma alta probabilidade espúria a um objeto. Em casos práticos,

no entanto, o mesmo limiar pode ser utilizado para o mesmo algoritmo de detecção de robô e objeto utilizado, uma vez que detém uma forte ligação com a precisão exigida do mapa. Assim, se detectores de objetos muito precisos são usados para obter mapas precisos, um baixo limiar pode ser definido e utilizado para todos os ambientes, desde que os objectos de interesse são ativamente procurados e medidas suficientes são tomadas a partir delas.

7.2 Perspectivas

Uma vez que esta tese apresenta a primeira técnica para SLAM baseados em objetos, o presente trabalho abre muitos ramos em que futuras pesquisas possam ser realizadas. Esta seção discute sobre três eventuais medidas: representação do mapa, reconhecimento de objeto e inferência. O presente trabalho pode ser pensado como um ponto de partida esses ramos de pesquisa, e há ainda muito a fazer.

A representação de objetos por O-Maps pode ser melhorada permitindo que formas geométricas mais genéricas possam ser representadas. Por exemplo, misturar objetos com características geométricas mais básicas, tais como pontos, linhas e planos poderiam aumentar a representação do mapa com paredes, corredores, cantos, etc. Além disso, a utilização de formas geométricas mais complexas poderia enriquecer a representação do mapa, permitindo uma melhor interação homem-máquina, fornecendo possivelmente uma visualização mais realista.

A técnica de detecção de objetos utilizada neste trabalho apresenta sérias limitações para uso em ambientes não controlados. No entanto, substituir CMVision por detectores de objeto mais robustos não iria avançar a ideia subjacente à abordagem proposta, apesar de produzir resultados mais impressionantes. Por sua vez, investigar métodos para o reconhecimento de objetos que não precisem de um algoritmo pronto seria muito interessante para um futuro trabalho nesse ramo. Neste sentido, o trabalho de Ranganathan & Dellaert (2007) Complementa esta tese, e a fusão dos modelos probabilísticos utilizados aqui com os propostos por esse trabalho pode ser um bom caminho a percorrer.

Embora muito tenha que ser feito ainda sobre a representação de mapas no O-Maps e reconhecimento de objetos no O-SLAM, o tema mais desafiador a ser desenvolvido corresponde ao ramo da inferência. Em muitas obras, a formulação exata do problema da associação de dados irrestrita foi substituída por uma heurística, tais

como NN e ML por causa de sua intratabilidade (HÄHNEL et al., 2003). No entanto, técnicas de inferência para distribuição em forma fatorada enfrentaram grandes avanços na última década (WEISS; FREEMAN, 1999; YEDIDIA; FREEMAN; WEISS, 2000; WAINWRIGHT; JAAKKOLA; WILLSKY, 2003; WAINWRIGHT; JAAKKOLA; WILLSKY, 2005), mas não cobriram ainda um extenso campo de aplicação no domínio da robótica. Investigar como esses métodos podem ser aplicados para inferir sobre o espaço das soluções da associação de dados é uma interessante continuação para o presente trabalho. Bayes Variacionais é uma abordagem promissora. Enquanto MCMC executa uma inferência aproximada utilizando a formulação exata da probabilidade de distribuição, métodos Bayes variacionais executam uma inferência exata ou aproximada em uma formulação aproximada da distribuição alvo. Portanto, a abordagem Bayes variacionais pode proporcionar um método para técnicas de inferência recentemente desenvolvidas para uma posterior fatoração aproximada sobre as soluções de associação de dados.

Referências

- ALLEN, C. R.; LEGGETT, I. C. 3D scene reconstruction and object recognition for use with autonomously guided vehicles (AGVs). In: *Proc. IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation*. Orlando, USA, 1995. v. 2, p. 1219–1224.
- BERGER, J. O.; PERICCHI, L. R. *On the justification of Default and Intrinsic Bayes Factor*. West Lafayette, USA, 1995. Tech Report 95-18C.
- BLANCO, J. L.; FERNANDEZ-MADRIGAL, J. A.; GONZALEZ, J. Toward a unified bayesian approach to hybrid metric–topological slam. *IEEE Transactions on Robotics*, v. 24, n. 2, p. 259–270, 2008.
- BRUCE, J.; BALCH, T.; VELOSO, M. Fast and inexpensive color image segmentation for interactive robots. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2000*. Takamatsu, Japan, 2000. p. 2061–2066.
- CAPPÉ, O.; MOULINES, E.; RYDÉN, T. *Inference In Hidden Markov Models*. Heidelberg, Germany: Springer, 2005.
- CASTLE, R. O.; GAWLEY, D. J.; KLEIN, G.; MURRAY, D. W. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2007*. Rome, Italy, 2007. p. 4102–4107.
- CRIMINISI, A.; CROSS, G.; BLAKE, A.; KOLMOGOROV, V. Bilayer segmentation of live video. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, USA, 2006. v. 1, p. 53–60.
- DAVISON, A. J. Real-time simultaneous localisation and mapping with a single camera. In: *Proc. IEEE International Conference on Computer Vision*. Nice, France, 2003. p. 1403–1410.
- DELLAERT, F. *Monte Carlo EM for Data Association and its Applications in Computer Vision*. Tese (Doutorado) — Carnegie Mellon, Pittsburg, USA, 2001.
- DELLAERT, F. Square Root SAM. In: *Proceedings of Robotics: Science and Systems I*. Cambridge, MA, 2005.
- DELLAERT, F.; KAESSE, M. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, v. 25, n. 12, p. 1181–1203, 2006.

- DELLAERT, F.; SEITZ, S. M.; THORPE, C. E.; THRUN, S. Structure from motion without correspondence. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2000*. Hilton Head, USA, 2000. v. 2, p. 557–564.
- DISSANAYAKE, M. W. M. G.; NEWMAN, P.; CLARK, S.; DURRANT-WHYTE, H. F.; CSORBA, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, v. 17, n. 3, p. 229–241, 2001.
- DUDEK, G.; JENKIN, M. *Computational Principles of Mobile Robotics*. Cambridge, UK: Cambridge University Press, 2000.
- DUDEK, G. L. Environment representation using multiple abstraction levels. *Proceedings of the IEEE*, v. 84, n. 11, p. 1684–1704, 1996.
- EKVALL, S.; KRAGIC, D.; JENSFELT, P. Object detection and mapping for service robot tasks. *Robotica*, Cambridge University Press, v. 25, n. 2, p. 175–187, 2007.
- ELFES, A. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, v. 3, n. 3, p. 249–265, 1987.
- ELIAZAR, A. I.; PARR, R. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In: *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico, 2003. p. 1335–1142.
- ELIAZAR, A. I.; PARR, R. DP-SLAM 2.0. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2004*. New Orleans, USA, 2004. v. 2, p. 1314–1320.
- FISCHLER, M.; BOLLES, R. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Communications of the Association for Computing Machinery*, v. 24, n. 6, p. 381–395, 1981.
- FOLKESSON, J.; CHRISTENSEN, H. Graphical SLAM — A self-correcting map. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2004*. New Orleans, USA, 2004. v. 1, p. 383–390.
- FOLKESSON, J.; JENSFELT, P.; CHRISTENSEN, H. I. Graphical SLAM using vision and the measurement subspace. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005*. Edmond, Canada, 2005. p. 325–330.
- FORSSSEN, P. E.; MEGER, D.; LAI, K.; HELMER, S.; LITTLE, J. J.; LOWE, D. G. Informed visual search: Combining attention and object recognition. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2008*. Pasadena, USA, 2008. p. 935–942.
- FOX, D. *Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation*. Tese (Doutorado) — University of Bonn, Bonn, Germany, 1998.
- FRIEDMAN, S.; FOX, D.; PASULA, H. Voronoi random fields: Extracting the topological structure of indoor environments via place labeling. In: *Proc. of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*. Hyderabad, India, 2007. p. 2109–2114.

- GALINDO, C.; SAFFIOTTI, A.; CORADESCHI, S.; BUSCHKA, P.; FERNÁNDEZ-MADRIGAL, J.; GONZÁLEZ, J. Multi-hierarchical semantic maps for mobile robotics. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2005*. Edmond, Canada, 2005. p. 3492–3497.
- GEYER, C. J. Markov chain Monte Carlo maximum likelihood. In: *Proc. of the 23rd Symposium on the Interface*. Fairfax Station, USA, 1991. p. 156–163.
- GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. Introducing Markov chain Monte Carlo. In: GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. (Ed.). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall, 1996. cap. 1, p. 1–20.
- GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. (Ed.). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall, 1996.
- GILKS, W.; ROBERTS, G. O. Strategies for improving MCMC. In: GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. (Ed.). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall, 1996. cap. 6, p. 89–114.
- GOLUB, G.; LOAN, C. V. *Matrix Computations*. Baltimore, USA: Johns Hopkins University Press, 1996.
- GUTMANN, J. S.; KONOLIGE, K. Incremental mapping of large cyclic environments. In: *Proc. IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA '99*. Kyongju, South Korea, 1999. p. 318–325.
- HÄHNEL, D.; BURGARD, W.; WEGBREIT, B.; THRUN, S. Towards lazy data association in SLAM. In: *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*. Sienna, Italy, 2003.
- HASTINGS, W. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, v. 57, p. 97–109, 1970.
- HOETING, J. A.; MADIGAN, D.; RAFTERY, A. E.; VOLINSKY, C. T. Bayesian Model Averaging: A tutorial. *Statistical Science*, v. 14, n. 4, p. 382–417, 1999.
- HOIEM, D.; EFROS, A. A.; HEBERT, M. Putting objects in perspective. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, USA, 2006. v. 2, p. 2137–2144.
- KAESS, M. *Incremental Smoothing and Mapping*. Tese (Doutorado) — Georgia Institute of Technology, Atlanta, USA, 2008.
- KAESS, M.; DELLAERT, F. A Markov chain Monte Carlo approach to closing the loop in SLAM. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2005*. Barcelona, Spain, 2005. p. 643–648.
- KAESS, M.; RANGANATHAN, A.; DELLAERT, F. iSAM: Fast incremental smoothing and mapping with efficient data association. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2007*. Rome, Italy, 2007. p. 1670–1677.

- KASS, R. E.; WASSERMAN, L. The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, American Statistical Association, v. 91, n. 435, p. 1343–1370, 1996.
- KELLY, C. T. *Iterative Methods for Optimization*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1999.
- KHAN, Z.; BALCH, T.; DELLAERT, F. Multitarget tracking with split and merged measurements. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2005*. San Diego, USA, 2005. v. 1, p. 605–610.
- KHAN, Z.; BALCH, T.; DELLAERT, F. MCMC data association and sparse factorization updating for real time multitarget tracking with merged and multiple measurements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 28, n. 12, p. 1960–1972, 2006.
- KLEIN, G.; MURRAY, D. Parallel tracking and mapping for small AR workspaces. In: *Proc. 6th IEEE and ACM International Symposium on Mixed and Augmented Reality ISMAR 2007*. Nara, Japan, 2007. p. 225–234.
- KOLLAR, T.; ROY, N. Using reinforcement learning to improve exploration trajectories for error minimization. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2006*. Orlando, USA, 2006. p. 3338–3343.
- KUIPERS, B. Modeling spatial knowledge. *Cognitive Science*, v. 2, n. 2, p. 129–153, 1978.
- LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *Proc. of the 18th International Conference on Machine Learning*. San Francisco, USA, 2001. p. 282–289.
- LEONARD, J.; COX, I.; DURRANT-WHYTE, H. Dynamic map building for an autonomous mobile robot. *International Journal of Robotics Research*, v. 11, n. 4, p. 286–289, 1992.
- LEONARD, J.; DURRANT-WHYTE, H.; COX, I. J. Dynamic map building for autonomous mobile robot. In: *Proc. IEEE International Workshop on Intelligent Robots and Systems '90. 'Towards a New Frontier of Applications' (IROS '90)*. Ibaraki, Japan, 1990. p. 89–96.
- LIMKETKAI, B.; LIAO, L.; FOX, D. Relational object maps for mobile robots. In: *Proc. of 19th. International Joint Conference on Artificial Intelligence (IJCAI)*. Edinburgh, UK, 2005. p. 1471–1476.
- LIU, Y.; THRUN, S. Results for outdoor-SLAM using sparse extended information filters. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2003*. Taipei, Taiwan, 2003. p. 1227–1233.
- LOWE, D. G. Object recognition from local scale-invariant features. In: *Proc. IEEE International Conference on Computer Vision*. Kerkyra, Greece, 1999. v. 2, p. 1150–1157.

- LU, F.; MILIOS, E. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, Springer Netherlands, v. 4, n. 4, p. 333–349, 1997. ISSN 0929-5593.
- METROPOLIS, N.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, v. 21, p. 1087–1091, 1953.
- MIKOLAJCZYK, K.; LEIBE, B.; SCHIELE, B. Multiple object class detection with a generative model. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, USA, 2006. v. 1, p. 26–36.
- MINKA, T. Automatic choice of dimensionality for PCA. In: *Advances in Neural Information Processing Systems 13*. Vancouver, Canada, 2000. p. 598–604.
- MONTEMERLO, M.; THRUN, S.; KOLLER, D.; WEGBREIT, B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In: *Proc. of the AAAI National Conference on Artificial Intelligence*. Edmonton, Canada, 2002. p. 593–598.
- MORAVEC, H. *Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids*. Pittsburg, PA, 1996. Tech Report CMU-RI-TR-96-34.
- MURPHY, R. *Introduction to AI Robotics*. Cambridge, USA: The MIT Press, 2000.
- NEIRA, J.; TARDOS, J. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, v. 17, n. 6, p. 890–897, 2001.
- NI, K.; STEEDLY, D.; DELLAERT, F. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2007*. Rome, Italy, 2007. p. 1678–1685.
- NIETO, J. I.; GUIVANT, J. E.; NEBOT, E. M. The hybrid metric maps (HYMMs): a novel map representation for DenseSLAM. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2004*. New Orleans, USA, 2004. v. 1, p. 391–396.
- PASKIN, M. A. Thin junction tree filters for simultaneous localization and mapping. In: *Proc. of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*. Acapulco, Mexico, 2003. p. 1157–1164.
- PFISTER, S. T.; ROUMELIOTIS, S. I.; BURDICK, J. W. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In: *Proc. IEEE International Conference on Robotics and Automation (ICRA) 2003*. Taipei, Taiwan, 2003. v. 1, p. 1304–1311.
- PINZ, A. Object categorization. *Foundations and Trends in Computer Graphics and Vision*, v. 1, n. 4, p. 255–353, 2005.
- PORTA, J. M.; KRÖSE, B. J. A. Appearance-based concurrent map building and localization. *Journal of Robotics and Autonomous Systems*, v. 54, n. 2, p. 159–164, 2006.

- QUATTONI, A.; COLLINS, M.; DARREL, T. Conditional random fields for object recognition. In: *Advances in Neural Information Processing Systems 17*. Vancouver, Canada, 2004. p. 1097–1104.
- RABINER, L. R. A tutorial on hidden Markov models and selected application in speech recognition. *Proceedings of IEEE*, v. 77, n. 2, p. 257–286, 1989.
- RANGANATHAN, A. *Probabilistic Topological Maps*. Tese (Doutorado) — Georgia Institute of Technology, Atlanta, USA, 2008.
- RANGANATHAN, A.; DELLAERT, F. Inference in the space of topological maps: An MCMC-based approach. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2004*. Sendai, Japan, 2004. v. 2, p. 1518–1523.
- RANGANATHAN, A.; DELLAERT, F. Semantic modeling of places using objects. In: *Proceedings of Robotics: Science and Systems I*. Atlanta, USA, 2007.
- ROBERTS, G. Markov chain concepts related to sampling algorithms. In: GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. (Ed.). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall, 1996. cap. 3, p. 45–58.
- ROTA, G. C. The number of partitions of a set. *The American Mathematical Monthly*, v. 71, n. 5, p. 498–504, 1964.
- SCHWARZ, G. Estimating the dimension of a model. *Annals of Statistics*, v. 6, n. 2, p. 461–464, 1978.
- SE, S.; LOWE, D.; LITTLE, J. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, v. 21, n. 8, p. 735–758, 2002.
- SELVATICI, A. H. P.; COSTA, A. H. R. Fast loopy belief propagation for topological SAM. In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2002*. San Diego, USA, 2007. p. 664–669.
- SELVATICI, A. H. P.; COSTA, A. H. R. A hybrid adaptive architecture for mobile robots based on reactive behaviours. In: *Mobile Robots: The Evolutionary Approach*. The Netherlands: Springer-Verlag, 2007. cap. 11, p. 161–184.
- SELVATICI, A. H. P.; COSTA, A. H. R.; DELLAERT, F. Object-based visual SLAM: How object identity informs geometry. In: *IV Workshop de Visão Computacional*. Bauru, Brazil, 2008.
- SELVATICI, A. H. P.; COSTA, A. H. R.; DELLAERT, F. Building object-based maps from visual input. Submitted to Robotics: Science and Systems 2009 (RSS'09). 2009.
- SHOTTON, J.; JOHNSON, M.; CIPOLLA, R. Semantic texton forests for image categorization and segmentation. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2008*. Anchorage, UK, 2008. p. 1–8.
- SMITH, R.; SELF, M.; CHEESEMAN, P. Estimating uncertain spatial relationships in Robotics. In: *Autonomous Robot Vehicles*. The Netherlands: Springer-Verlag, 1990. p. 167–193.

- SMITH, R. C.; CHEESEMAN, P. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, SAGE Publications, v. 5, n. 4, p. 56–68, 1986.
- STACHNISS, C.; GRISETTI, G.; BURGARD, W. Information gain-based exploration using Rao-Blackwellized particle filters. In: *Proceedings of Robotics: Science and Systems I*. Cambridge, MA, 2005.
- SWAIN, M.; BALLARD, D. Indexing via color histograms. In: *Proc. IEEE International Conference on Computer Vision*. Osaka, Japan, 1990. p. 390–393.
- THOMAS, A.; FERRAR, V.; LEIBE, B.; TUYTELAARS, T.; SCHIEL, B.; GOOL, L. V. Towards multi-view object class detection. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, USA, 2006. v. 2, p. 1589–1596.
- THRUN, S. Robotic mapping: A survey. In: *Exploring Artificial Intelligence in the New Millenium*. São Francisco, USA: Morgan Kaufmann, 2002. cap. 1, p. 1–38.
- THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics*. Cambridge, USA: The MIT Press, 2005.
- THRUN, S.; FOX, D.; BURGARD, W. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, v. 31, n. 3, p. 29–53, 1998.
- THRUN, S.; MARTIN, C.; LIU, Y.; HAHNEL, D.; EMERY-MONTEMERLO, R.; CHAKRABARTI, D.; BURGARD, W. A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, v. 20, n. 3, p. 433–443, 2004.
- TIERNEY, L. Introduction to general state-space Markov chain theory. In: GILKS, W.; RICHARDSON, S.; SPIEGELHALTER, D. (Ed.). *Markov chain Monte Carlo in practice*. London, UK: Chapman and Hall, 1996. cap. 4, p. 59–88.
- VASSALLO, R. F. *Uso de mapeamentos visuomotores com imagens omnidirecionais para aprendizagem por imitação em robótica*. Tese (Doutorado) — Universidade Federal do Espírito Santo, Vitória, Brazil, 2004.
- VASUDEVAN, S.; GACHTER, S.; BERGER, M.; SIEGWART, R. Cognitive maps for mobile robots — An object based approach. *Journal of Robotics and Autonomous Systems*, v. 55, n. 5, p. 359–371, 2007.
- WAINWRIGHT, M. J.; JAAKKOLA, T. S.; WILLSKY, A. S. Tree-based reparameterization framework for analysis of sum-product and related algorithms. *IEEE Transactions on Information Theory*, v. 49, n. 5, p. 1120–1146, 2003.
- WAINWRIGHT, M. J.; JAAKKOLA, T. S.; WILLSKY, A. S. MAP estimation via agreement on trees: Message-passing and linear programming. *IEEE Transactions on Information Theory*, v. 51, n. 11, p. 3697–3717, 2005.
- WEISS, Y.; FREEMAN, W. T. Correctness of belief propagation in Gaussian graphical models of arbitrary topology. In: *Advances in Neural Information Processing Systems 13*. Vancouver, Canada, 1999. p. 673–679.

- WILLIAMS, B.; KLEIN, G.; REID, I. Real-time SLAM relocalisation. In: *Proc. IEEE International Conference on Computer Vision 2007*. Rio de Janeiro, Brazil, 2007. p. 1–8.
- WINN, J.; SHOTTON, J. The layout consistent random field for recognizing and segmenting partially occluded objects. In: *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) 2006*. New York, USA, 2006. v. 1, p. 37–44.
- YEDIDIA, J. S.; FREEMAN, W. T.; WEISS, Y. Generalized belief propagation. In: *Advances in Neural Information Processing Systems 13*. Vancouver, Canada, 2000. p. 689–695.
- ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, Kluwer Academic Publishers, Hingham, MA, USA, v. 13, n. 2, p. 119–152, 1994. ISSN 0920-5691.
- ZIVKOVIC, Z.; BOOIJ, O. *How did we built our hyperbolic mirror omnidirectional camera — practical issues and basic geometry*. The Netherlands, 2005. Tech Report IAS-UVA-05-04.
- ZIVKOVIC, Z.; BOOIJ, O.; KRÖSE, B. From images to rooms. *Journal of Robotics and Autonomous Systems*, North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, v. 55, n. 5, p. 411–418, 2007. ISSN 0921-8890.
- ZIVKOVIC, Z.; BOOIJ, O.; KROSE, B.; TOPP, E. A.; CHRISTENSEN, H. I. From sensors to human spatial concepts: An annotated data set. *IEEE Transactions on Robotics*, v. 24, n. 2, p. 501–505, 2008. Data set url: <http://staff.science.uva.nl/~zivkovic/FS2HSC/dataset.html>.

Apêndice A – Modelos utilizados em SLAM geométrico

A solução para SLAM geométrico adotada neste trabalho depende da definição de movimento do robô e modelos de observações, representados pelas funções $f(x_{i-1}, v_i)$, $h^u(x_i, l_j)$, e $h^s(x_i, l_j, g_j)$, bem como pelas matrizes de covariância Q_i , $R_{i,k}$, and $W_{i,k}$, tal como definidas na Seção 4.1. Este apêndice descreve como esses modelos são definidos, e como calcular seus parâmetros.

A.1 Modelo de movimento

O modelo de movimento\ do robô descreve como um comando de controle de entrada v_i deve comandar o robô da posição x_{i-1} para a posição x_i . Aqui é assumido que este modelo corresponde a uma função determinista $x_i = f(x_{i-1}, v_i)$ corrompida por ruído branco com covariância Q_i . Supondo o robô se move sobre o piso plano, o controle tem a forma $v_i = (v_i^r, v_i^\phi)$, onde v_i^r é um comando de deslocamento e v_i^ϕ é um comando de mudança de orientação. Seguindo o livro de Thrun, Burgard & Fox (2005), o modelo atual assume que os comandos de deslocamentos podem ser diretamente determinados a partir da leitura dos hodômetros.

Seja a postura do robo definida por $x \triangleq (x, y, \theta)$, conforme representado na figura A.1. Se uma constante de velocidade para frente v e uma taxa de giro ω forem impostas ao robô, a equação de tempo contínuo equação para posicionar o robô é

$$\dot{x} = v \cos(\theta) \quad (\text{A.1})$$

$$\dot{y} = v \sin(\theta) \quad (\text{A.2})$$

$$\dot{\theta} = \omega. \quad (\text{A.3})$$

Assumindo que o comando de controle definido por v_i impõem uma velocidade

Figura A.1: Ilustração das variáveis envolvidas no modelo de movimento do robô. O círculo representa o robô e o segmento de linha dentro indica a orientação com respeito ao sistema de global de coordenadas.

constante e taxa de giro do tempo $i - 1$ para i , de modo que $v_i^r = v$ e $v_i^\phi = \omega$, o deslocamento do robô é dado integrando-se (A.1)-(A.3) para este período é

$$x_i - x_{i-1} = \frac{v_i^r}{v_i^\phi} \sin(\theta_i + v_i^\phi) - \frac{v_i^r}{v_i^\phi} \sin(\theta_i) \approx v_i^r \cos(\theta_{i-1} + v_i^\phi / 2) \quad (\text{A.4})$$

$$y_i - y_{i-1} = -\frac{v_i^r}{v_i^\phi} \cos(\theta_i + v_i^\phi) + \frac{v_i^r}{v_i^\phi} \cos(\theta_i) \approx v_i^r \sin(\theta_{i-1} + v_i^\phi / 2) \quad (\text{A.5})$$

$$\theta_i - \theta_{i-1} = v_i^\phi \quad (\text{A.6})$$

para $|v_i^\phi|$ pequenos o bastante. No entanto, se o deslocamento do robô é desprezível, com $v_i^r \approx 0$, qualquer valor para a mudança de direção encaixa-se no modelo.

O modelo de incerteza do movimento do robô assume que a disparidade entre o movimento comandado e atual é basicamente proporcional ao deslocamento comandado. A matriz de covariância do erro de movimento é dada por $Q_i = Q_i^{add} + Q_i^{mult} v_i$, onde Q_i^{add} representa o nível de erro mínimo e Q_i^{mult} é um fator que multiplica o comando de entrada. Para ambos os robôs utilizados nos experimentos, estes parâme-

Figura A.2: Representação esquemática da câmera omnidireccional usada nos experimentos do capítulo 6 (ZIVKOVIC; BOOIJ, 2005).

tro são

$$Q_i^{add} = \begin{bmatrix} 5\text{mm} & 0 & 0 \\ 0 & 5\text{mm} & 0 \\ 0 & 0 & 1^\circ \end{bmatrix} \quad \text{e} \quad Q_i^{mult} = \begin{bmatrix} 0.1 & 0 \\ 0.01 & 0 \\ 0 & 0.15 \end{bmatrix}.$$

A.2 Modelo de observação

O modelo de observação descreve como uma medida $z_{i,k}$, é obtida pelo robô na postura x_i , quando este detecta o objeto o_j , onde $j = J(i,k)$ e J é a função de correspondência assumida. Além da classe do objeto detectado $a_{i,k}$, cada medição é composta pela posição aparente do objeto na imagem $u_{i,k}$, e o tamanho da projeção do objeto no plano de imagem $s_{i,k}$. Em todos os experimentos descritos neste trabalho, uma câmera omnidireccional foi utilizada, correspondendo à um sistema catadiótrico composto por uma câmera de vídeo e um espelho hiperbólico, como demonstrado na Figura A.2.

Aqui assumimos que, a posição aparente e tamanho medidos são gerados por funções determinísticas $u_{i,k} = h^u(x_i, l_{J(i,k)})$ e $s_{i,k} = h^s(x_i, l_{J(i,k)}, g_{J(i,k)})$, corrompidas por ruído branco com covariâncias $R_{i,k}$ and $W_{i,k}$, respectivamente. A figura A.3 ilustra as geome-

trias básicas descrevendo como estas medidas são tomadas. Objetos são modelados como cilindros e, por questão de simplicidade, as medições são consideradas como geradas pela projeção da seção longitudinal de um objeto. A posição aparente do objeto na imagem $u_{i,k} \triangleq (u_{i,k}^r, u_{i,k}^\phi)$, dadas em coordenadas polares, é a projeção direta do centro da base do cilindro $l_j \triangleq (x_j, y_j, z_j)$. O tamanho aparente do objeto $s_{i,k} \triangleq (s_{i,k}^r, s_{i,k}^\phi)$, é parametrizado pelo comprimento radial $s_{i,k}^r$ e a largura angular $s_{i,k}^\phi$. O comprimento radial é a projeção da altura do objeto h_j , e a largura angular da projeção do diâmetro do objeto d_j . As medidas são dadas por

$$u_{i,k}^r = T\left(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, z_j\right) \quad (\text{A.7})$$

$$u_{i,k}^\phi = \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) - \theta_i \quad (\text{A.8})$$

$$s_{i,k}^r = T\left(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, z_j + h_j\right) - T\left(\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}, z_j\right) \quad (\text{A.9})$$

$$s_{i,k}^\phi = 2 \arctan\left(\frac{d_j}{2\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}\right), \quad (\text{A.10})$$

onde $T(r, z)$ é a distância do centro da imagem, em pixels, resultantes da projeção de um ponto na altura z e distância r do eixo do espelho, nas coordenadas do mundo. O modelo de incerteza do modelo de observação assume um parâmetro único E , em pixels, que reflete a incerteza sobre a projeção de um ponto na imagem. A medida das matrizes de covariância são dadas por:

$$R_{i,k} = \begin{bmatrix} E^2 & 0 \\ 0 & (E/u_{i,k}^r)^2 \end{bmatrix} \quad \text{and} \quad W_{i,k} = 2R_{i,k} \quad (\text{A.11})$$

Em todos os experimentos, a incerteza no plano de imagem utilizada foi $E = 30$ pixels.

A projeção de um ponto do mundo no plano de imagem é explicada por Zivkovic & Booij (2005) e representado na figura A.3. Aqui, o sistema catadióptrico assume o uso de um espelho hiperbólico com seção transversal circular e seção longitudinal hiperbólicas, descritos pela equação

$$\left(\frac{z - z_m + e}{a}\right)^2 - \left(\frac{r}{b}\right)^2 = 1, \quad (\text{A.12})$$

onde $r^2 = x^2 + y^2$ a distância radial a partir do eixo do espelho, a e b são parâmetros do espelho denotando os eixos semi-maior e semi-menor da hipérbole, respectivamente. A excentricidade do espelho é $e = \sqrt{a^2 + b^2}$, e z_m é a altura do ponto focal do espelho. Para um ponto do mundo localizado a uma altura z e distância r do eixo do

Figura A.3: Representação esquemática de como um ponto no mundo é projetado no plano de imagem

espelho, a sua distância de projeção do centro da imagem é dada por

$$T(r, z) = \frac{f}{c} \frac{\lambda(r, z) r}{\lambda(r, z) (z - z_m) + 2e},$$

onde f é a distância focal da lente da câmara, c é o tamanho do pixel

$$\lambda(r, z) = \frac{b^2 \left(e(z - z_m) - a\sqrt{(z - z_m)^2 + r^2} \right)}{b^2(z - z_m)^2 - a^2 r^2}.$$

Os parâmetros da câmera e do espelho usados neste trabalho são:

- Conjunto de dados Cogniron: $a = 42.09\text{mm}$, $b = 25.09\text{mm}$, $z_m = 1.0615\text{m}$, and $f/c = 1715\text{pixels}$
- Conjunto de dados LPA: $a = 39.29\text{mm}$, $b = 19.65\text{mm}$, $z_m = 0.76\text{m}$, and $f/c = 1800\text{pixels}$