

**RECONSTRUÇÃO GEOMÉTRICA DE CENAS NÃO  
ESTRUTURADAS: UMA ABORDAGEM MONOCULAR  
COM PLANEJAMENTO ESTOCÁSTICO**



VILAR FIUZA DA CAMARA NETO

**RECONSTRUÇÃO GEOMÉTRICA DE CENAS NÃO  
ESTRUTURADAS: UMA ABORDAGEM MONOCULAR  
COM PLANEJAMENTO ESTOCÁSTICO**

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

**ORIENTADOR: MARIO FERNANDO MONTENEGRO CAMPOS**

Belo Horizonte

Março de 2012

© 2012, Vilar Fiuza da Camara Neto.  
Todos os direitos reservados.

da Camara Neto, Vilar Fiuza

C172r      Reconstrução Geométrica de Cenas Não  
Estruturadas: Uma Abordagem Monocular com  
Planejamento Estocástico / Vilar Fiuza da Camara  
Neto. — Belo Horizonte, 2012  
xxxvi, 164 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas  
Gerais — Departamento de Ciência da Computação  
Orientador: Mario Fernando Montenegro Campos

1. Computação — Teses. 2. Visão Computacional —  
Teses. 3. Robótica — Teses. I. Orientador. II. Título.

CDU 519.6\*84(043)



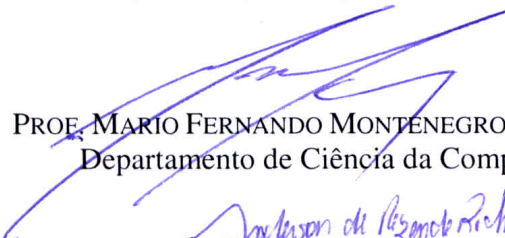
UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO


Reconstrução geométrica de cenas não estruturadas: Uma abordagem monocular  
com planejamento estocástico

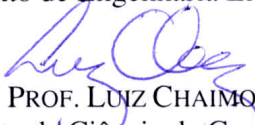
**VILAR FIUZA DA CAMARA NETO**

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROF. ANDERSON DE REZENDE ROCHA  
Instituto de Computação - UNICAMP

  
PROF. BRUNO OTÁVIO SOARES TEIXEIRA  
Departamento de Engenharia Eletrônica - UFMG

  
PROF. LUIZ CHAIMOWICZ  
Departamento de Ciência da Computação - UFMG

  
PROF. SIOME KLEIN GOLDENSTEIN  
Instituto de Computação - UNICAMP

Belo Horizonte, 16 de março de 2012.



*Para os meus filhos, Mariana e Eduardo,  
que me mostraram que a minha vida era incompleta sem eles.*





# Agradecimentos

UM TRABALHO DE DOUTORADO não é, como se pode imaginar à primeira vista, uma peça a quatro mãos executada com um orientador. É um projeto de vida — e, como tal, envolve um time de participantes nada modesto, mais abrangente até do que o círculo dos colegas de laboratório ou dos coautores de publicações.

De fato, o processo todo é, de certa forma, uma prova de resistência e uma batalha psicológica. Meu orientador me advertiu, logo no começo, sobre o que viria pela frente: a pressão familiar e profissional, os conflitos com o próprio orientador, as dúvidas a respeito de ter feito a escolha correta ao entrar no curso, as decepções consigo mesmo e principalmente as várias fases em que questionamos a própria capacidade e cogitamos a desistência, numa montanha-russa de emoções e humores. Na época, incrédulo, anuí sem convicção, e hoje me recordo do aviso com humor e certa reverência profética: ele estava certo.

No fim das contas, muito do que nos mantém nos trilhos está não na determinação pessoal, mas nas pessoas que estão ao redor. Em várias ocasiões, foram a família e os amigos que me ajudaram a sair do atoleiro e reencontrar o rumo. São as declarações de amor e apreço, o acalento nas horas de choro, a presença espontânea e a ajuda inesperada quando o bicho pega, é o “força, hêmi” e o “estamos aí”. É incrível o valor desses gestos para quem os recebe, ainda que os que os fazem geralmente não atinem com a importância deles.

Por isto e por muito mais, é claro que eu não poderia começar os agradecimentos sem falar da minha família. Sem que percebam, talvez sejam os maiores responsáveis pela conclusão deste trabalho. A minha esposa, Giselle, e os meus pais, Araguacy e Vilar Jr., formam o tripé que me manteve equilibrado, mesmo quando o chão me faltou. A eles, jamais serei capaz de manifestar plenamente a minha gratidão.

Aos meus filhos, Eduardo e Mariana, agradeço por terem me proporcionado o prazer incomensurável da paternidade. Embora sejam muito novos ainda para compreender o que é uma pós-graduação, eles também pagaram parte do preço

com os anos de convivência à distância intercalados com uma semana de visita a cada dois meses. São dois faróis, pequenos porém intensos, que mantiveram o meu rumo firme.

Feliz é o estudante que tem como orientador um profissional bem preparado e entusiasta em sua profissão. Mais feliz ainda é quando ele extrapola a figura do mentor intelectual e trata seu pupilo como um aprendiz da vida, não somente de uma ciência específica. Tive a sorte de ter o Prof. Mario Campos como orientador. Dele recebi muito mais do que orientação acadêmica. Não foram poucas as vezes que discursou sobre o valor da família, fazendo-me mesmo colocá-la acima da academia em tal ou qual situação; exercitou meu papel como futuro orientador (“orientando não vem com manual de instruções”, me disse certa vez); e deixou sua marca indelével na minha formação como professor. Aqui agradeço ao mestre, com carinho.

No curso do doutoramento, um dos terrores do estudante é a defesa de proposta de seu trabalho. Trata-se do primeiro grande escrutínio de sua pesquisa por parte da comunidade científica, e mesmo o estudante bem orientado passa as semanas anteriores mergulhado no medo das críticas que receberá. Bom, tive que rever completamente meus conceitos sobre o evento quando passei pela defesa. Uma boa banca é aquela que não só destaca os pontos fracos, mas sugere o rumo a seguir para corrigi-los; que instiga o estudante a considerar possíveis pontos de vista alternativos, sempre tratando o estudante com o respeito devido a um colega em formação; e que ali demonstra que investiu horas de seu tempo para vestir o papel de “orientador de aluguel”, no melhor de seus esforços. A expressão “crítica construtiva” é um clichê, mas representa bem o presente que recebi na ocasião. Agradeço aos professores Siome K. Goldenstein (IC/UNICAMP), Bruno Otávio S. Teixeira (CPDEE/UFMG) e Luiz Chaimowicz (PPGCC/UFMG) por terem feito de minha defesa de proposta uma experiência rica e sem precedentes. Posteriormente, por ocasião da defesa, pude novamente contar com uma nova rodada de contribuições da mesma equipe, dessa vez incluindo o Prof. Anderson Rocha (IC/UNICAMP), a quem estendo os meus agradecimentos.

Ao Prof. Guilherme Augusto S. Pereira, agradeço pelo apoio inestimável na execução dos experimentos práticos. Deu-me acesso irrestrito ao Laboratório de Robótica do Deptº de Engenharia Elétrica, chegando mesmo a sacrificar algumas horas de suas férias para me acompanhar. Sem sua ajuda valiosa, não consigo imaginar as dificuldades que eu teria para levar a termo os experimentos práticos. Muito obrigado!

Estar longe de casa por um tempo prolongado é um teste e tanto para aquele

que, muito ligado à sua terra, se ressentia do afastamento de sua família e de seus amigos. Mais uma vez, nisso também tive muita sorte. Durante a maior parte do tempo que vivi em Belo Horizonte, tive contato constante com uma turma animada de grandes amigos manauaras. Na necessidade, vários me prestaram todo tipo de ajuda, em tantas e tão variadas ocasiões que não é possível enumerá-las. Fora da necessidade, com eles estive sempre próximo das delícias de minha terra: o tucunaré, o tambaqui e o pirarucu, o tucupi e a farinha do Uarini, a banana pacovã frita, o mungunzá, a tapioca e o bolo de macaxeira... Entre necessidades e esbórnias gastronômicas, estiveram sempre conosco: Eduardo e Fabíola Nakamura, Maurício e Ingrid, José Pinheiro e Michelle, Ruitter e Ruth, José Pio e Sylvie, Renata Onety, os irmãos Horácio e David Fernandes e Marco Cristo. No mesmo grupo, não posso deixar de citar amigos que criei e firmei na temporada: Pedro e Aracelly, André “Alla” Lins e Raquel Cabral, Fernando Tanure e Maria do Céu, David “Deivid1” Viscarra e Elbena Noronha, Luiz Henrique, David Menotti, Guillermo Chávez, Martin Gomez, entre tantos outros.

O VeRLab (Laboratório de Visão Computacional e Robótica do DCC/UFMG) foi, por assim dizer, o meu reduto pessoal de debates intelectuais e filosóficos por uma temporada valiosa. Ali aprendi muito e consolidei diversas amizades para a vida. Deixar o VeRLab quase me faz lamentar que a jornada tenha chegado ao fim. Entre tantos que entraram e saíram, não posso deixar de citar: Renato “Python” Cunha, grande miguxo com uma fantástica verve para assuntos filosóficos de toda qualidade (e um providencial conhecimento sobre administração em Linux), e Alice; Marcelo Borghetti, o rei do bife com pimentão; Douglas “Michael Jackson” Macharet; Armando “Pitagoreana Hodográfica” Alves Neto e seu inseparável *shuriken* (este compulsoriamente aposentado por motivos impúblicáveis); Erickson “Buffett” Nascimento, com grandes ideias e discussões sobre como nos tornarmos investidores de peso (e também pelas “curtas caronas” da Pampulha até a Praça da Bandeira), e Marcela; Wagner “Tranquilo e Tenso” Barros e Kátia, bons companheiros de garfo; Renato “Capoeira” Garcia, em especial pelo acolhimento em seu apartamento em várias ocasiões; Dimas “TinyOS” Dutra e a sua indefectível técnica de depuração por LEDs; Daniel “Balbson” Balbino e suas jornadas *Highlander* de programação de 48h sem descanso (mas nas próximas vezes, pelo menos indente o código!); Víctor “Mangá Boy” Costa (e o braço do Nomad, hein?); Leandro “Soriba” Marcolino, escritor e precursor do “Cassino VeRLab” com seu tabuleiro mastigável de Go; Wolmar “555” Pimenta, meu tutor de eletrônica nas horas [que não deveriam ter sido] vagas; Luiz Cantoni pelas várias horas de discussão sobre o que há entre o céu e a terra (também sobre o céu e dentro dos motores de caminhões) e, putz,

pelo inesquecível almoço árabe; e Antônio Wilson “Point Cloud” Vieira, assessor de correspondências de nuvens de pontos e grande cicerone em terras cariocas. Deixo também um abraço caloroso aos diversos colegas com quem me diverti e aprendi em várias ocasiões, como Anderson Pires, Celso Brennand, Elizabeth Duane, Gabriel Leivas, Paulo Drews, Pedro “Comutar” Shiroma e Wallace Santos Lages. Certamente cometi a tremenda indelicadeza de me esquecer de alguns nomes: a esses, peço que não se considerem menos importantes ou que passaram em branco durante a nossa convivência, apenas assumam que sou um desmemoriado sem miolos (o que, aliás, é verdade).

Ao corpo de secretariado do PPGCC devo não só agradecimentos por terem me auxiliado em diversas ocasiões, mas também minha admiração por manterem, com competência e boa vontade singulares, a melhor secretaria que tive o prazer de conhecer em toda a minha vida. Deixo aqui um abraço para Renata Viana, Sheila dos Santos, Túlia Fernandes e Maristela Marques.

Por fim, manifesto meus sinceros agradecimentos à empresa em que trabalho, Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI), por ter me concedido, por meio do Programa de Formação de Mestres e Doutores (PROEMD), a valiosa oportunidade pessoal e profissional de realizar este curso.

*“Nenhum vento ajuda a quem não sabe para que porto velejar.”*

(Michel de Montaigne)



# Resumo

**E**STE TRABALHO ABORDA o problema da reconstrução da geometria de objetos de interesse presentes em um ambiente tridimensional por meio de uma câmera montada em um robô móvel. Embora a reconstrução geométrica seja um dos problemas clássicos da Visão Computacional, a maior parte das soluções apresentadas não trata do problema do *planejamento*, isto é, da determinação das poses futuras necessárias para garantir a conclusão satisfatória da tarefa de reconstrução.

O processo de reconstrução, objeto deste trabalho, é realizado apenas com uma câmera, e não depende de qualquer outro sistema auxiliar para a determinação de pose absoluta ou da existência de marcos visuais dispostos no ambiente. Como consequência, este trabalho se enquadra na classe de problemas conhecidos como SLAM (*Simultaneous Localization and Mapping*), onde a estimação da pose atual do robô está intrinsecamente ligada à tarefa de mapeamento. Mais especificamente, a estimação da geometria do objeto de interesse a partir de imagens depende do conhecimento da sequência de poses da câmera, que por sua vez deve ser estimada a partir de informações acerca das próprias imagens que estão sendo adquiridas.

Como consequência direta do caráter autônomo adotado, a determinação contínua da geometria parcial do objeto é essencial. De fato, a geometria incompleta será a única fonte de informações para o planejador, que identificará áreas carentes de reconstrução e determinará periodicamente os pontos de vista que o robô deve adotar para possibilitar que a câmera seja sempre posicionada para garantir a reconstrução completa do objeto de interesse. Este aspecto torna este trabalho distinto de diversas técnicas de reconstrução que buscam estimar a geometria do objeto a partir de um banco de imagens previamente coletadas.

**Palavras-chave:** Reconstrução visual, SLAM monocular, Exploração robótica autônoma.





# Abstract

**T**HIS WORK FOCUSES ON the autonomous three-dimensional geometric reconstruction of objects, using a single camera mounted on a mobile robot. Although the geometric reconstruction is a classic Computer Vision problem, most approaches up to date do not deal with the planning aspect, i.e., they do not provide autonomous solutions to determine best camera poses in order to obtain the most complete reconstruction possible.

The approach presented here is based solely on images from a single camera and does not depend on any absolute positioning system. As a consequence, this work deals with a class of problems known as *Simultaneous Localization and Mapping*, or SLAM. In other words, the estimation of the object's geometry depends on the estimation of current camera poses, which in turn is computed from data that is extracted from the acquired images.

The stochastic planning technique developed here requires the continuous determination of the object's partial geometry. As a matter of fact, partial reconstruction is the single source of information for the planner, which identifies unexplored and under-explored regions and determines the next pose that the camera must adopt in order to continue the exploratory task. This contrasts with several multiple-view geometry algorithms that estimate the entire object's geometry at once based on a dataset of already available images.

**Keywords:** Visual reconstruction, Monocular SLAM, Autonomous robotic exploration.



# Lista de figuras

1.1	Exemplos de aplicações críticas de robôs autônomos . . . . .	3
1.2	Possíveis interações entre os problemas de Localização e de Reconstrução em sistemas dinâmicos . . . . .	8
2.1	Os campos da robótica exploratória e suas interseções . . . . .	19
2.2	Problema da observação parcial em <i>bearing-only SLAM</i> . . . . .	25
2.3	Abstração do processo de inicialização atrasada em <i>bearing-only SLAM</i> . . . . .	26
2.4	Diagrama de estados para a seleção do objetivo a ser perseguido por um robô . . . . .	33
2.5	Taxonomia das estratégias de reconstrução da geometria de objetos por meio de câmeras . . . . .	33
3.1	Diagrama de organização das camadas e módulos da arquitetura proposta . . . . .	39
3.2	Diagrama da camada de sensores . . . . .	39
3.3	Diagrama da camada de pré-processamento . . . . .	41
3.4	Diagrama da camada de estimação de estado . . . . .	42
3.5	Diagrama da camada de planejamento . . . . .	44
3.6	Diagrama da camada de execução . . . . .	47
4.1	Visão geral da metodologia de correspondência de pontos salientes . . . . .	52
4.2	Determinação dos limites aceitáveis para a razão de escala . . . . .	54
5.1	Ciclo principal de estimação de estados adotado neste trabalho . . . . .	58
5.2	Espaço observável por uma câmera . . . . .	60
5.3	As três formas de associação entre pontos salientes e marcos . . . . .	63
5.4	Fluxograma representativo das operações realizadas no módulo de manutenção de marcos . . . . .	65
5.5	Exemplos das <i>PDFs</i> para a localização de um marco no instante de sua criação . . . . .	72

5.6	Posição das hipóteses recém-criadas (a partir do centro de projeção da câmera) e respectivas incertezas ao longo do eixo de projeção . . . . .	72
5.7	Análise da tendência de privilégio de distâncias entre a câmera e a cena	74
5.8	Exemplo ilustrativo da evolução das PDFs dos estimadores diante da criação e descarte de hipóteses . . . . .	80
6.1	Representação gráfica dos fatores de ocupação de algumas instâncias de volumes de exploração . . . . .	92
6.2	Estimação da direção do vetor normal de um ponto $\mathbf{p}$ da superfície de um objeto, dada uma nuvem de pontos que representa essa superfície .	93
6.3	Avaliação da utilidade das células e dos pontos de vista . . . . .	95
7.1	Conjunto de imagens usadas nos experimentos do módulo de correspondência de pontos salientes . . . . .	102
7.2	Resultados obtidos com valores diferentes para o fator de amostragem, $f_a$ , para o caso de mudança de ponto de vista . . . . .	105
7.3	Resultados obtidos com valores diferentes para o fator de amostragem, $f_a$ , para o caso de aproximação da câmera . . . . .	106
7.4	Resultados obtidos com valores diferentes para o fator de amostragem, $f_a$ , para o caso de rotação da câmera . . . . .	107
7.5	Correspondência para o caso de mudança de ponto de vista (da Figura 7.1(a) para a 7.1(b)) . . . . .	109
7.6	Correspondência para o caso de aproximação da câmera (da Figura 7.1(a) para a Figura 7.1(c)) . . . . .	110
7.7	Correspondência para o caso de rotação da câmera (da Figura 7.1(a) para a Figura 7.1(d)) . . . . .	111
7.8	Vetores de erro de homografia ( $\vec{\xi}_{i,j}$ ) . . . . .	112
7.9	Algoritmo para visualização renderizada dos resultados . . . . .	114
7.10	Configuração experimental para o caso de testes ZOGIS . . . . .	115
7.11	Visualizações renderizadas da geometria reconstruída no caso de teste ZOGIS . . . . .	117
7.12	Análise quantitativa dos resultados para o caso de teste ZOGIS . . . . .	118
7.13	Contagem de marcos resolvidos e hipóteses para o caso de teste ZOGIS	118
7.14	Ilustração da interação entre as plataformas Matlab e Blender adotado nos experimentos simulados . . . . .	119
7.15	Caso de teste SPHERE . . . . .	120
7.16	Análise quantitativa dos resultados para o caso de testes SPHERE . . . . .	121

7.17	Contagem de marcos resolvidos e hipóteses para o caso de teste SPHERE	122
7.18	Caso de teste ICOSAHEDRON . . . . .	123
7.19	Evolução do número de marcos por face para o caso de teste ICOSAHE- DRON . . . . .	124
7.20	Contagem de marcos resolvidos e hipóteses para o caso de teste ICO- SAHEDRON . . . . .	124
7.21	Medidas de erro para o caso de teste ICOSAHEDRON . . . . .	125
7.22	Configuração do caso de teste OWL . . . . .	126
7.23	Algumas imagens da sequência de entrada para o caso de teste OWL . .	128
7.24	Visualizações renderizadas da geometria reconstruída para o caso de teste OWL . . . . .	130
A.1	Ciclo de estimações dos filtros discretos de Kalman . . . . .	142



## Lista de tabelas

7.1	Número de correspondências e das medidas de erro obtidas com o método clássico (com $\tau_M = 1,5$ ) e com o método proposto . . . . .	104
7.2	Comparações de tempo entre o método clássico (com $\tau_M = 1,5$ ) e o método proposto . . . . .	108

# Lista de algoritmos

5.1	Procedimento Corresponde_Imagem_Atual_com_Banco_de_Dados . . . . .	66
5.2	Procedimento Selecciona_Marcos_Resolvidos_Observáveis . . . . .	67
5.3	Procedimento Correspondência_por_Descriptores . . . . .	67
5.4	Procedimento Incrementa_Pontuação_de_Marco . . . . .	68
5.5	Procedimento Atualiza_Descriptor_e_Vetor_de_Reobservação . . . . .	68
5.6	Procedimento Decrementa_Pontuação_de_Marco . . . . .	68
5.7	Procedimento Corresponde_Imagem_Atual_com_Imagem_Recente . . . . .	69
5.8	Procedimento Correspondência_por_Consistência_Geométrica . . . . .	69
5.9	Procedimento Cria_Marcos_e_Hipóteses . . . . .	70
6.1	Procedimento Rasteriza_Reta_em_3D . . . . .	88
6.2	Procedimento Incorpora_Evidências_De_Observação . . . . .	89
6.3	Procedimento Avalia_Utilidade_Exploratória_da_Célula . . . . .	97
B.1	Procedimento Incorpora_Evidência_Em_Probabilidade . . . . .	150



# Lista de símbolos e convenções matemáticas

A seguir é apresentada uma lista dos símbolos e convenções matemáticas usadas neste documento.

<i>Símbolo</i>	<i>Descrição</i>
$\sim$	Distribuição probabilística: $x \sim D$ indica que a distribuição das probabilidades das incertezas da variável $x$ obedece ao modelo $D$ . Em particular, $x \sim \mathcal{N}(\mu, \Sigma)$ indica que a incerteza da variável $x$ obedece à distribuição normal dada por $\mathcal{N}(\mu, \Sigma)$
$\approx$	Relação de igualdade aproximada entre dois valores ou duas distribuições probabilísticas
$\triangleq$	Definição: $A \triangleq B$ indica que $A$ por definição é igual a $B$
:	“Tal que”: Relação de predicado para composição de conjuntos. O conjunto $\{a : b\}$ significa “o conjunto de elementos $a$ que satisfazem o predicado $b$ ”. Por exemplo: $A = \{a : f(a) > 0\}$ indica que $A$ é o conjunto de elementos $a$ que satisfazem $f(a) > 0$
$\alpha_{\text{hip}}$	Razão de incerteza das hipóteses recém-criadas (Subseção 5.2.4)
$\beta_{\text{hip}}$	Razão geométrica de distribuição das hipóteses recém-criadas (Subseção 5.2.4)
$\Delta x_{\text{max}}$	Limite máximo de translação horizontal para aceitação de correspondência entre pontos salientes (Seção 4.3)
$\Delta x_{\text{min}}$	Limite mínimo de translação horizontal para aceitação de correspondência entre pontos salientes (Seção 4.3)
$\Delta y_{\text{max}}$	Limite máximo de translação vertical para aceitação de correspondência entre pontos salientes (Seção 4.3)

<i>Símbolo</i>	<i>Descrição</i>
$\Delta y_{\min}$	Limite mínimo de translação vertical para aceitação de correspondência entre pontos salientes (Seção 4.3)
$\vec{\epsilon}_{i,j}$	Vetor de erro de homografia: diferença entre as coordenadas do ponto saliente $i$ de uma imagem transformadas pela função de homografia e as coordenadas do ponto saliente $j$ de outra imagem (Seção 7.1)
$\Delta\phi_{\max}$	Limite máximo das diferenças de orientação entre correspondências de pontos salientes (Seção 4.3)
$\Delta\phi_{\min}$	Limite mínimo das diferenças de orientação entre correspondências de pontos salientes (Seção 4.3)
$\Delta\phi_{\text{peak}}$	Pico das diferenças de orientação entre correspondências de pontos salientes (Seção 4.3)
$\sigma_{xy}^h$	Desvio-padrão da $h$ -ésima hipótese recém-criada ao longo dos eixos ortogonais ao eixo óptico (Subseção 5.2.4)
$\sigma_z^h$	Desvio-padrão da $h$ -ésima hipótese recém-criada ao longo do eixo óptico (Subseção 5.2.4)
$\tau_{\text{hip}}$	Limiar de probabilidade para o descarte de hipóteses menos prováveis (Subseção 5.2.5)
$\tau_M$	Limiar de distinguibilidade para aceitação de correspondência entre pontos salientes nos métodos tradicionais (Seção 4.1, Eq. (4.2))
$\arg \max_x f(x)$	Argumento do máximo valor: $x_m = \arg \max_x f(x)$ indica que $x_m$ é igual ao valor do argumento $x$ que maximiza $f(x)$
$\arg \min_x f(x)$	Argumento do mínimo valor: $x_m = \arg \min_x f(x)$ indica que $x_m$ é igual ao valor do argumento $x$ que minimiza $f(x)$
$C$	Conjunto de correspondências entre pontos salientes (Seção 4.1)
$C_{\text{final}}$	Conjunto final de correspondências entre pontos salientes (Seção 4.4)
$C_{\text{inic}}$	Conjunto inicial de correspondências entre pontos salientes (Seção 4.2)
$C_{\text{er}}$	Conjunto intermediário de correspondências entre pontos salientes, após as restrições geométricas de escala e rotação (Seção 4.3)

<i>Símbolo</i>	<i>Descrição</i>
$\mathbf{c}_t$	Centro de projeção da câmera do robô no instante $t$ (Subseção 3.3.1)
$d_h$	Profundidade da $h$ -ésima hipótese recém-criada (Subseção 5.2.4)
$d_{\max}$	Distância máxima de observação de características da cena a partir do centro de projeção de uma câmera (Seção 5.1)
$d_{\min}$	Distância mínima de observação de características da cena a partir do centro de projeção de uma câmera (Seção 5.1)
$\mathbf{d}_t^f$	Vetor descritor do $f$ -ésimo ponto saliente detectado na imagem $\mathcal{I}_t$ (Seção 4.1)
dist	Função de distância entre descritores de pontos salientes (Subseção 2.3.2, Seção 4.1)
$F_t^f$	$f$ -ésimo ponto saliente detectado na imagem $\mathcal{I}_t$ (Subseção 3.3.2)
$\mathcal{F}_t$	Conjunto de pontos salientes encontrados em na imagem $\mathcal{I}_t$ (Seção 4.1)
$f_a$	Fator de amostragem para o subconjunto aleatório de pontos salientes (Seção 4.2)
$H_{\text{inic}}$	Número de hipóteses criadas para cada marco novo (Subseção 5.2.4)
$H(\cdot)$	Função de homografia entre duas imagens (Seção 7.1)
$\mathbf{h}_t^{m,h}$	$h$ -ésima hipótese associada ao $m$ -ésimo marco rastreado no instante $t$ (Subseção 3.3.3)
$\mathcal{I}_t$	Imagem capturada pela câmera no instante $t$ (Subseção 3.3.1)
$K$	Dimensão do vetor descritor fornecido por um algoritmo de detecção de pontos salientes (Seção 4.1)
$k_t^{x,y,z}$	Fator de ocupação da célula $(x, y, z)$ da discretização espacial (Subseção 6.1.4, Eq. (6.12))
$l_{\text{incond}}^{x,y,z}$	Logaritmo do razão de chances da probabilidade incondicional ( <i>a priori</i> ) de ocupação da célula $(x, y, z)$ da discretização espacial (Seção 6.1, Eq. (6.5))
$l_t^{x,y,z}$	Logaritmo do razão de chances da probabilidade de ocupação da célula da discretização espacial $(x, y, z)$ , dadas as evidências disponíveis até o instante $t$ (Subseção 6.1.1, Eq. (6.3))
logit( $x$ )	Função logit: $\text{logit}(x) = \log \frac{x}{1-x}$ (Eq. (B.2))
$\mathbf{m}_t^m$	$m$ -ésimo marco rastreado no instante $t$ (Subseção 3.3.3)
MAE( $X$ )	<i>Mean Absolute Error</i> (Média dos Erros Absolutos) de $X$

<i>Símbolo</i>	<i>Descrição</i>
$N_t$	Número de pontos característicos encontrados na imagem $\mathcal{I}_t$ (Subseção 3.3.2)
$\mathbb{N}$	Conjunto de números inteiros não negativos
$\mathcal{N}(\mu, \Sigma)$	Distribuição normal multivariada, sendo $\mu$ o vetor de médias (valores esperados) e $\Sigma$ a matriz de covariância
$O(\cdot)$	Limite assintótico superior
$o_{\text{incond}}^{x,y,z}$	Probabilidade incondicional ( <i>a priori</i> ) de ocupação da célula da discretização espacial $(x, y, z)$ (Seção 6.1, Eq. (6.2))
$o_{\text{livre}}$	Probabilidade de ocupação da célula $(x, y, z)$ da discretização espacial, dado que é possível observar um marco resolvido através dessa célula (Eq. (6.8))
$o_{\text{ocup}}$	Probabilidade de ocupação da célula $(x, y, z)$ da discretização espacial, dado que há um marco resolvido observado nessa célula (Eq. (6.6))
$o_t^{x,y,z}$	Probabilidade de ocupação da célula $(x, y, z)$ da discretização espacial, dadas as evidências disponíveis até o instante $t$ (Subseção 3.3.4, Eq. (3.5))
$\mathbb{P}$	Domínio de probabilidades: $\mathbb{P} = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$
$\mathcal{P}(X)$	Função Densidade de Probabilidade (PDF) sobre a variável aleatória $X$
$\mathbf{q}_t$	Orientação da câmera do robô no instante $t$ , expressada como um quatérnio e referenciada no sistema global de coordenadas (Subseção 3.3.1, Eq. (3.1))
$\mathbf{R}_M$	Matriz de rotação para a avaliação dos limites de translação durante a correspondência de pontos salientes (Seção 4.3)
$\mathbb{R}$	Conjunto de números reais
$\mathbb{R}^+$	Conjunto de números reais não negativos
$\mathcal{R}_M$	Rotações (diferenças de orientação) de um conjunto de correspondências de pontos salientes (Seção 4.3)
$\mathbf{r}_t$	Pose da câmera do robô no instante $t$ , referenciada no sistema global de coordenadas (Subseção 3.3.1, Eq. (3.1))
$\text{RMSE}(X)$	<i>Root Mean Square Error</i> (Erro Quadrático Médio) de $X$
$S_M$	Razões de escala de um conjunto de correspondências de pontos salientes (Seção 4.3)
$\mathbb{S}^2$	Conjunto de vetores unitários em $\mathbb{R}^3$

<i>Símbolo</i>	<i>Descrição</i>
$s_{\max}$	Limite máximo das razões de escala entre correspondências de pontos salientes (Seção 4.3)
$s_{\min}$	Limite mínimo das razões de escala entre correspondências de pontos salientes (Seção 4.3)
$s_{\text{peak}}$	Pico das razões de escala entre correspondências de pontos salientes (Seção 4.3)
$\text{sigm}(x)$	Função sigmoide: $\text{sigm}(x) = \frac{1}{1 + e^{-x}}$ (Eq. (B.5))
$U_E^{x,y,z}$	Utilidade exploratória da célula $(x, y, z)$ (Algoritmo 6.3)
$U_N^{x,y,z}$	Utilidade do isolamento da célula $(x, y, z)$ (Eq. (6.16))
$U_t^{x,y,z}$	Utilidade da célula $(x, y, z)$ (Eq. (6.21))
$\mathbf{u}_t$	Hodometria registrada pelo robô no intervalo de tempo $(t - 1, t]$ (Subseção 3.3.1)
$V(\mathbf{r}_t)$	Função de avaliação do espaço observável (Seção 5.1)
$\vec{\mathbf{v}}$	Vetor de deslocamento transformado de pontos salientes (Seção 4.3)
$(x_t^f, y_t^f)$	Coordenadas do $f$ -ésimo ponto saliente detectado na imagem $\mathcal{I}_t$ (Subseção 3.3.2)
$\mathcal{Z}$	Subconjunto aleatório de pontos salientes (Seção 4.2)



# Lista de acrônimos

A seguir é apresentada uma lista dos acrônimos usados neste documento.

---

<i>Acrônimo</i>	<i>Descrição</i>
<b>3DLRF</b>	<i>3-D Laser Range Finder</i> (Sensor de Distância Tridimensional a Laser)
<b>BO-SLAM</b>	<i>Bearing-Only SLAM</i>
<b>CML</b>	<i>Concurrent Mapping and Localization</i>
<b>EKF</b>	<i>Extended Kalman Filter</i> (Filtro de Kalman Estendido) [Maybeck, 1979]
<b>FoV</b>	<i>Field-of-View</i> (Campo de visada)
<b>FPGA</b>	<i>Field-Programmable Gate Array</i>
<b>GLOH</b>	<i>Gradient Location and Orientation Histogram</i> [Mikolajczyk & Schmid, 2005]
<b>GPGPU</b>	<i>General-Purpose computation on Graphics Processing Units</i>
<b>GPS</b>	<i>Global Positioning System</i> (Sistema de Posicionamento Global)
<b>GPU</b>	<i>Graphics Processor Unit</i> (Unidade de Processamento Gráfico)
<b>GSF</b>	<i>Gaussian Sum Filter</i> (Filtro de Soma de Gaussianas) [Sorenson & Alspach, 1971; Alspach & Sorenson, 1972]
<b>LKF</b>	<i>Linear Kalman Filter</i> (Filtro de Kalman Linear)
<b>LRF</b>	<i>Laser Range Finder</i> (Sensor de Distância a Laser)
<b>MonoSLAM</b>	<i>Monocular SLAM</i>
<b>MPC</b>	<i>Model Predictive Control</i>
<b>MVG</b>	<i>Multiple View Geometry</i> (Geometria Multiocular)
<b>OGM</b>	<i>Occupancy Grid Map</i> (Mapa de Grade de Ocupação) [Moravec & Elfes, 1985; Elfes, 1989]
<b>PCA</b>	<i>Principal Components Analysis</i> (Análise em Componentes Principais, também chamado de Análise de Componentes Principais)

---

<i>Acrônimo</i>	<i>Descrição</i>
<b>PCA-SIFT</b>	<i>Principal Components Analysis SIFT</i> [Ke & Sukthankar, 2004]
<b>PDF</b>	<i>Probability Density Function</i> (Função Densidade de Probabilidade)
<b>RANSAC</b>	<i>Random Sample Consensus</i> [Fischler & Bolles, 1981]
<b>SFM</b>	<i>Structure from Motion</i> (Estrutura a Partir de Movimento)
<b>SIFT</b>	<i>Scale Invariant Feature Transform</i> [Lowe, 1999, 2004]
<b>SLAM</b>	<i>Simultaneous Localization and Mapping</i> (Localização e Mapeamento Simultâneos)
<b>SM</b>	<i>Stochastic Mapping</i> (Mapeamento Estocástico)
<b>SoG</b>	<i>Sum of Gaussians</i> (Soma de Gaussianas)
<b>SPLAM</b>	<i>Simultaneous Planning, Localization and Mapping</i> (Planejamento, Localização e Mapeamento Simultâneos)
<b>SR-UKF</b>	<i>Square-Root Unscented Kalman Filter</i> [van der Merwe & Wan, 2001]
<b>SURF</b>	<i>Speeded Up Robust Features</i> [Bay et al., 2006]
<b>UKF</b>	<i>Unscented Kalman Filter</i> [Julier & Uhlmann, 1997; Wan & van der Merwe, 2000]
<b>U-SURF</b>	<i>Upright SURF</i> [Bay et al., 2006]
<b>UT</b>	<i>Unscented Transform</i> (Transformação <i>Unscented</i> )



# Sumário

<b>Agradecimentos</b>	<b>ix</b>
<b>Resumo</b>	<b>xv</b>
<b>Abstract</b>	<b>xvii</b>
<b>Lista de figuras</b>	<b>xix</b>
<b>Lista de tabelas</b>	<b>xxiii</b>
<b>Lista de algoritmos</b>	<b>xxiv</b>
<b>Lista de símbolos e convenções matemáticas</b>	<b>xxv</b>
<b>Lista de acrônimos</b>	<b>xxxi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivações . . . . .	2
1.2 Apresentação dos problemas-chave . . . . .	4
1.2.1 Descrição de cenas: Mapeamento e reconstrução geométrica .	4
1.2.2 Localização . . . . .	6
1.2.3 A interdependência entre os problemas de Reconstrução Geo- métrica e de Localização . . . . .	7
1.2.4 SLAM em ambientes externos e não estruturados . . . . .	10
1.2.5 Planejamento . . . . .	11
1.3 Formalização do problema e contribuições . . . . .	11
1.4 Organização deste documento . . . . .	13
<b>2 Trabalhos relacionados</b>	<b>15</b>
2.1 Mapeamento estocástico . . . . .	15

2.1.1	Mapeamento Estocástico e os filtros de Kalman . . . . .	17
2.2	Planejamento, Localização e Mapeamento Simultâneos (SPLAM) . . .	18
2.2.1	Estratégias para planejamento restrito à próxima ação . . . . .	19
2.2.2	Estratégias para planejamento de um conjunto de ações . . . . .	21
2.2.3	Análise teórica do ganho de informação em SPLAM . . . . .	22
2.3	SLAM visual . . . . .	23
2.3.1	Os problemas de <i>Bearing-only SLAM</i> . . . . .	25
2.3.2	Identificação e correspondência de pontos salientes em imagens	26
2.3.3	<i>Bearing-only SLAM</i> e SLAM monocular . . . . .	30
2.4	A relação entre o estado-da-arte e este trabalho . . . . .	32
2.4.1	Estimador de estados e mapeamento estocástico . . . . .	32
2.4.2	SLAM ativo (SPLAM) . . . . .	32
2.4.3	SLAM visual . . . . .	33
<b>3</b>	<b>Metodologia</b>	<b>35</b>
3.1	Visão geral . . . . .	35
3.2	Suposições . . . . .	38
3.3	Arquitetura proposta . . . . .	38
3.3.1	Camada de sensores . . . . .	39
3.3.2	Camada de pré-processamento . . . . .	41
3.3.3	Camada de estimação de estado . . . . .	42
3.3.4	Camada de planejamento . . . . .	44
3.3.5	Camada de execução . . . . .	47
<b>4</b>	<b>Correspondência entre pontos salientes de duas imagens</b>	<b>49</b>
4.1	Definições preliminares . . . . .	49
4.2	Correspondência inicial . . . . .	52
4.3	Determinação dos limites geométricos . . . . .	53
4.3.1	Determinação dos limites de escala e rotação . . . . .	53
4.3.2	Determinação dos limites de translação . . . . .	55
4.4	Correspondência final . . . . .	56
<b>5</b>	<b>Estimação de estados</b>	<b>57</b>
5.1	Definições preliminares . . . . .	58
5.2	O módulo de manutenção de marcos . . . . .	60
5.2.1	Marcos resolvidos reobserváveis . . . . .	61
5.2.2	Associação entre pontos salientes e marcos . . . . .	62
5.2.3	Execução do módulo de manutenção de marcos . . . . .	65

5.2.4	Criação de marcos e hipóteses . . . . .	71
5.2.5	Avaliação e descarte de hipóteses . . . . .	74
5.2.6	Descarte de marcos não observados . . . . .	75
5.2.7	Detecção e eliminação de marcos espúrios . . . . .	76
5.3	O módulo SLAM . . . . .	77
5.3.1	Estimadores de estados . . . . .	78
5.3.2	O processo de estimação de estados . . . . .	80
<b>6</b>	<b>Planejamento</b>	<b>83</b>
6.1	O módulo de manutenção do volume de exploração . . . . .	83
6.1.1	Os valores armazenados na estrutura de dados . . . . .	84
6.1.2	Inicialização e atualização das células: Evidências de ocupação e de não ocupação . . . . .	85
6.1.3	O algoritmo de atualização do volume de ocupação . . . . .	87
6.1.4	O fator de ocupação . . . . .	87
6.2	O módulo de planejamento de configurações . . . . .	90
6.2.1	Visão geral . . . . .	90
6.2.2	Bordas de exploração . . . . .	91
6.2.3	Estimação do vetor normal de um ponto da superfície . . . . .	93
6.2.4	Determinação do ponto-alvo . . . . .	94
6.2.5	Determinação da orientação da câmera . . . . .	98
6.2.6	Desistência do plano atual . . . . .	99
<b>7</b>	<b>Experimentos</b>	<b>101</b>
7.1	Módulo de correspondência de pontos salientes . . . . .	101
7.1.1	Escolha do fator de amostragem . . . . .	104
7.1.2	Análise de qualidade e performance . . . . .	107
7.1.3	Conclusões . . . . .	109
7.2	Módulos de manutenção de marcos e de SLAM . . . . .	112
7.2.1	Nota a respeito da exibição das nuvens de pontos . . . . .	113
7.2.2	Caso de teste: ZOGIS . . . . .	115
7.2.3	Resultados e discussões . . . . .	117
7.3	Avaliação do sistema integrado . . . . .	119
7.3.1	Testes simulados . . . . .	119
7.3.2	Experimentos reais . . . . .	126
<b>8</b>	<b>Conclusões e trabalhos futuros</b>	<b>131</b>
8.1	Contribuições . . . . .	131

8.1.1	Correspondência de pontos salientes entre imagens . . . . .	134
8.1.2	Estimação conjunta da geometria do objeto de interesse e da pose da câmera . . . . .	134
8.1.3	Planejamento . . . . .	135
8.2	Limitações e trabalhos futuros . . . . .	136
<b>Apêndice A Filtros de Kalman discretos</b>		<b>141</b>
A.1	O Filtro de Kalman linear . . . . .	143
A.2	O Filtro de Kalman Estendido . . . . .	144
A.3	O Filtro de Kalman <i>Unscented</i> . . . . .	144
A.3.1	A Transformação <i>Unscented</i> . . . . .	145
A.3.2	Os cálculos do Filtro de Kalman <i>Unscented</i> . . . . .	147
<b>Apêndice B Probabilidades e o logaritmo da razão de chances</b>		<b>149</b>
<b>Referências bibliográficas</b>		<b>153</b>

# Capítulo 1

## Introdução

**E**STE TRABALHO TRATA DA RECONSTRUÇÃO GEOMÉTRICA AUTÔNOMA de objetos não estruturados por meio de robôs dotados de câmeras. Em Visão Computacional, “reconstruir a geometria” de uma determinada cena consiste em criar uma descrição geométrica das entidades visíveis que compõem a cena. A forma e a posição de objetos, a topografia do terreno e a planta baixa de um andar são exemplos típicos de informações obtidas a partir da reconstrução geométrica.

A autonomia na realização da tarefa é um ponto-chave deste trabalho. A reconstrução será realizada sem interferência humana, buscando soluções de atuação para reduzir o tempo e a energia necessários para completar a missão e aumentar a confiabilidade dos resultados obtidos. Em particular, a atuação do robô será periodicamente planejada durante a execução da tarefa, identificando áreas carentes de reconstrução e decidindo a melhor estratégia de observação (a trajetória do robô e o posicionamento da câmera) para efetuar a cobertura.

“Ambientes estruturados” são aqueles que, construídos ou modificados pelo homem, possuem características visuais ou geométricas simples e facilmente identificáveis, que podem ser aproveitadas para auxiliar a navegação de um robô e simplificar a representação geométrica do ambiente. Elementos arquitetônicos, como paredes e portas (que geralmente apresentam retas, planos e quinas), e faixas rodoviárias são exemplos muito usados. Em contrapartida, “ambientes não estruturados” são os que não apresentam essas características.

O restante deste capítulo é dedicado à análise mais profunda do problema abordado neste trabalho e é organizado da seguinte maneira: a [Seção 1.1](#) discute as motivações que despertaram o interesse para o desenvolvimento deste trabalho; a [Seção 1.2](#) formaliza alguns conceitos necessários para a definição mais precisa do problema a ser abordado, que é discutido na [Seção 1.3](#); e a [Seção 1.4](#) apresenta

a organização do documento.

## 1.1 Motivações

A percepção popular sobre a Robótica é bastante variada e em geral se polariza em dois nichos opostos. De um lado estão aqueles que, influenciados pela visão artística de livros e filmes, percebem a Robótica como o estudo e desenvolvimento de máquinas inteligentes, emocionais e com aparência orgânica, em particular humana. No extremo oposto estão os que entendem que o conceito é muito mais abrangente, incluindo qualquer aparato mecânico com braços ou peças móveis com o objetivo de ajudar ou substituir o homem em suas tarefas.

Nenhum dos dois extremos é preciso como definição. Formalmente, o que é crítico é que um robô seja *autônomo*, isto é, ele deve ser capaz de tomar decisões, com base em sua percepção do ambiente (obtida por *sensores*), sobre suas ações futuras (realizadas por *atuadores*). Desta definição estão excluídos todos os equipamentos cujas ações sejam resultado direto da operação remota por seres humanos (os chamados *sistemas teleoperados*) ou da obediência precisa e repetitiva de um conjunto de movimentos previamente programados, como alguns braços mecânicos industriais.

O crescente interesse dedicado à Robótica, em especial à Robótica Móvel, se deve em parte pelo fato de que há muito tempo os robôs deixaram de ser meros substitutos do homem. Deixando de lado as comparações mais óbvias (capacidade de processamento de dados, velocidade e precisão de atuação, imunidade a cansaço, distrações e tédio, etc.), o problema é que várias tarefas só podem ser executadas por robôs. (Algumas poderiam teoricamente ser executadas por seres humanos, mas a um custo atualmente inviável.)

Examinemos três casos interessantes:

1. *Exploração extraterrena*: Nas últimas décadas, um número crescente de sondas não tripuladas têm sido enviadas a vários astros do Sistema Solar. Nesse contexto, Marte tem sido foco de interesse e destino recorrente nos últimos anos. Atualmente não há tecnologia para enviar naves tripuladas a Marte, e a teleoperação (a partir da Terra) é inviável por causa do tempo de propagação do sinais de comunicação: por exemplo, a última missão envolvendo sondas móveis (*rovers*), a *Mars Exploration Rover* da NASA (Figura 1.1(a)), está em operação desde janeiro de 2004; nesse tempo, a distância média entre Marte e a Terra foi de cerca de 14 minutos-luz, com picos que ultrapassaram 22



**Figura 1.1.** Exemplos de aplicações críticas de robôs autônomos. **(a)** Um dos veículos robóticos enviados a Marte (*Spirit* e *Opportunity*) em 2004, como parte da *Mars Exploration Rover Mission* da NASA (concepção artística; crédito: Maas Digital LLC, NASA/JPL); **(b)** Veículo teleoperado *Groundhog* para mapeamento tridimensional de minas subterrâneas [Thrun et al., 2003]. Nota-se o cordão umbilical para a comunicação com o operador externo.

minutos-luz. Portanto, o envio de veículos robóticos — capazes de receber comandos de alto nível, como “caminhar até tal região de interesse” ou “analisar aquela formação rochosa”, e decidir autonomamente os detalhes da execução do comando — é simplesmente a única alternativa possível nesse cenário.

2. *Explorações em ambientes fechados:* Frequentemente os objetos de interesse de arqueólogos, paleontólogos e espeleólogos encontram-se encerrados em ambientes fechados, como formações naturais (cavernas, grutas) ou estruturas antigas (pirâmides, ruínas, etc.). Parte da dificuldade de se deslocar em tais ambientes está nos perigos para uma eventual equipe de exploração: instabilidade estrutural, presença de gases tóxicos ou falta de oxigênio, presença de explosivos não detonados, etc. Nesse caso, o emprego de robôs é uma alternativa importante para evitar o risco desnecessário a vidas humanas. A teleoperação nesse caso é complicada porque o meio impede a comunicação sem fio com o ambiente externo, e a comunicação com fio (Figura 1.1(b)) limita a mobilidade do aparelho. Portanto, idealmente os robôs empregados devem ter autonomia para se movimentar e tomar decisões sobre a melhor estratégia de exploração.
3. *Operações de busca e salvamento:* Em cenários de catástrofes causadas por deslizamentos, desmoronamentos, incêndios, explosões e outras situações,

as equipes de socorro são levadas a trabalhar no limite de sua capacidade, principalmente quando vidas humanas estão em jogo. Para piorar, os recursos disponíveis para o planejamento da operação muitas vezes são escassos: o tempo é curto para a tomada de decisões, e os mapas do ambiente afetado, quando disponíveis, podem se revelar inúteis por causa das modificações causadas pelo acidente [Ferranti et al., 2007].

Nesse contexto, a adoção de robôs especializados pode trazer um apoio crucial para as equipes de resgate. Várias aplicações podem ser identificadas, como por exemplo: localização de vítimas e prestação de socorro básico; mapeamento do ambiente [Ryde & Hu, 2006] e planejamento de rotas seguras; guiamento das equipes de socorro, tanto para infiltração quanto para evasão; e identificação de áreas de risco. Em alguns casos, a teleoperação é uma opção possível; no entanto, a força de trabalho humano em campo é um recurso valioso, de modo que a autonomia dessas unidades de apoio deve ser considerada um atributo altamente desejável.

Vários aspectos desses cenários de motivação serão revistos nas seções seguintes, com o objetivo de sustentar a definição formal dos problemas abordados neste trabalho e identificar as áreas de pesquisa relevantes.

## 1.2 Apresentação dos problemas-chave

Esta seção tem por objetivo apresentar e discutir algumas definições e problemas de Robótica Móvel e Visão Computacional, necessários para a compreensão do objetivo a ser alcançado pelo trabalho em andamento.

### 1.2.1 Descrição de cenas: Mapeamento e reconstrução geométrica

Todos os cenários apresentados como motivações deste trabalho tratam de uma premissa básica: *Os robôs devem adquirir conhecimento sobre a geometria do ambiente em que estão atuando*, tanto para cumprir a missão de prover informações aos operadores remotos quanto para tomar decisões e planejar suas próprias ações futuras. De fato, a captura de informações geométricas de uma cena é uma tarefa recorrente em várias áreas de interesse, e transcendem em muito a Robótica Móvel.

Em sua proposição mais simples, o Problema de Descrição de Cenas pode ser apresentado da seguinte forma:



**Definição 1 (Problema de Descrição de Cenas):** *A partir de sensores cujas poses absolutas (em relação a um referencial estacionário) são conhecidas, construir uma representação geométrica da cena com base nos dados capturados por esses sensores.* □

Esta proposição é bastante genérica, pois não sugere o uso de robôs (móveis ou estáticos). Com efeito, a definição se aplica a áreas totalmente diversas à Robótica, como a tomografia computadorizada ou a aerofotogrametria.

Em Robótica Móvel, o problema é tipicamente apresentado e tratado com algumas peculiaridades. Além de considerar que os sensores são embarcados em um veículo móvel (nesse caso, a posição relativa entre os sensores e um referencial fixo no corpo do robô em geral também é conhecida), é comum que a reconstrução seja feita ao longo do tempo — ou seja, a posição absoluta dos sensores não permanece estática. Nesse escopo, duas definições similares são recorrentes:

**Definição 2 (Problema de Mapeamento):** *A partir de um robô móvel cujas poses absolutas (em relação a um referencial estacionário) ao longo do tempo são conhecidas, construir o mapa da cena com base nos dados capturados pelos sensores embarcados.* □

**Definição 3 (Problema de Reconstrução Geométrica com Robô Móvel):** *A partir de um robô móvel cujas poses absolutas (em relação a um referencial estacionário) ao longo do tempo são conhecidas, construir uma representação geométrica de objetos de interesse da cena com base nos dados capturados pelos sensores embarcados.* □

Com efeito, a diferença fundamental entre as duas proposições está no foco de interesse do que deve ser reconstruído. No primeiro caso, visa-se obter o *mapa* do ambiente (“uma lista de objetos no ambiente e as suas respectivas localizações” [Thrun et al., 2005]), útil para a posterior navegação (por seres humanos ou por outros robôs) no mesmo ambiente; no segundo caso, o foco recai sobre uma ou mais *entidades* da cena.

Curiosamente, apesar da proximidade filosófica entre os dois problemas, dificilmente eles são tratados como sendo de uma mesma classe. A explicação é simples:

- No caso do mapeamento, espera-se recompor o mapa de um ambiente cuja dimensão espacial é potencialmente muito maior do que a dimensão do próprio robô (portanto, o volume de dados é alto). Entretanto, a precisão da

reconstrução não é crítica: deve ser suficiente apenas para que seja uma base confiável para posterior navegação no ambiente;

- No caso da reconstrução de objetos, as proposições são inversas: Enquanto o objeto possui dimensões espaciais mais limitadas, em geral objetiva-se uma reconstrução bem mais precisa, possivelmente combinada com outras características, como propriedades radiométricas, composição físico-química, etc.

A comunidade científica já apresentou diversas soluções para esses dois problemas, e (a não ser quando a proposição do problema inclui particularidades não discutidas nas definições aqui apresentadas) são ambos considerados satisfatoriamente resolvidos.

### 1.2.2 Localização

As definições dos problemas apresentados na [Subseção 1.2.1](#) apresentam uma restrição forte, embora sutil: dependem todas do conhecimento preciso da posição absoluta dos sensores (ou dos robôs). Pode-se, portanto, considerar que tais problemas dependem da determinação prévia da pose do robô (a cada instante, no caso de sistemas dinâmicos).

Intuitivamente, o Problema de Localização pode ser formalizado como segue:

**Definição 4 (Problema de Localização):** *Estimar a localização de um robô móvel em relação ao ambiente.* □

Pode-se pensar que o Problema de Localização pode ser trivialmente resolvido com o uso de sistemas como o [Sistema de Posicionamento Global \(Global Positioning System, ou GPS\)](#). Na prática, porém, nem sempre é possível dispor de um sistema de localização absoluta: o [GPS](#) possui várias limitações quanto à precisão, cobertura e portabilidade, que podem ser críticos conforme a aplicação; outros sensores exteroceptivos (como câmeras externas que observam os robôs) requerem a preparação prévia de um ambiente bem estruturado, o que muitas vezes não é possível; e sensores proprioceptivos (embarcados no robô, como hodômetros ou sensores inerciais) são fundamentalmente falhos porque acumulam erros e desvios (*drift*) ao longo do tempo [[Thrun et al., 2002](#)]. Portanto, a *localização* pode se revelar um problema nada trivial, e de fato muitos consideram que esse é o problema fundamental da robótica móvel [[Cox, 1991](#); [Ceccarelli et al., 2006](#)].

Uma abordagem bastante utilizada em Robótica Móvel para o Problema de Localização está em prover o robô de uma descrição do ambiente em que ele

navegará. Com isso, pode-se manter continuamente uma estimaco acerca da posico do rob; sempre que o ambiente é observado, esses dados so comparados com aqueles que deveriam ser obtidos de acordo com o modelo terico conhecido *a priori*, e essa comparaco é usada para ajustar a estimaco corrente da pose do rob.<sup>a</sup> Com isso, é possvel lanar uma definico mais restrita para o problema de localizaco:

**Definico 5 (Problema de Localizaco com Mapa Conhecido):** *A partir do conhecimento prvio do mapa do ambiente, estimar a localizaco absoluta de um rob mvel em relaco a esse ambiente.* □

Essa abordagem é particularmente vantajosa porque no depende de sensores como o GPS e porque no acumula desvios com o tempo, desde que seja possvel observar continuamente pontos de referncia no ambguos. O uso de mapas previamente conhecidos em conjunto com outros sensores (hodmetros, bssolas, acelermetros) permite manter uma estimativa bastante precisa da pose do rob mesmo com um certo grau de ambiguidade do mapa.

### 1.2.3 A interdependncia entre os problemas de Reconstruco Geomtrica e de Localizaco

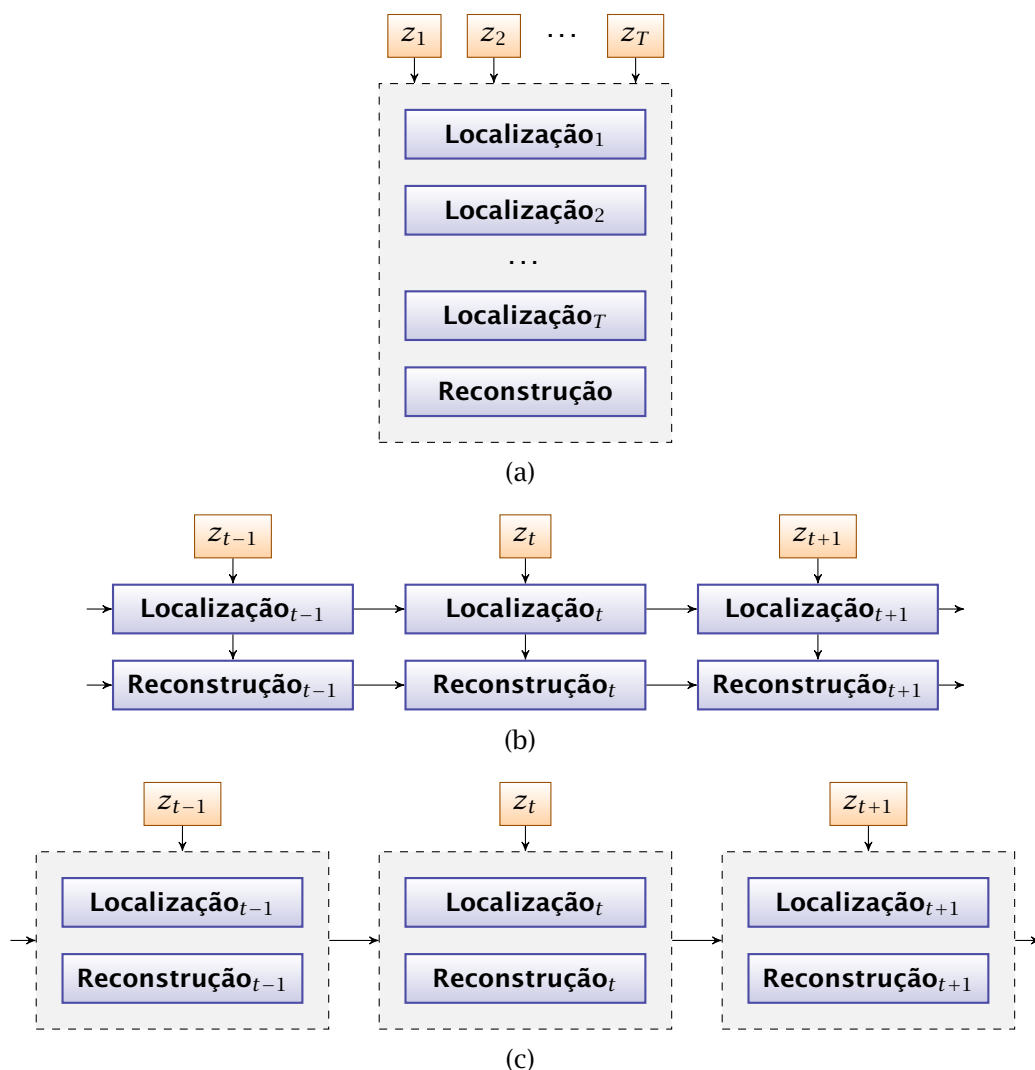
De acordo com as proposices lanadas anteriormente, as Definices 2 (Problema de Mapeamento) e 5 (Problema de Localizaco com Mapa Conhecido) so incompatveis entre si — no primeiro caso o mapeamento depende da localizaco precisa, enquanto no segundo a localizaco baseia-se no conhecimento prvio do mapa.

Infelizmente, é comum encontrar situaes prticas em que as duas tarefas devem ser resolvidas em conjunto, a exemplo dos cenrios discutidos como motivaes na Seo 1.1. Em termos gerais, sempre que a localizaco e o mapeamento (ou a reconstruco geomtrica) forem resolvidos em paralelo, pode-se identificar trs abordagens distintas (Figura 1.2):

1. *Soluo global off-line* (Figura 1.2(a)): Faz-se o rob transitar pelo ambiente coletando dados; em seguida, busca-se uma soluo global que melhor explique a trajetria do rob (a localizaco em cada instante) e a geometria da cena. Vrios algoritmos de *Estrutura a Partir de Movimento (Structure from Motion, ou SFM)* e *Geometria Multiocular (Multiple View Geometry, ou MVG)* tratam da reconstruco *off-line* de ambientes paralelamente à recuperao

---

<sup>a</sup>Esse processo é anlogo ao de estarmos em um veculo, acompanhando a trajetria com o mapa rodovirio da regio.



**Figura 1.2.** Possíveis interações entre os problemas de Localização e de Reconstrução em sistemas dinâmicos, onde  $z_t$  representa o conjunto de dados obtidos pelos sensores em um instante  $t$ . **(a)** Busca de uma solução ótima para o problema após a coleta de todos os dados (solução estática); **(b)** Solução progressiva, considerando-se o Problema de Localização independentemente do Problema de Mapeamento (“localização seguida de mapeamento”); **(c)** Solução progressiva, considerando-se a interdependência entre os dois problemas. No caso de reconstrução de mapas, este último cenário caracteriza o problema de SLAM.

de trajetória. Em comparação com as demais abordagens, essas são mais precisas e robustas nas situações em que podem ser aplicadas [Davison, 2003]. No entanto, todo o processo de obtenção de dados não pode ser realizado de forma autônoma, já que durante essa etapa as informações essenciais para a navegação (localização e mapa) não estão disponíveis. Nesse

caso, a necessidade de teleoperação torna a abordagem incompatível com as motivações apresentadas para este trabalho;

2. *Solução incremental de localização seguida de mapeamento* (Figura 1.2(b)): A cada instante, o problema de localização é resolvido independentemente do mapeamento, e a reconstrução geométrica é progressivamente executada a partir da pose recém-recuperada dos sensores. Embora o prerequisite para o tratamento do Problema de Descrição de Cenas seja satisfeito — a disponibilidade de uma estimativa da posição absoluta do sensor —, a localização está sujeita às questões e limitações discutidas na Subseção 1.2.2: precisão e indisponibilidade do GPS, desvios cumulativos, etc.;
3. *Solução incremental de localização e mapeamento simultâneos* (Figura 1.2(c)): A cada instante, a observação do ambiente é utilizada para avaliar simultaneamente a melhor localização corrente dos sensores e a reconstrução parcial da cena. Essa é a alternativa matematicamente mais complexa e computacionalmente mais cara, e somente nos últimos anos foi possível encontrar soluções satisfatórias [Durrant-Whyte & Bailey, 2006; Bailey & Durrant-Whyte, 2006].

A última abordagem é o cerne do problema de *Localização e Mapeamento Simultâneos* (*Simultaneous Localization and Mapping*, ou SLAM), também chamado de *Concurrent Mapping and Localization* (CML), que pode ser assim formalizado:

**Definição 6 (Problema de Localização e Mapeamento Simultâneos):** *Construir incrementalmente uma representação geométrica do ambiente e estimar simultaneamente a localização de um robô em relação a essa representação, a partir dos dados fornecidos pelos sensores do robô.* □

Esse problema já foi descrito como sendo o “Santo Graal” da comunidade de robótica móvel, já que é o ponto crucial para criar robôs verdadeiramente autônomos [Russell & Norvig, 2003; Durrant-Whyte & Bailey, 2006].

Já foi argumentado anteriormente (Subseção 1.2.1) que os problemas de Mapeamento e Reconstrução Geométrica podem ser considerados de uma mesma classe. Dessa forma, mesmo que historicamente o SLAM tenha sido quase que exclusivamente utilizado para recompor um mapa do ambiente, neste trabalho ele será utilizado para criar uma representação geométrica de objetos de interesse.<sup>b</sup>

---

<sup>b</sup>Por esse motivo, a Definição 6 foi deliberadamente redigida de modo a citar “a construção de uma representação geométrica”, em lugar da definição mais clássica (e mais restritiva) que cita “a construção de um mapa”.

#### 1.2.4 SLAM em ambientes externos e não estruturados

Muitas soluções foram propostas para o problema de SLAM desde que ele foi formalizado no artigo de Smith & Cheeseman [1986]. Em sua grande maioria, essas soluções adotam duas restrições significativas:

- O ambiente explorado é *interno* (*indoor*), ou seja, a movimentação é feita sobre superfícies planas horizontais e não é necessário considerar problemas como trepidação ou deslizamento severo. Em contrapartida, um robô terrestre operando em ambientes externos (*outdoor*) certamente sofrerá movimentos de arfagem e rolamento, de modo que algoritmos nesse cenário devem ser tratados como problemas tridimensionais [Howard et al., 2004];
- Os ambientes são *estruturados*: é possível detectar estruturas geométricas (planos, arestas, etc.) ou visuais (retas, quinas, retângulos, etc.) com relativa facilidade, o que reduz significativamente os custos de armazenamento, busca e correspondência das informações coletadas. Por outro lado, a operação em ambientes não estruturados deve tratar com características menos óbvias, de modelagem mais complexa e observação mais difícil.

No âmbito dos problemas de mapeamento (e SLAM) em três dimensões, há uma dificuldade que é abordada de maneira recorrente na literatura: trata-se dos sensores usados para capturar a geometria (tridimensional) da cena. Uma possibilidade é o uso de Sensores de Distância Tridimensional a Laser (*3-D Laser Range Finders*, ou 3DLRFs). Ryde & Hu [2007] notam que há vários 3DLRFs comerciais, porém o custo de um equipamento desses ainda é particularmente elevado para a maioria das aplicações, especialmente quando se considera o uso de várias unidades com times de robôs. Alguns pesquisadores criaram soluções para adaptar sensores bidimensionais (bem mais baratos) a bases ou espelhos rotativos, de modo a conseguir varreduras tridimensionais, mas essas soluções demandam um controle complexo e apresentam consumo de energia elevado [Ryde & Hu, 2006]. De um modo ou de outro, a maior parte desses sensores foi projetada para o uso estacionário, ou pelo menos em veículos lentos, por conta do tempo gasto para executar uma varredura completa (tipicamente na ordem de vários segundos).

Outras pesquisas abordam o problema de maneira totalmente diversa e usam câmeras como base para a percepção tridimensional. Essas soluções são menos precisas do que as dependentes de 3DLRFs e exigem um tratamento cuidadoso das incertezas dos resultados e da ordem de complexidade dos algoritmos usados, especialmente quando a aplicação exige a estimação em tempo real.

### 1.2.5 Planejamento

A maior parte das pesquisas realizadas em **SLAM** deixa de lado questões sobre as decisões acerca do controle a ser aplicado ao robô: subentende-se que ele é teleoperado, ou que a movimentação independe de todo o processo de localização e reconstrução (ou seja, são utilizados algoritmos de caminhar aleatório — *wandering* — ou seguimento de paredes, por exemplo). Perde-se, portanto, a oportunidade de avaliar futuras configurações do robô que maximizem a expectativa de incorporação de informação acerca do conhecimento corrente do ambiente.

Por outro lado, nenhuma missão de **SLAM**, por definição, pode realizar planejamento a longo prazo [Huang et al., 2005]: afinal de contas, o planejamento somente pode ser feito com base em um horizonte limitado de informações sobre o ambiente (já que o conhecimento acerca dele é adquirido aos poucos).

Com essas limitações em vista, este trabalho apresenta critérios para decidir a melhor pose futura de cada robô a partir da informação disponível até o presente. Como será visto no **Capítulo 3**, tais critérios visam encontrar um equilíbrio entre três objetivos contraditórios: (i) a manutenção das incertezas de localização abaixo de certos limites previamente definidos; (ii) a exploração de vistas desconhecidas do objeto de interesse; e (iii) o refinamento da informação previamente obtida em áreas já exploradas.

## 1.3 Formalização do problema e contribuições

Considera-se um ambiente com as seguintes características:

- *Tridimensional*: A geometria dos objetos estende-se arbitrariamente no espaço tridimensional;
- *Não estruturado*: A cena não é necessariamente composta por elementos cuja geometria possa ser razoavelmente explicada por modelos simples (retas, circunferências, planos, esferas, cilindros, etc.);
- *Estático*: Desconsiderando o veículo (e suas partes móveis), os objetos que compõem a cena permanecem estacionários ao longo do tempo;
- *Lambertiano*: A superfície dos objetos de interesse apresenta características lambertianas, isto é, sem especularidades. Apenas efeitos de iluminação global são considerados, ou seja, não há efeitos de transparência e/ou translucência.

Considera-se que um veículo autônomo (robô móvel) está presente nesse ambiente. As seguintes propriedades são atribuídas ao veículo:

- O veículo é equipado com um sensor de imagem (câmera), possivelmente dotado de capacidade de movimento (*pan-tilt*) e/ou *zoom*;
- O veículo não possui meios de estimar diretamente a sua localização segundo um sistema de coordenadas global (por exemplo, não é dotado de sensores **GPS**).

Considera-se também que o ambiente possui um objeto de particular interesse, cujas características geométricas são inicialmente desconhecidas.

A partir das considerações lançadas, o problema a ser tratado pode ser formalizado da seguinte maneira:

**Definição 7 (Problema Principal):** *Dados um objeto de interesse em um ambiente tridimensional e um robô móvel dotado de uma câmera, planejar e realizar uma estratégia de ações para construir uma representação geométrica do objeto de interesse, representada por um conjunto de pontos tridimensionais  $\mathcal{M} = \{\mathbf{m}^1, \dots, \mathbf{m}^M\}$  da superfície do objeto.* □

O conjunto de contribuições propostas neste trabalho pode ser compreendido como o conjunto das três metodologias apresentadas a seguir:

1. *Planejamento da atuação dos robôs em ambientes tridimensionais:* Os trabalhos que tratam de **Planejamento, Localização e Mapeamento Simultâneos** (*Simultaneous Planning, Localization and Mapping*, ou **SPLAM**) restringem-se a ambientes bidimensionais. Isso se deve ao fato de que as métricas desenvolvidas baseiam-se na representação do ambiente por meio de *Occupancy Grid Maps* (**OGMs**) — uma forma de representação de estados que, pela sua complexidade de armazenamento, não pode ser escalada para ambientes tridimensionais. Este trabalho apresenta métricas de avaliação de expectativa de ganho de informação aplicáveis a ambientes tridimensionais.
2. *Reconstrução progressiva de objetos de interesse com vias à autonomia:* O processo contínuo de planejamento de ações futuras depende da análise da reconstrução parcial obtida até o momento. Em particular, o planejamento apresentado neste trabalho requer um conjunto de informações que extrapola a geometria parcial e a estimação da pose corrente do robô, já que essas informações são insuficientes para diferenciar os pontos de vista não visitados (a fim de prosseguir com a exploração do objeto) dos previamente visitados (que permitem o refinamento da estimação da geometria do objeto e a correção



da pose estimada do robô). Este trabalho apresenta uma metodologia de reconstrução progressiva aliada à estimação da pose do observador, de modo a fornecer informações essenciais para a execução do planejamento em cenas tridimensionais.

3. *Correspondência geometricamente consistente de características visuais entre pares de imagem*: Há diversos métodos modernos para a identificação e assinatura de pontos de salientes em imagens, como *Scale Invariant Feature Transform* (SIFT) [Lowe, 1999, 2004] e *Speeded Up Robust Features* (SURF) [Bay et al., 2006]. Em geral, a associação de pontos salientes entre pares de imagens é realizada com base na similaridade das assinaturas, sem levar em consideração a consistência geométrica do pareamento. Este trabalho apresenta uma metodologia para realizar esta associação que elimina a maior parte das associações errôneas (*outliers*) observadas em outros métodos.

## 1.4 Organização deste documento

O restante deste documento está organizado em diversos capítulos, apresentados a seguir:

- *Trabalhos relacionados* (Capítulo 2): Apresenta uma extensa revisão da literatura científica relacionada ao tema desta tese, discutindo o estado-da-arte e identificando temas não explorados pela comunidade científica;
- *Metodologia* (Capítulo 3): Descreve, em linhas gerais, o sistema completo desenvolvido e discute os avanços científicos pretendidos pela linha de pesquisa deste trabalho;
- *Correspondência entre pontos salientes de duas imagens* (Capítulo 4): Apresenta uma metodologia para a correspondência de características visuais entre pares de imagens. Esta metodologia analisa a consistência geométrica das correspondências, provendo resultados com uma quantidade significativamente menor de falsos positivos e capaz de lidar com os efeitos de perspectiva causados por objetos em profundidades distintas. Esta é a primeira contribuição desta tese;
- *Estimação de estados* (Capítulo 5): Apresenta a metodologia para a estimação do estado do sistema: a geometria parcialmente conhecida do objeto e a pose da câmera. O processo de estimação é multi-hipotético e é conduzido de modo a detectar e eliminar resultados espúrios antes que estes contaminem

o processo recursivo de estimação. Esta constitui a segunda contribuição desta tese;

- *Planejamento* (Capítulo 6): Apresenta uma metodologia inédita para a avaliação de poses futuras para a câmera, com base no conhecimento parcial da geometria do objeto até o momento. O planejamento também avalia continuamente a execução da trajetória planejada e determina se esta deve ser abandonada e substituída por um novo planejamento, face às novas informações coletadas. Esta metodologia é a terceira contribuição desta tese.
- *Experimentos* (Capítulo 7): Apresenta os experimentos simulados e reais que foram executados a fim de validar a metodologia proposta e discute os resultados obtidos;
- *Conclusões e trabalhos futuros* (Capítulo 8): Discute os avanços obtidos com a tese e propõe extensões que podem ser adotadas por outros trabalhos na mesma linha de pesquisa.

Algumas informações adicionais que não constituem contribuições científicas deste trabalho são apresentadas em alguns apêndices, para referência aos leitores:

- *Filtros de Kalman discretos* (Apêndice A): Provê informações de referência para o Filtro de Kalman linear e as variações não lineares (*Extended Kalman Filter* e *Unscented Kalman Filter*) que são citadas e utilizadas neste trabalho;
- *Probabilidades e o logaritmo da razão de chances* (Apêndice B): Apresenta a forma de representação de probabilidades por meio do logaritmo da razão de chances e discute as vantagens de sua adoção no escopo deste trabalho.

# Capítulo 2

## Trabalhos relacionados

CONFORME A ARGUMENTAÇÃO APRESENTADA no [Capítulo 1](#), o presente trabalho de reconstrução geométrica de cenas possui uma relação forte com as pesquisas desenvolvidas na área de [SLAM](#). Nesse contexto, este capítulo tem por objetivo apresentar uma revisão da literatura científica relacionada ao problema de [SLAM](#). Em particular, duas grandes classes do problema são de interesse para o tema abordado:

- [Planejamento, Localização e Mapeamento Simultâneos](#), ou [SPLAM](#);
- [Monocular SLAM](#), ou [MonoSLAM](#).

Este capítulo é apresentado em três grandes partes. A primeira parte ([Seção 2.1](#)) apresenta e formaliza o [Mapeamento Estocástico](#), uma ferramenta essencial para o desenvolvimento deste trabalho e para a compreensão das diversas pesquisas. A segunda parte apresenta e discute as pesquisas registradas na literatura que foram realizadas acerca dos subproblemas que compõem este trabalho, relacionados a seguir:

- [SLAM](#) com planejamento de movimento do veículo ([Seção 2.2](#)); e
- [SLAM](#) por meio de câmeras ([Seção 2.3](#)).

Finalmente, a [Seção 2.4](#) posiciona o trabalho proposto em relação ao estado-da-arte apresentado na revisão bibliográfica.

### 2.1 Mapeamento estocástico

O [Mapeamento Estocástico](#) (*Stochastic Mapping*, ou [SM](#)) é uma técnica proposta por [Smith et al. \[1990\]](#) que serviu como fundamento teórico para o desenvolvimento

de técnicas de **SLAM** baseadas em observação de características distintivas da cena. O **SM** parte do princípio de que, a partir dos dados fornecidos pelos sensores, é possível identificar, de maneira confiável e repetitiva, certos pontos estacionários da cena (chamados de *features* ou *marcos*). A reobservação desses marcos ao longo do tempo (de diferentes pontos de vista, enquanto o veículo se movimenta) é o fundamento para duas tarefas complementares: (i) localizar o veículo e (ii) mapear os marcos, ou seja, construir uma representação espacial de suas localizações.

O **SM** possui o mérito de tratar essas informações de maneira estocástica, ou seja, os dados são representados e calculados sob a óptica probabilística. Mais importante, o trabalho provou que as incertezas associadas à localização e ao mapeamento são necessariamente interdependentes, já que os erros de estimação da localização se propagam pelo mapa estimado [Leonard & Durrant-Whyte, 1991], e portanto devem ser tratadas como uma unidade.

O **SM** trata o **SLAM** como um problema de estimação recursiva, onde o estado do sistema engloba a localização do veículo e de todo o conjunto de  $M$  marcos observados até o momento.<sup>a</sup> Assim, em um instante de tempo  $t$  o estado do sistema,  $\mathbf{x}_t$ , compreende a estimativa corrente (isto é, baseada em todas as informações disponíveis até o instante  $t$ ) da localização do robô, representada pelo vetor  $\mathbf{r}_t$ , e a da localização de cada um dos  $M$  marcos, representada pelos vetores  $\mathbf{m}_t^1 \dots \mathbf{m}_t^{M_t}$ :

$$\mathbf{x}_t \triangleq \begin{bmatrix} \mathbf{r}_t \\ \mathbf{m}_t^1 \\ \vdots \\ \mathbf{m}_t^{M_t} \end{bmatrix}. \quad (2.1)$$

No contexto do tratamento estocástico do problema, todas as informações são representadas pelos dois primeiros momentos de suas **Funções Densidade de**

---

<sup>a</sup>O estado do sistema inclui *todos* os marcos, não apenas as que são atualmente observáveis pelos sensores. Portanto,  $M$  tende a crescer com o tempo (à medida que novos marcos são observados). O processo também pode incluir critérios para perceber que determinados marcos deixaram de existir (não podem mais ser observados) e que devem ser retirados do vetor de estado, causando decréscimo de  $M$ .

Probabilidade (*Probability Density Functions*, ou PDFs), a média e a covariância:

$$\hat{\mathbf{x}}_t = \begin{bmatrix} \hat{\mathbf{r}}_t \\ \hat{\mathbf{m}}_t^1 \\ \vdots \\ \hat{\mathbf{m}}_t^{M_t} \end{bmatrix} \quad \text{e} \quad (2.2a)$$

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{P}_t^{r;1} & \dots & \mathbf{P}_t^{r;M_t} \\ \mathbf{P}_t^{r;1} & \mathbf{M}_t^{1;1} & \dots & \mathbf{M}_t^{1;M_t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_t^{r;M_t} & \mathbf{M}_t^{M_t;1} & \dots & \mathbf{M}_t^{M_t;M_t} \end{bmatrix}, \quad (2.2b)$$

onde  $\mathbf{R}_t$  é a matriz de covariância (incerteza) da localização do veículo;  $\mathbf{M}_t^{i;j}$  é a covariância cruzada entre a posição dos marcos  $i$  e  $j$ ; e  $\mathbf{P}_t^{r;i}$  é a covariância cruzada entre a localização do veículo e a do marco  $i$ .

### 2.1.1 Mapeamento Estocástico e os filtros de Kalman

A representação da PDF do sistema por meio de seus dois primeiros momentos (Eqs. (2.2a)–(2.2b)) é particularmente conveniente para a abordagem de soluções para o SM baseadas nos filtros de Kalman, já que a média e a covariância descrevem completamente uma PDF gaussiana (Eqs. (A.5)–(A.6)) e os erros de processo e de observação são em geral considerados brancos, gaussianos, aditivos e de média zero, obedecendo às restrições das Eqs. (A.3)–(A.4).

A principal limitação da aplicação de filtros de Kalman para a solução do SM trata da otimalidade do estimador. Mesmo em face das simplificações descritas anteriormente, no problema geral de SLAM tanto o modelo de transição quanto o de observação não são lineares, o que desqualifica o *Filtro de Kalman Linear* (*Linear Kalman Filter*, ou LKF)<sup>b</sup> como suporte à solução do SM. As variações não lineares dos filtros de Kalman não são ótimos; no entanto, eles têm sido largamente usados com elevado grau de sucesso [Durrant-Whyte & Bailey, 2006]. De fato, o próprio trabalho original de Smith et al. [1990] sugere a adoção do *Filtro de Kalman Estendido* (*Extended Kalman Filter*, ou EKF) para sistemas não lineares.

Embora o *Unscented Kalman Filter* (UKF) seja muitas vezes apontado como mais preciso [Thrun et al., 2005] e mais fácil de implementar [Chekhlov et al.,

<sup>b</sup>Em geral, a literatura se refere a *Filtro de Kalman-Schmidt* ou simplesmente *Filtro de Kalman*. Neste trabalho, adotamos a designação “*Filtro de Kalman Linear*” para esse filtro, em contraposição ao termo “KF” que se refere genericamente à classe de filtros de Kalman, incluindo as proposições não lineares lançadas posteriormente.

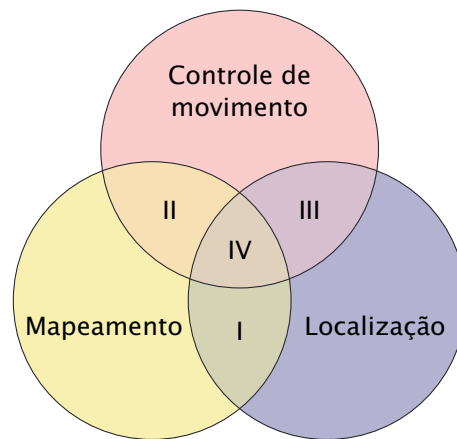
2007], o EKF ainda é o método preferencial nas pesquisas correntes sobre SLAM. Isso se deve, em parte, à simples cronologia da publicação desses filtros: o UKF foi descrito quase três décadas após o EKF, quando este método já tinha se estabelecido como padrão para as tarefas de localização e SLAM com base em modelos unimodais de distribuições de probabilidades de estados. Nos últimos anos, a comunidade científica tem dedicado atenção aos casos em que o EKF produz resultados inconsistentes (“superconfiantes”) [Julier & Uhlmann, 2001; Castellanos et al., 2004; Martinell et al., 2005; Bailey et al., 2006; Frese, 2006; Huang & Dissanayake, 2006, 2007; Castellanos et al., 2007], sinalizando que outras alternativas devem ser estudadas e aplicadas conforme o caso.

## 2.2 Planejamento, Localização e Mapeamento Simultâneos (SPLAM)

O desenvolvimento de técnicas de SLAM prosseguiu durante muito tempo dissociado de quaisquer estratégias exploratórias. A maioria dos trabalhos na área considera implicitamente que a trajetória dos veículos é determinada por outros meios: os veículos são guiados por humanos ou por métodos simples, como caminhada aleatória ou seguimento de paredes. As abordagens clássicas de SLAM são passivas no sentido de que elas apenas processam os dados do sensor, não influenciando o movimento do robô [Stachniss et al., 2004].

No entanto, tanto o mapeamento quanto a localização são assuntos de interesse para a grande área de pesquisas chamada de Robótica Exploratória. Segundo Makarenko et al. [2002], essa área é composta por três linhas principais (Figura 2.1): *mapeamento, localização e controle de movimento* (ou *controle de trajetória*), sendo esta última responsável pela tomada de decisões sobre a trajetória que deve ser executada pelo veículo para otimizar uma função-objetivo, como maximizar o ganho de informação ou minimizar o tempo gasto ou a distância percorrida. A interseção entre as três áreas constitui o SPLAM, também chamado de *Adaptive Concurrent Mapping and Localization* [Feder et al., 1999], *Integrated Exploration* [Makarenko et al., 2002] ou *Active SLAM* [Leung et al., 2006a].

A dificuldade da integração do aspecto exploratório ao SLAM deve-se principalmente ao fato de que não é possível planejar a trajetória com antecedência, já que o próprio mapa do ambiente é desconhecido [Huang et al., 2005]. Portanto, ao contrário das técnicas tradicionais de exploração clássica, o planejamento das ações em SPLAM não pretende buscar uma solução eficiente para um horizonte



**Figura 2.1.** Os campos da robótica exploratória e suas interseções: (I) SLAM, (II) exploração clássica, (III) localização ativa e (IV) SPLAM. (*Adaptado de Makarenko et al. [2002]*)

extenso: o objetivo é estabelecer as atuações a serem aplicadas imediatamente, por meio de técnicas de baixo custo computacional e que sejam adaptativas (capazes de incorporar as informações constantemente coletadas pelos sensores).

### 2.2.1 Estratégias para planejamento restrito à próxima ação

As abordagens pioneiras para o planejamento em **SPLAM** baseiam-se em escolher a próxima ação de movimento a ser executada pelo veículo de modo a maximizar alguma função-objetivo do tipo:

$$\mathbf{u}_{t+1} = \arg \max_{\mathbf{u}} f(\cdot), \quad (2.3)$$

onde  $\mathbf{u}_{t+1}$  representa o controle a ser aplicado ao robô no próximo instante e os argumentos da função-objetivo  $f(\cdot)$  dependem das proposições de cada trabalho publicado sobre o assunto. Se somente o **SM** for estimado, a função-objetivo depende apenas dos parâmetros da **PDF** do estado do sistema: essa é a base dos primeiros trabalhos de planejamento da ação seguinte. Posteriormente outras pesquisas foram feitas para analisar o ganho da manutenção paralela de um **OGM**, como será visto adiante.

O primeiro trabalho a descrever uma metodologia para o **SPLAM** foi descrito por **Feder et al. [1999]**, que apresenta um arcabouço estocástico para estimar o movimento a ser aplicado ao veículo (e aos sensores, se esses forem móveis) com maior probabilidade de maximizar a informação disponível sobre o sistema. Uma maneira de representar essa medida de informação é a partir da matriz de

informação, correspondente ao inverso da matriz de covariância,  $\mathbf{X}_{t+1}$ , de onde:

$$\mathbf{u}_{t+1} = \arg \max_{\mathbf{u}} f(\mathbf{X}_{t+1}^{-1}). \quad (2.4)$$

Como não existe um conceito geral para “maximizar uma matriz”, os autores apresentam um critério baseado no determinante da matriz para reduzir a incerteza das coordenadas dos marcos. Embora as considerações estocásticas sejam teóricas e genericamente aplicáveis a qualquer estimador de estados e a qualquer dimensionalidade do sistema, os autores desenvolveram o trabalho apenas para a estimação por EKF e em ambientes bidimensionais (posição e orientação). Por se restringir ao planejamento apenas do próximo passo de ação, a Eq. (2.4) representa uma instância de *single-step look-ahead strategy* [Huang et al., 2005]. Um dos efeitos indesejados na realização dessa estratégia é o aspecto errático do controle ao longo do tempo, evidenciado nos resultados experimentais apresentados no trabalho de Feder et al.

Um aspecto ainda mais crítico é que a proposição apresentada na Eq. (2.4) não privilegia a exploração de áreas desconhecidas: apenas tende a melhorar a qualidade da localização dos marcos já observados. Essa estratégia foi contestada por Bourgault et al. [2002], que propuseram uma nova função-objetivo. Enquanto Feder et al. buscam a redução geral da incerteza do estado do sistema (Eq. (2.4)), Bourgault et al. procuram maximizar uma função que engloba dois objetivos distintos e contraditórios:

$$\mathbf{u}_{t+1} = \arg \max_{\mathbf{u}} (w_I U_I + w_L U_L), \quad (2.5)$$

onde  $U_I$  é a *utilidade do ganho de informação* (que privilegia a exploração de novas áreas),  $U_L$  é a *utilidade da localização* (que privilegia a qualidade da localização do veículo) e  $w_*$  são pesos arbitrados de ponderação das funções de utilidade. Makarenko et al. [2002] incluem na função-objetivo um terceiro termo: a *utilidade da navegação*,  $U_N$ , que penaliza trajetórias longas. Assim, a função-objetivo é representada por:

$$\mathbf{u}_{t+1} = \arg \max_{\mathbf{u}} (w_I U_I + w_L U_L + w_N U_N). \quad (2.6)$$

Um mérito do trabalho de Makarenko et al. é levar explicitamente em consideração a otimização do custo de navegação, importante em várias aplicações práticas. Em ambos os trabalhos, as funções-objetivo são calculadas com base em um OGM, que



deve ser mantido paralelamente à estimativa de estados do sistema. Em princípio, essa abordagem limita o algoritmo a ambientes internos e bidimensionais, já que o OGM não escala bem para cenários externos ou tridimensionais.

### 2.2.2 Estratégias para planejamento de um conjunto de ações

Alguns pesquisadores criticam a estratégia de planejamento restrita a uma única ação, argumentando que o planejamento em horizontes maiores traz resultados melhores do que os métodos gulosos baseados na Eq. (2.3). Assim, a proposição original foi modificada para o planejamento de um conjunto de passos seguintes, ou:

$$\{\mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+N}\} = \arg \max_{\mathbf{u}} f(\cdot), \quad (2.7)$$

onde  $N$  é o horizonte de predição. Como não é possível prever o ganho de informação pelas observações futuras ou mudanças da dimensão do estado do sistema (observação de novos marcos ou descarte de marcos não observáveis), uma suposição recorrente é que não haverá ganho de informação e que o estado permanecerá com a mesma dimensão.

Huang et al. [2005] lançam as bases matemáticas que provam que o uso de *Model Predictive Control* (MPC) gera resultados melhores do que a aplicação recursiva do planejamento guloso (*one-step*). A função-objetivo procura minimizar o traço da matriz final de covariância:

$$\{\mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+N}\} = \arg \min_{\mathbf{u}} \text{tr}(\mathbf{X}_{t+N}). \quad (2.8)$$

(O uso do traço da matriz de covariância como base de cálculo é discutido na Seção 2.2.3.) Embora o planejamento seja feito para os  $N$  passos seguintes, a trajetória é replanejada a cada passo, à medida que novos dados são incorporados à estimação do estado do sistema. O trabalho é posteriormente estendido por Leung et al. [2006b], que apresentam metodologias para otimizar o custo do MPC.

No entanto, o critério de otimização apresentado na Eq. (2.8) possui a mesma limitação inerente ao trabalho de Feder et al. [1999]: a exploração de áreas desconhecidas não é privilegiada. Uma solução para este problema foi apresentada por Leung et al. [2006a], onde uma máquina de estados determina o comportamento do veículo entre três alternativas: *exploração* (navegação para áreas inexploradas), *melhoria da localização* (navegação para áreas com marcos bem localizados) e

*melhoria do mapa* (navegação para áreas com marcos mal localizados). Para não modificar a implementação original do MPC, os autores introduziram o conceito de “atrator”, que consiste em um falso marco com alta covariância posicionado em locais estratégicos. A incerteza associada ao atrator eventualmente leva o MPC a traçar uma trajetória às áreas de interesse. Uma implementação prática da metodologia é apresentada em Leung et al. [2008]. Infelizmente esse trabalho é bastante restrito: o ambiente em questão é interno (*indoor*), bidimensional e fortemente estruturado (considera-se que a cena é suficientemente simples para que possa ser representada por um conjunto de segmentos de reta). Os autores não apresentam qualquer análise de custo computacional, portanto não é possível perceber se a metodologia é escalável para ambientes tridimensionais.

### 2.2.3 Análise teórica do ganho de informação em SPLAM

A exploração em SLAM já foi tratada sob um ponto de vista mais teórico, pela análise probabilística do ganho de informação esperado com diferentes estratégias de otimização. O artigo de Sim & Roy [2005] foi o ponto de partida para alguns trabalhos nessa área — que, por tratarem de um problema mais abstrato, relaxam o problema do SLAM em vários aspectos: consideram que o número de marcos é conhecido *a priori*, a sua observação não apresenta ambiguidades e o ambiente é interno, bidimensional e não possui obstáculos, permitindo o deslocamento livre dos veículos e a observação dos marcos sem oclusão.

Apesar dessas restrições, os autores chegaram a algumas conclusões importantes. Em Sim & Roy [2005], provam que o critério de entropia relativa (usado em alguns algoritmos de Visual SLAM, como no trabalho de Bailey [2003], que será discutido posteriormente na Seção 2.3) é essencialmente falho. Esse critério equivale à estratégia chamada de *D-Otimalidade* [Montgomery, 2000], que busca reduzir o determinante da matriz de covariância do processo:

$$\{\mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+N}\} = \arg \min[\det(\mathbf{X}_t)] = \arg \min \prod_i \lambda_i, \quad (2.9)$$

onde  $\lambda_i$  são os autovalores de  $\mathbf{X}_t$ . Porém, o problema se torna malcondicionado se a incerteza da localização de um único marco tender a zero, tornando  $\mathbf{X}_t$  singular. A alternativa proposta é a estratégia da *A-Otimalidade* [Montgomery, 2000], que

procura reduzir o traço de  $\mathbf{X}_t$ , ou:

$$\{\mathbf{u}_{t+1}, \dots, \mathbf{u}_{t+N}\} = \arg \min[\text{tr}(\mathbf{X}_t)] = \arg \min \sum_i \lambda_i. \quad (2.10)$$

Ainda que a estratégia seja considerada pobre quando o estado é composto por valores de unidades diferentes (medidas espaciais  $\times$  angulares), na prática os autores demonstraram que é possível chegar a melhores resultados.

Ainda no campo teórico, [Sim \[2005b\]](#) demonstra uma contradição inerente ao uso do **EKF** em estratégias exploratórias no campo de *bearing-only SLAM*: Quando um veículo se encontra próximo a um marco, maior é o ganho esperado de informação; entretanto, nesse caso a função de predição se comporta de modo altamente não linear, tornando o **EKF** um estimador ruim por causa da linearização adotada no algoritmo (Eq. (A.12b)). Essa condição atinge os trabalhos de [Bourgault et al. \[2002\]](#) e [Stachniss et al. \[2004\]](#), entre outros. Para evitar instabilidade na estimação do estado, [Sim \[2005a\]](#) sugere ignorar a observação de marcos que estejam mais próximos do que um limite mínimo de distância e apresenta resultados que comprovam que os resultados são mais estáveis quando as observações muito próximas são descartadas.

## 2.3 SLAM visual

A formulação teórica do **SM** é propositalmente vaga a respeito de como os marcos são descritos ou percebidos. Mesmo quando a formulação é escrita em termos dos filtros de Kalman, o modelo de observação (Eq. (A.2)) deve apenas respeitar as imposições quanto à distribuição dos ruídos (Eq. (A.4)): brancos, aditivos, gaussianos e de média zero. É interessante notar que o artigo de [Smith et al. \[1990\]](#), responsável pela definição do **SM**, foi o resultado da convergência de trabalhos com sensores distintos: [Ayache & Faugeras \[1988\]](#), sobre navegação visual, e [Chatila & Laumond \[1985\]](#) e [Crowley \[1989\]](#), dedicados a técnicas de navegação baseada em sonar [[Durrant-Whyte & Bailey, 2006](#)].

De fato, soluções para o **SM** em geral (e **SLAM** em particular) têm sido desenvolvidas com o uso das mais variadas classes de sensores. Os sonares foram usados há mais tempo; porém, por causa das desvantagens inerentes desse sensor (campo de vista largo, interferência cruzada entre várias unidades (*crosstalk*), etc.), seu uso tem sido atualmente restrito a ambientes subaquáticos, pelo fato de outras modalidades sensoriais baseados em luz (**Sensores de Distância a Laser** (*Laser Range*

*Finders*, ou LRFs), câmeras, etc.) serem limitados quando utilizados em meios participativos. Sensores de intensidade (câmeras analógicas e digitais) também são usadas há décadas, mas somente nos últimos anos se dispõe de capacidade computacional para o processamento em tempo real da quantidade de dados gerados. Recentemente observa-se a popularização dos LRFs, motivada pela queda do preço desses dispositivos. No entanto, esses sensores são limitados a varreduras em um plano, restringindo sua aplicabilidade em problemas que consideram ambientes tridimensionais. Embora tenham sido adaptados para a varredura 3D pelo uso de bases rotativas [Lingemann et al., 2004; Nüchter et al., 2007] ou espelhos [Ryde & Hu, 2007], a dificuldade do controle das partes móveis e o tempo elevado de uma varredura completa do ambiente são apontados como fatores limitantes para a dinâmica dos veículos e do ambiente. Há modelos específicos de 3DLRFs (por exemplo, Velodyne HDL-64E [Velodyne Lidar, Inc., 2009]), mas seu custo atual (na ordem de centenas de milhares de dólares) impede a sua adoção em larga escala. Há outras soluções que estão fora do escopo deste trabalho — como as baseadas em marcos que emitem sinais de rádio (dispositivos RFID ou nós sensores, por exemplo) —, pois requerem estruturação prévia do ambiente.

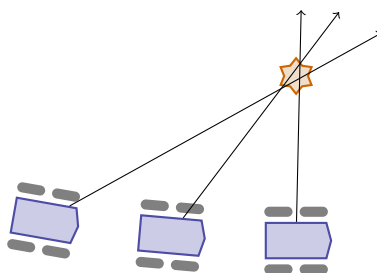
Diversos fatores motivam a adoção de câmeras no tratamento de SLAM, incluindo o preço, a portabilidade, o baixo consumo de energia quando comparado a outras classes de sensores, a possibilidade de uso em ambientes subaquáticos e extraterrenos e a frequência de amostragem relativamente alta. Além disso, são sensores naturalmente adequados para o SLAM tridimensional. No entanto, as câmeras convencionais não são capazes de fornecer a profundidade dos pontos da cena. Embora o eixo de projeção de cada ponto amostrado possa ser recuperado com precisão, a posição tridimensional do marco somente pode ser estimada pela triangulação de sua observação sob mais de um ponto de vista [Bailey, 2003] (Figura 2.2). A incapacidade de um sensor perceber a profundidade dos objetos o classifica como um *bearing-only sensor* (sensor apenas de orientação); por extensão, o SLAM executado com tais sensores é conhecido como *Bearing-Only SLAM* (BO-SLAM), ou “SLAM baseado em orientação”, em uma tradução livre.<sup>c</sup>

Os problemas inerentes ao BO-SLAM serão discutidos na subseção seguinte. No entanto, é importante ressaltar que há outras abordagens de SLAM com câmeras que evitam a necessidade de observação sequencial, a saber:

1. *Uso de estruturas estéreo*: Nesse caso, a triangulação é resolvida por algorit-

---

<sup>c</sup>O *bearing-only SLAM* é parte de uma classe de problemas chamada de “SLAM parcialmente observável”, na qual se inclui também o *range-only SLAM*, que se baseia geralmente no uso de sonares.



**Figura 2.2.** Problema da observação parcial em *bearing-only SLAM*. Nessa modalidade de percepção, a localização do marco não pode ser estimada com uma única observação, requerendo a fusão da observação a partir de pontos de vista diferentes. (Adaptado de *Bailey & Durrant-Whyte [2006]*)

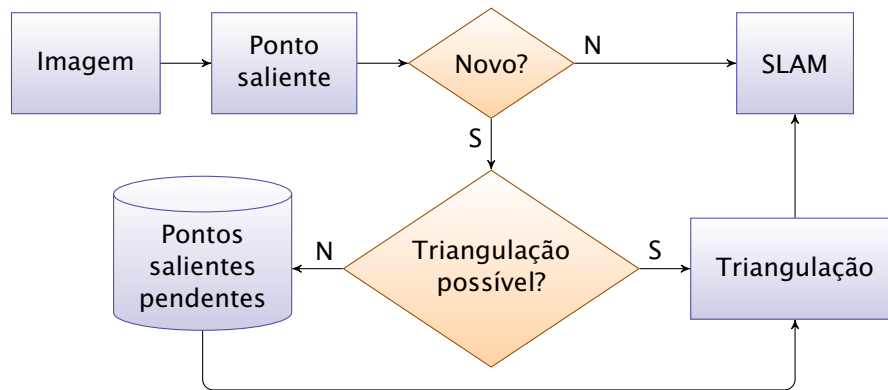
mos clássicos de visão estéreo, gerando uma estimativa (com a respectiva incerteza) da poses tridimensionais dos marcos da cena relativas ao veículo. Essa proposição já foi usada tanto em ambientes internos [Se et al., 2002; Davison & Murray, 2002] como externos [Jung & Lacroix, 2003; Hygounenc et al., 2004]. Sistemas estéreo, no entanto, requerem calibração prévia, podem gerar resultados espúrios na presença de vibração e em geral dificultam o uso de recursos de *zoom* das câmeras;

2. *Disposição de alvos de tamanho conhecido na cena:* Isso torna desnecessária a triangulação para a determinação da distância do alvo até a câmera, pois essa informação pode ser calculada a partir do tamanho, em pixels, da imagem do alvo. Nesse campo, destaca-se o trabalho de Kim & Sukkarieh [2003], que combinam visão e sensores inerciais para a realização de SLAM em veículo aéreo com alta precisão. Entretanto, o uso de alvos artificiais não se estende a ambientes não estruturados em geral.

### 2.3.1 Os problemas de *Bearing-only SLAM*

Pela definição formal de SM, a qualquer marco representado pelo vetor de estados do sistema está associada uma incerteza (finita). No entanto, marcos percebidos por um sensor do tipo *bearing-only* possuem incerteza infinita na direção do eixo de profundidade (Figura 2.2); portanto, a inclusão dos marcos no vetor de estados deve ser adiada até que uma posterior observação do mesmo marco permita a triangulação. A Figura 2.3 ilustra esse processo, conhecido como *inicialização atrasada* (*delayed initialization*).

Embora conceitualmente simples, o processo requer que alguns subproblemas sejam resolvidos:



**Figura 2.3.** Abstração do processo de inicialização atrasada em *bearing-only SLAM*.

1. Como localizar pontos salientes nos dados fornecidos pelo sensor?
2. Como determinar que pontos salientes identificados em duas observações diferentes correspondem ao mesmo marco da cena?
3. Quando a triangulação deve ser executada?
4. Qual a melhor pose dos marcos e do veículo (no instante de cada observação) que melhor explica os dados de entrada?

As duas primeiras questões tratam da *identificação e correspondência* dos pontos salientes. A terceira questão é relevante porque a triangulação feita de dois pontos de vista muito próximos pode ser criticamente malcondicionada [Bailey, 2003]. A quarta questão é o cerne de **BO-SLAM**, pois objetiva recuperar tanto a pose do veículo (localização) quanto a dos marcos da cena (mapeamento).

O interesse particular do presente trabalho está em técnicas de **BO-SLAM** baseadas em uma única câmera, o que é conhecido como *SLAM visual, single-camera SLAM* ou *Monocular SLAM* (MonoSLAM). Dessa forma, os sensores fornecem imagens bidimensionais de intensidade. A **Subseção 2.3.2** apresenta a revisão bibliográfica das técnicas de identificação e correspondência de pontos salientes em imagens. Em seguida, a **Subseção 2.3.3** trata dos trabalhos sobre *bearing-only SLAM* e *Monocular SLAM* (MonoSLAM), que buscam a resposta para as duas últimas questões apresentadas.

### 2.3.2 Identificação e correspondência de pontos salientes em imagens

Até recentemente, os algoritmos existentes de identificação de pontos salientes em imagens — como quinas [Harris & Stephens, 1988; Shi & Tomasi, 1994; Wang

& Brady, 1995; Smith & Brady, 1997; Trajkovic & Hedley, 1998] ou de pontos e regiões salientes [Tuytelaars & Gool, 1999, 2000; Matas et al., 2002; Mikolajczyk & Schmid, 2002; Schaffalitzky & Zisserman, 2003; Kadir et al., 2004] eram capazes de retornar apenas as coordenadas bidimensionais (no espaço da imagem) dos marcos encontrados. Entretanto, a correspondência entre pares desses pontos de duas imagens só é tratável se a posição relativa da câmera entre as observações for previamente conhecida, como nas montagens tradicionais de visão estéreo. No caso geral de *SLAM* — em que o deslocamento do sensor é apenas estimado ou totalmente desconhecido —, o espaço de busca de correspondência é quadrático e a falta de robustez pode facilmente gerar soluções irreais.

A última década viu surgir vários trabalhos sobre *descritores de características*. Trata-se de técnicas para avaliar um *descriptor* — um vetor, em geral de tamanho fixo e previamente estabelecido — que representa a informação da vizinhança de cada ponto de interesse identificado na imagem. Dados dois descritores,  $\mathbf{d}_i$  e  $\mathbf{d}_j$ , associados respectivamente aos pontos salientes  $\mathbf{p}_i$  e  $\mathbf{p}_j$  de duas imagens distintas, uma função de distância entre dois descritores<sup>d</sup>,  $\text{dist}(\mathbf{d}_i, \mathbf{d}_j)$ , avalia a similaridade entre os descritores: valores baixos de  $\text{dist}(\mathbf{d}_i, \mathbf{d}_j)$  indicam alta probabilidade dos pontos  $\mathbf{p}_i$  e  $\mathbf{p}_j$  serem projeções do mesmo marco da cena.

Cada metodologia para a descrição de características é invariante ou robusto a um determinado conjunto de transformações. Geralmente os métodos são invariantes a translações, escalas isotrópicas e rotações. As demais transformações, como cisalhamento, escala anisotrópica e efeitos de perspectiva, são classificadas como “efeitos de segunda ordem” [Bay et al., 2006] e quase sempre deixadas de lado. A modelagem de tais transformações é bem mais complexa e só apresenta vantagens quando se procura robustez em mudanças amplas do ponto de vista [Lowe, 2004].

Os descritores não são um conceito recente: o assunto já vem sendo pesquisado há mais de duas décadas (Koenderink & van Doorn [1987]; Freeman & Adelson [1991]; Gool et al. [1996], entre outros). No entanto, os trabalhos mais antigos não são invariantes a escala, o que é uma característica essencial para a correspondência de marcos observados a distâncias diferentes.

O primeiro algoritmo especificamente voltado para a invariância de escala foi publicado por Lowe [1999] e é chamado de *SIFT*. Para cada ponto de interesse (localizado por um dos detectores citados anteriormente), os seguintes passos são executados: (i) busca-se uma escala de vizinhança que maximiza a informação;

---

<sup>d</sup>Nos algoritmos para os quais os descritores possuem tamanho fixo, em geral adota-se a distância euclidiana ou de Mahalanobis.

(ii) descartam-se pontos em regiões de pouco contraste ou localizados em bordas; (iii) determinam-se as orientações alinhadas com os picos de histogramas angulares de gradientes de intensidade; e (iv) calcula-se o descritor, composto por (novos) histogramas angulares de 8 posições para cada uma das  $4 \times 4$  regiões vizinhas ao ponto de interesse (gerando, portanto, um vetor de  $8 \cdot 4 \cdot 4 = 128$  elementos). O passo (i) garante a invariância à escala e o passo (iii) garante a invariância à rotação. O SIFT foi posteriormente melhorado pelo autor [Lowe, 2004] introduzindo um passo intermediário entre (i) e (ii), onde as coordenadas do ponto de interesse são ajustadas, o que melhora a estabilidade e a correspondência do algoritmo.

O SIFT foi um trabalho marcante e serviu de base para vários outros algoritmos publicados posteriormente, que aproveitam boa parte do raciocínio original e em geral sugerem alternativas para o cálculo do descritor (passo (iv)). Nessa linha, o primeiro algoritmo relevante foi apresentado por Ke & Sukthankar [2004], chamado de *Principal Components Analysis SIFT* (PCA-SIFT). Toma-se uma vizinhança de  $41 \times 41$  pixels (possivelmente escalada e rotacionada pelos passos anteriores) em torno do ponto de interesse e calculam-se os gradientes horizontal e vertical, gerando um vetor de  $2 \cdot (41 - 2)^2 = 3042$  elementos. Esse vetor é submetido a *Análise em Componentes Principais* (*Principal Components Analysis*, ou PCA) [Jolliffe, 2002] para gerar uma representação mais compacta, de  $n$  elementos. Os autores avaliam várias alternativas para a seleção de  $n$ : concluem que  $n = 36$  gera descritores mais confiáveis, embora sugeriram a adoção de  $n = 20$  por questões de performance (o tempo médio gasto no cálculo de descritores com  $n = 20$  é da mesma ordem dos descritores do SIFT).

Uma análise extensiva dos métodos anteriores (com exceção do PCA-SIFT, que foi publicado posteriormente) é feita por Mikolajczyk & Schmid [2003], estabelecendo uma comparação da eficácia de cada um deles diante de diversas transformações geométricas e fotométricas: rotação, mudança de escala (*zoom*), transformações afins (mudança do ponto de vista) e alteração das condições de iluminação da cena. O trabalho chega à conclusão de que o SIFT é sistematicamente mais eficaz do que os demais métodos, independentemente do detector de pontos salientes utilizado. Posteriormente [Mikolajczyk & Schmid, 2005] a avaliação é estendida, incluindo o PCA-SIFT na comparação; porém, a avaliação conclui que este método não é mais robusto do que o SIFT.

Ainda no mesmo trabalho, os autores definem um algoritmo chamado de *Gradient Location and Orientation Histogram* (GLOH). Diferentemente do SIFT, este algoritmo divide a vizinhança de cada ponto em regiões log-polares, com 3 divisões no sentido radial e 8 divisões angulares (exceto na região central), perfazendo



$8 \cdot 2 + 1 = 17$  regiões. Um histograma angular de gradientes de intensidade (de 16 posições) é gerado para cada região, o que gera um vetor de  $17 \cdot 16 = 272$  elementos. Para o descritor final, esse vetor é analisado por PCA para chegar a um vetor de 128 dimensões. Os autores demonstram que o novo algoritmo é mais distintivo do que o SIFT; no entanto, o GLOH requer uma carga computacional maior. Em nenhum dos trabalhos os autores apresentam uma comparação do tempo de execução dos diversos algoritmos analisados.

Apesar das alternativas, o SIFT permaneceu como algoritmo mais usado para a geração de descritores. Isso se deve, em parte, às conclusões dos estudos de Mikolajczyk & Schmid [2003, 2005] e ao fato de que o SIFT, por ser o algoritmo mais antigo entre os invariantes a transformações de escala, tem sido usado por mais tempo e implementado em diversas plataformas.<sup>e</sup> Posteriormente, a prova matemática da invariância à escala foi apresentada por Morel & Yu [2008], cujos resultados explicam por que o SIFT supera todos os demais métodos no que diz respeito à invariância de escala.

Essas vantagens motivaram o desenvolvimento de um novo método por Bay et al. [2006]. Chamado de SURF, o algoritmo possui forte ênfase no custo computacional de execução. Assim como o SIFT, a vizinhança do ponto de interesse é dividida em  $4 \times 4$  regiões. Cada uma é dividida em  $5 \times 5$  subregiões, sobre as quais são calculadas as respostas das *wavelets* de Haar (*Haar wavelets*) na horizontal e na vertical. Essas respostas são somadas, assim como os seus valores absolutos, gerando 4 valores para cada região; portanto, o descritor possui  $4 \cdot 4^2 = 64$  elementos. A escolha da orientação da vizinhança (passo (iii) do SIFT) também é baseada nas respostas das *wavelets* de Haar. Três variações do descritor são apresentadas no mesmo trabalho: uma com menos dimensões (porém de cálculo mais rápido), chamada SURF-36, onde a vizinhança é dividida em apenas  $3 \times 3$  regiões (o descritor possui  $4 \cdot 3^2 = 36$  dimensões); outra, chamada SURF-128, onde as respostas das *wavelets* são separadas pelo seu sinal, gerando um descritor de 128 elementos; e ainda outra, chamada de *Upright SURF* (U-SURF): nesta variante, o passo (iii) (escolha da orientação) é ignorado. Nesse caso, os descritores gerados não são invariantes à rotação, mas podem ser adequados em casos em que não se espera a ocorrência de rotação da câmera em torno do eixo óptico.

---

<sup>e</sup>Há poucos anos o SIFT foi implementado em *Field-Programmable Gate Array* (FPGA) [Se et al., 2004], permitindo a execução do algoritmo em 60 ms, cerca de um décimo do tempo consumido por um PC da época. Mais recentemente, implementações em Unidades de Processamento Gráfico (*Graphics Processor Units*, ou GPUs) [Sinha et al., 2006, 2007] também alcançaram um *speedup* de  $10 \times$  em relação às CPUs contemporâneas, permitindo a execução do algoritmo a 30 quadros por segundo.

Os autores apresentam resultados que demonstram que o SURF é sistematicamente mais rápido e eficaz do que outros algoritmos. No entanto, não foram encontrados estudos comparativos, feitos por fontes independentes, que confrontam o SURF com os demais algoritmos. De modo mais geral, os estudos se concentram principalmente na comparação da confiabilidade e repetitividade dos descritores; não há estudos voltados especificamente para a comparação da performance desses algoritmos, em especial no campo de navegação e/ou mapeamento visual (onde tanto a geração como a comparação de descritores ocorre a cada quadro).

### 2.3.3 *Bearing-only SLAM* e SLAM monocular

O problema de inicialização atrasada (ilustrado na Figura 2.3), embora simples em sua apresentação teórica, apresenta várias dificuldades de implementação prática. Uma dificuldade importante está na forma de armazenar os pontos salientes pendentes: por não terem sido ainda triangulados, tais pontos são descritos no espaço de coordenadas bidimensionais (em pixels), possivelmente de várias imagens adquiridas ao longo do tempo.

Nesse escopo, um dos primeiros trabalhos significativos foi publicado por Leonard & Rikoski [2000]. Os autores não tratam especificamente de imagens de câmeras, ou sequer do problema de BO-SLAM. No entanto, descrevem uma abordagem teórica para a implementação da inicialização atrasada que serviu de ferramenta para vários trabalhos posteriores. A ideia é simples: o vetor de estados do SM mantém o histórico da trajetória do veículo, em vez de estimar somente sua pose corrente. Não é necessário manter a trajetória completa: apenas as poses históricas correspondentes aos pontos pendentes de triangulação. Com isso, a triangulação pode ser adiada indefinidamente, até que algum critério avalie que a triangulação pode ser realizada de maneira robusta e bem condicionada. Os autores estenderam posteriormente o trabalho [Leonard et al., 2002] para detalhar o algoritmo e apresentar experimentos práticos. O problema de correspondência automática de pontos é explicitamente ignorado no trabalho, sendo feito manualmente.

Esses trabalhos não propuseram uma métrica para avaliar se a triangulação é suficientemente bem condicionada para que o marco seja inicializado. Bailey [2003] descreve um critério probabilístico de normalidade da distribuição da PDF de uma triangulação. Dada a incerteza de cada observação, o autor avalia a divergência de Kullback-Leibler (entropia relativa) [Kullback & Leibler, 1951] para decidir se a incerteza da triangulação pode ser razoavelmente aproximada por uma

distribuição normal multivariada. Na falta de uma forma fechada para as equações apresentadas, a avaliação da entropia relativa foi feita por amostragem de Monte Carlo. Os autores notam que essa abordagem é custosa e imprecisa, mesmo para o caso bidimensional; não se sabe se é escalável para o caso tridimensional.

Costa et al. [2004] apresentam um método para inicialização atrasada sem a necessidade de correspondência prévia. Para cada par de imagens, todas as correspondências possíveis são avaliadas, e as mais persistentes são automaticamente consideradas válidas. Não há uma análise probabilística para suportar esse critério. Além disso, o custo do teste de todas as correspondências possíveis é da ordem de  $O(n^2)$ , portanto não escala para um grande número de pontos salientes (os experimentos apresentados no trabalho foram feitos com no máximo 18 marcos).

Outros pesquisadores experimentaram alternativas para conduzir o processo de triangulação. Um paradigma recorrente é o uso de *Soma de Gaussianas* (*Sum of Gaussians*, ou *SoG*). Quando se percebe um novo ponto de interesse (isto é, sem correspondência com algum marco conhecido), várias hipóteses são criadas ao longo de seu eixo de incerteza, de modo que a superposição das PDFs (gaussianas) dessas hipóteses aproximam a PDF (não gaussiana) da incerteza do marco em um intervalo preestabelecido de distância a partir do centro de projeção da câmera. Com a incorporação de novos dados, uma dessas hipóteses é selecionada como a mais plausível e as demais são descartadas.

O grupo de pesquisa de Davison foi um dos pioneiros no uso de SoG [Davison, 2002, 2003; Davison et al., 2007]. Seus trabalhos apresentam a interessante característica de não depender de uma estimativa prévia de deslocamento da câmera (por odometria ou sensores inerciais), mas os critérios usados para avaliar e aceitar triangulações não são descritos. Além disso, os autores sugerem [Davison et al., 2007] que a adaptabilidade para ambientes externos depende da adoção de descritores invariantes como o SIFT.

Os trabalhos de Kwok & Dissanayake [2004] e Kwok et al. [2005] também adotam uma solução baseada em SoG. Os autores usam um Filtro de Soma de Gaussianas (*Gaussian Sum Filter*, ou GSF) [Sorenson & Alspach, 1971; Alspach & Sorenson, 1972], que consiste em manter um banco de EKFs executando em paralelo (um filtro para cada componente da SoG) e um processo independente de avaliação dos resultados de cada EKF para determinar a associação mais plausível. O método não é escalável para ambientes externos, já que, segundo Solà et al. [2005], a manutenção de um grande banco de EKFs possui custo computacional proibitivo.

Solà et al. [2005, 2008] apresentam uma abordagem estocasticamente mais precisa e com custo computacional menor do que o banco de EKFs sugerido por

**Davison.** Em vez de manter vários filtros de Kalman em paralelo, o processo gera várias alternativas possíveis para as coordenadas de cada marco. À medida que as observações são feitas, as hipóteses menos prováveis são eliminadas e as demais são corrigidas, até restar uma única alternativa, quando então considera-se que a triangulação está concluída. O aspecto estocástico é tratado com cuidado, de modo que a informação disponível em cada imagem seja dividida na estimação de cada hipótese, evitando que o sistema realize estimações superconfiantes pela superestimação das observações. Uma implementação prática da metodologia é apresentada em **Lemaire et al. [2005]**; porém, os experimentos são realizados em ambientes internos e o espaço de atuação é planar (bidimensional), de modo que o método não foi avaliado para ambientes externos e tridimensionais em geral.

## 2.4 A relação entre o estado-da-arte e este trabalho

Esta seção tem por objetivo situar o presente trabalho em relação a cada um dos tópicos discutidos nas seções anteriores, identificando as técnicas a serem utilizadas e as inovações propostas.

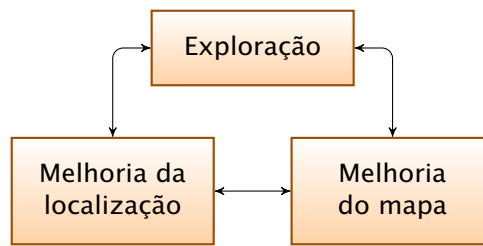
### 2.4.1 Estimador de estados e mapeamento estocástico

Dado o caráter não linear dos modelos de transição de estados em um ambiente tridimensional (6 graus de liberdade) e de observação baseada em transformações projetivas de pontos da cena em imagens, observa-se que o **LKF** não pode ser utilizado neste trabalho. Quanto às alternativas não lineares, será adotado o **UKF**, diante dos argumentos apresentados na **Subseção 2.1.1**: maior estabilidade face às não linearidades do modelo, maior facilidade de implementação, resultados mais consistentes e a mesma complexidade assintótica, quando comparado com o **EKF**.

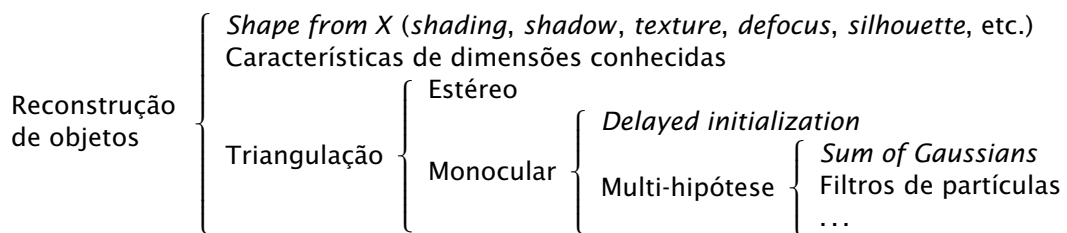
### 2.4.2 SLAM ativo (SPLAM)

No que diz respeito à taxonomia apresentada na **Seção 2.2**, neste trabalho o planejamento visa gerar um conjunto futuro de ações. No entanto, propõe-se uma abordagem distinta dos trabalhos apresentados, baseada no planejamento em dois passos:

1. determina-se a melhor configuração futura para o robô, considerando-se um comportamento baseado na máquina de estados vista em **Leung et al. [2006a]** (alternando entre *melhoria da localização*, *melhoria do mapa* e *exploração*)



**Figura 2.4.** Diagrama de estados para a seleção do objetivo a ser perseguido por um robô.



**Figura 2.5.** Taxonomia das estratégias de reconstrução da geometria de objetos por meio de câmeras.

(Figura 2.4) e uma função de utilidade associada a cada comportamento, desenhada para avaliar as estimativas e incertezas do estado corrente representadas por **SM** no âmbito de ambientes tridimensionais; e

2. a trajetória do robô é traçada e executada, a fim de levar a configuração atual à configuração avaliada no passo anterior. É neste escopo que os vários passos de atuação são gerados (e possivelmente alterados durante a execução da trajetória).

### 2.4.3 SLAM visual

Os trabalhos publicados sobre reconstrução de geometria de objetos por meio de câmeras podem ser divididos em três grandes grupos de estratégias principais para a estimação da profundidade dos pontos (Figura 2.5):

1. A análise de características do espaço de imagem, como gradientes e bordas: Compreende os diversos algoritmos de *shape from X*. Em geral, tais algoritmos recuperam a geometria a menos de um fator de escala, ou seja, a profundidade absoluta dos pontos é justamente a informação que não é estimada. Apenas os algoritmos de *shape from defocus* recuperam a profundidade; porém, requerem a observação da cena do mesmo ponto de vista com

diferentes ajustes do foco da câmera, o que impede a execução do algoritmo com a câmera em movimento (embarcada em um robô móvel, por exemplo);

2. A identificação, na imagem, de características da cena cujas dimensões são previamente conhecidas: Requer a prévia estruturação da cena (a disposição de marcos visuais em locais visíveis) e não é genericamente aplicável se a cena for desconhecida. Portanto, não é adequada ao presente trabalho, que parte do pressuposto de que a cena não é estruturada;
3. A triangulação de uma mesma característica da cena observada em múltiplas imagens: Requer a observação da cena sob pontos de vista diferentes. Esta é a única estratégia que satisfaz todos os requisitos deste trabalho: Permite a recuperação da profundidade absoluta dos objetos e não requer a prévia estruturação da cena.

Neste trabalho, a reconstrução da cena será baseada em estratégias de triangulação de pontos salientes. Conforme discutido anteriormente (Seção 2.3), tais estratégias se dividem em dois grupos principais [Lemaire et al., 2007]: (i) o uso de uma única câmera (abordagem monocular) e (ii) o uso de múltiplas câmeras em montagens estéreo. Em essência, a abordagem estéreo simplifica o problema fundamental da recuperação da profundidade dos pontos observados, enquanto a abordagem monocular prevalece como tema recorrente de pesquisas recentes.

O presente trabalho adotará a linha de **SLAM** monocular, o que configura o problema como uma instância de **BO-SLAM**. A inicialização dos marcos será feita por **SoG**, a exemplo do trabalho de Solà et al. [2005, 2008]. No entanto, diversas adaptações importantes serão propostas para adaptar o método ao problema de **SLAM**, entre elas: a adoção de descritores de características, como **SIFT** e **SURF**, necessária para manter uma boa qualidade no processo de correspondência dos pontos salientes em ambientes externos; uma nova proposição para a determinação da posição das hipóteses da **SoG**, para evitar resultados tendenciosos; e o desacoplamento entre a correção da pose dos marcos tentativos e a estimação das demais variáveis de estado do sistema, para evitar a contaminação da geometria da cena pela informação fornecida pelas falsas hipóteses.

# Capítulo 3

## Metodologia

ESTE CAPÍTULO TEM POR OBJETIVO apresentar a arquitetura adotada para a solução do problema proposto. Os formalismos apresentados neste capítulo são essenciais para a compreensão dos três capítulos seguintes, que tratam do detalhamento matemático dos módulos que constituem a contribuição deste trabalho: a correspondência entre pontos salientes (Capítulo 4); o processo de estimação da geometria da cena e a estimação da configuração atual do robô<sup>a</sup> (SLAM) (Capítulo 5; e o planejamento de configurações futuras com base nas informações disponíveis (Capítulo 6).

Este capítulo está assim organizado: A Seção 3.1 resgata a definição do problema, vista no capítulo introdutório, e detalha de maneira informal a solução proposta neste trabalho; a Seção 3.2 discute as suposições adotadas (acerca da cena, do modelo de iluminação, dos ruídos, etc.); e a Seção 3.3 ilustra e detalha as camadas da arquitetura criada para abordar o problema.

### 3.1 Visão geral

Para iniciar a apresentação da metodologia deste trabalho, a formalização do problema (Definição 7, pág. 12) será transcrita a seguir por conveniência:

**Definição 7 (Problema Principal):** *Dados um objeto de interesse em um ambiente tridimensional e um robô móvel dotado de uma câmera, planejar e realizar uma estratégia de ações para construir uma representação geométrica do objeto de interesse, representada por um conjunto de pontos tridimensionais  $\mathcal{M} = \{\mathbf{m}^1, \dots, \mathbf{m}^M\}$  da superfície do objeto.* □

---

<sup>a</sup>No escopo deste documento, os termos “robô” e “veículo” serão utilizados com o mesmo significado.

Conforme discutido anteriormente, a indisponibilidade de um referencial global de localização para os veículos requer que a pose do veículo seja estimada juntamente com a geometria do objeto, o que configura uma instância do problema de **SLAM**.

O sistema de reconstrução proposto baseia-se em imagens obtidas em intervalos regulares de tempo por uma câmera embarcada em um robô móvel. Neste trabalho não são impostas restrições em relação ao espaço de configurações do robô em um espaço tridimensional ou ao seu possível conjunto de restrições não holonômicas — o que permite a adoção de robôs aéreos, por exemplo. O acoplamento entre a câmera e o robô pode ser articulado, permitindo o uso de unidades *pan-tilt* ou câmeras montadas em braços manipuladores.

Para cada imagem capturada, um conjunto de pontos salientes é detectado utilizando algoritmos como **SIFT** ou **SURF**, bem como as respectivas assinaturas visuais (vetores descritores). Parte-se do princípio de que cada um desses pontos salientes corresponde a algum ponto na superfície de algum objeto da cena<sup>b</sup>. Para evitar o uso do termo ambíguo “ponto”, que pode se referir tanto a uma característica na imagem quanto na cena, as seguintes definições serão usadas:

- Um *ponto saliente* é uma característica visual associada a um par de coordenadas de uma imagem — portanto, sempre descrito em pixels — e a um vetor descritor que representa a assinatura visual do entorno desse ponto. Os algoritmos de detecção aqui utilizados também fornecem algumas informações geométricas: um fator de escala e um ângulo de orientação;
- Um *marco* é um ponto da superfície de um objeto da cena. A observação repetida de um marco pela câmera em poses diferentes é evidenciada pela correspondência repetida dos respectivos pontos salientes detectados em cada imagem.

Em geral, nem todos os pontos salientes detectados em uma imagem correspondem a algum marco. Diversos efeitos na imagem podem produzir pontos salientes sem relação com um ponto na superfície de um objeto. Entre esses efeitos estão ruídos produzidos durante o processo de aquisição de imagem e entidades localizadas virtualmente no infinito (nuvens, por exemplo). A proximidade entre dois objetos também pode gerar pontos salientes no espaço entre eles.

Os objetivos deste trabalho podem ser assim resumidos:

---

<sup>b</sup>Como será descrito posteriormente, o sistema não considera efeitos de iluminação local causados por transparência, refração ou reflexão.



1. Determinar quais pontos salientes estão de fato associados a algum marco da cena e estimar sua posição tridimensional, bem como a incerteza dessa estimação;
2. Utilizar a informação inferida sobre a posição dos marcos e a pose da câmera para refinar simultaneamente a estimação de ambos (*SLAM*); e
3. Utilizar toda a informação disponível para determinar uma configuração que deve ser buscada a fim de prosseguir com a reconstrução do objeto, buscando regiões não exploradas ou reobservando regiões conhecidas a fim de refinar a estimação corrente.

Para cada um dos pontos salientes detectados em uma dada imagem, uma única entre as seguintes alternativas é sempre válida em relação ao marco correspondente:

- O marco nunca foi observado anteriormente, portanto não é possível inferir qualquer informação sobre a sua distância em relação à câmera. Nesse caso, várias hipóteses são lançadas para a profundidade<sup>c</sup> desse marco;
- O marco já foi observado anteriormente, portanto a observação corrente (a partir de um ponto de vista diferente) adiciona informação sobre a posição do marco. Nesse caso, duas ações são realizadas:
  1. A posição do ponto saliente no espaço de imagem é usada para reduzir a incerteza sobre a localização de cada hipótese sobre a posição do marco;
  2. As hipóteses menos prováveis (isto é, cujas projeções sobre o plano de imagem possuem a menor verossimilhança quando comparadas com as coordenadas do correspondente ponto saliente da imagem corrente) são eliminadas.

Após algumas observações repetidas de um determinado marco, o conjunto de hipóteses é gradativamente reduzido até restar uma única hipótese. Essa passa a ser considerada a melhor estimativa atual da posição do marco. A partir desse ponto, as novas observações desse marco serão utilizadas para o processo de *SLAM*, ou seja, para simultaneamente (i) reduzir ainda mais a incerteza de sua localização e (ii) corrigir a estimativa da pose do robô.

---

<sup>c</sup>Neste trabalho, o termo *profundidade* será utilizado para denotar a distância de um marco ao centro de projeção da câmera.

## 3.2 Suposições

As seguintes suposições são adotadas como base para a metodologia proposta:

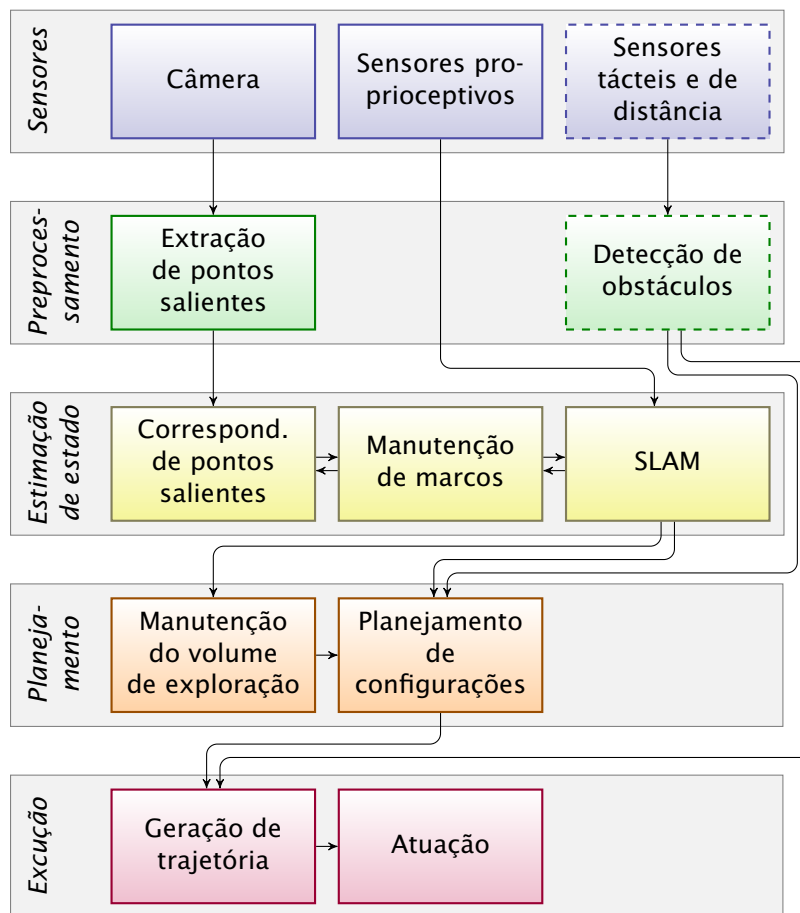
- A cena é suficientemente iluminada e possui características geométricas e/ou radiométricas suficientemente salientes para que possam ser capturadas por detectores de pontos salientes discutidos na [Subseção 2.3.2](#). As condições de iluminação da cena permanecem constantes ao longo do tempo, ou suas variações não influenciam os resultados desses detectores de pontos salientes;
- Adota-se o modelo de iluminação global, ou seja, na cena não há objetos transparentes ou translúcidos ou efeitos de reflexão, refração ou difração;
- Os parâmetros intrínsecos da câmera são conhecidos, assim como a relação geométrica (isto é, sua pose relativa) em relação ao robô. Mais especificamente, dada uma pose para a câmera e as coordenadas de um ponto em sua imagem, é necessário que seja conhecida a relação biunívoca entre este ponto e a reta da cena<sup>d</sup> que contém o conjunto de pontos da cena que são projetados sobre esse ponto;
- O sistema proposto é discreto no tempo, ou seja, todos os eventos tratados referem-se a um determinado instante de tempo  $t$ . Sem perda de generalidade, convencionou-se que  $t \in \mathbb{N}$  e que os eventos ocorrem a partir de um instante inicial  $t = 0$ ;
- Considera-se que todos os ruídos são gaussianos, brancos, aditivos, de média zero e descorrelacionados entre si. Também considera-se que as incertezas acerca das informações podem ser aproximadas por distribuições gaussianas. Por extensão, todos os vetores de estado, de observação e de controle, quando tratados estocasticamente, podem ser completamente representados por [PDFs gaussianas multivariadas](#).

## 3.3 Arquitetura proposta

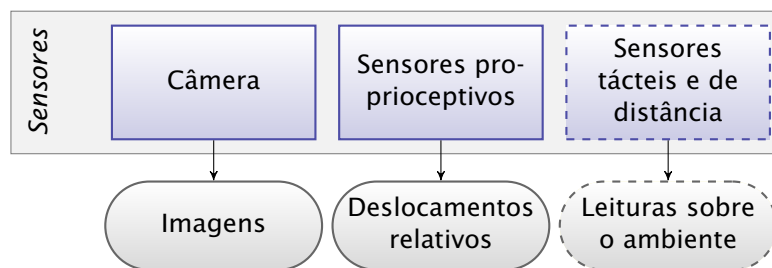
A arquitetura desenvolvida para o presente trabalho é ilustrada na [Figura 3.1](#) e compõe-se de cinco camadas: *Sensores*, *Preprocessamento*, *Estimação de estado*, *Planejamento* e *Execução*. Cada camada, assim como os módulos que a compõe, será apresentada nas subseções seguintes.

---

<sup>d</sup>Como a suposição anterior estabelece o modelo de iluminação global, o conjunto de possíveis pontos da cena que gera um determinado ponto na imagem é restrito a uma reta.



**Figura 3.1.** Diagrama de organização das camadas e módulos da arquitetura proposta.



**Figura 3.2.** Diagrama da camada de sensores.

### 3.3.1 Camada de sensores

Esta camada (Figura 3.2) é responsável pela coleta de dados da cena pelo robô. Na prática, representa todos os sensores embarcados no robô: a *câmera*, responsável pela captura de imagens; os *sensores proprioceptivos*, responsáveis pela percepção do deslocamento do robô; e opcionalmente os *sensores tácteis e de distância*, responsáveis pela detecção de obstáculos durante o deslocamento do robô.

Esta camada é composta pelos seguintes módulos:

**Câmera.** A *câmera* é o sensor fundamental para este trabalho. Adota-se o modelo de câmera projetiva *pinhole*, onde as coordenadas de pontos da cena e da imagem são relacionadas por uma transformação projetiva [Horn, 1986]. Convenciona-se chamar de  $\mathcal{I}_t$  a imagem obtida pela câmera no instante  $t$ .

A câmera também representa um aspecto essencial da metodologia desenvolvida. Neste trabalho, a pose do robô é sempre descrita em relação à pose da câmera. Portanto, “estimar a pose do robô” na realidade refere-se ao processo de estimação da posição e orientação da câmera. Da mesma maneira, “planejar a pose futura do robô” significa traçar um conjunto de ações capazes de levar a câmera para uma pose desejada (respeitadas as restrições holonômicas e o espaço de configurações do robô). Caso o acoplamento entre o robô e a câmera seja articulado, isto possivelmente significa coordenar o movimento do robô com técnicas de cinemática inversa.

Uma pose arbitrária da câmera em um instante  $t$  é representada por um vetor-coluna  $\mathbf{r}_t$  assim definido:

$$\mathbf{r}_t \triangleq \begin{bmatrix} \mathbf{c}_t^\top & \mathbf{q}_t^\top \end{bmatrix}^\top, \quad (3.1)$$

onde  $\mathbf{c}_t$  é um vetor contendo as coordenadas tridimensionais do centro de projeção da câmera e  $\mathbf{q}_t$  é um quatérnio unitário que representa a orientação da câmera no espaço tridimensional, ambos em relação a um sistema de coordenadas da cena arbitrariamente definido.

**Sensores proprioceptivos.** Este módulo é responsável pela aquisição dos dados gerados pelos sensores proprioceptivos instalados no robô (hodômetros, sensores inerciais, etc.).

O deslocamento relativo registrado pelo robô no intervalo de tempo  $(t - 1, t]$  é representado por um vetor  $\mathbf{u}_t$  e utilizado como entrada para a etapa de predição do método de estimação do estado do robô (Subseção 5.3.2). Assim, dada uma pose arbitrária da câmera  $\mathbf{r}_{t-1}$ ,  $\mathbf{u}_t$  abrange toda a informação necessária para estimar a pose *a priori* (isto é, sem incorporar as evidências da observação)  $\mathbf{r}_t^-$ , que corresponde à aplicação do deslocamento apontado pelos sensores proprioceptivos sobre  $\mathbf{r}_{t-1}$ .

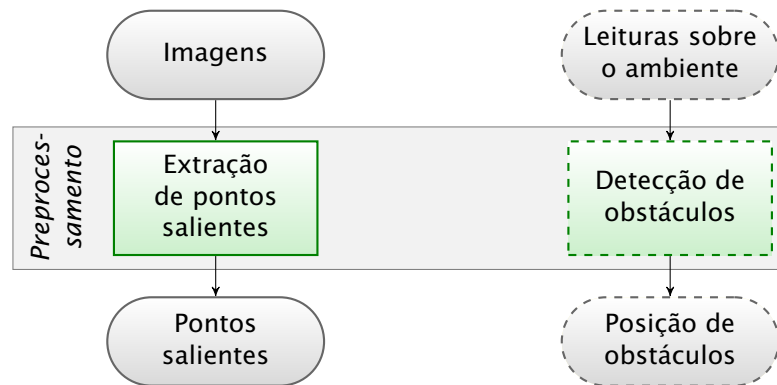


Figura 3.3. Diagrama da camada de pré-processamento.

**Sensores tácteis e de distância.** Este módulo compreende todos os sensores capazes de localizar obstáculos na cena, tanto pela avaliação da distância entre o robô e o obstáculo (LRFs, sonares, sensores infravermelhos, etc.) quanto pelo contato (*bumpers*).

Os sensores de obstáculos não são essenciais para a metodologia deste trabalho. Se estiverem disponíveis, as informações geradas por este módulo servem como dados de entrada para o módulo de detecção de obstáculos.

### 3.3.2 Camada de pré-processamento

Esta camada (Figura 3.3) tem por objetivo extrair informações relevantes a partir dos dados fornecidos pelos sensores da camada anterior. É composta por dois módulos:

**Extração de pontos salientes.** É responsável pela identificação de pontos salientes em uma imagem. Formalmente, dada uma imagem de entrada  $\mathcal{I}_t$ , este módulo é responsável por detectar e retornar um conjunto de pontos salientes:

$$\mathcal{F}_t \triangleq \{F_t^1, \dots, F_t^N\}, \quad (3.2)$$

onde  $N$  é o número de pontos salientes detectados. Cada ponto saliente detectado  $F_t^f$  é definido como uma tupla contendo os seguintes dados:

$$F_t^f \triangleq \langle x_t^f, y_t^f, s_t^f, \phi_t^f, \mathbf{d}_t^f \rangle, \quad (3.3)$$

onde  $(x_t^f, y_t^f)$  são as coordenadas do centroide do ponto saliente, em pixels;  $s_t^f$  é um fator de escala;  $\phi_t^f$  é a orientação de alguma característica particular da vizi-

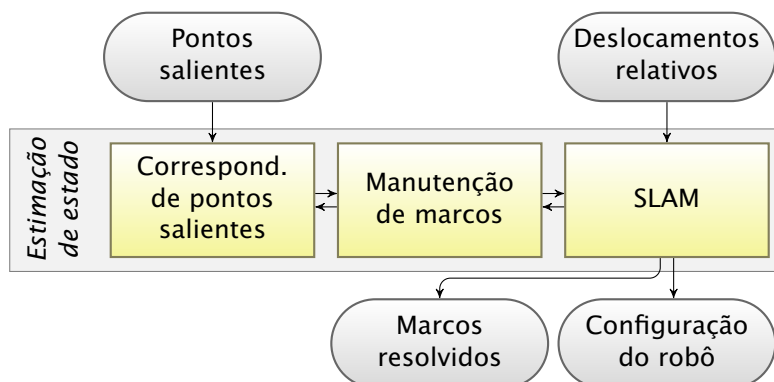


Figura 3.4. Diagrama da camada de estimação de estado.

nhança do ponto saliente; e  $\mathbf{d}_t^f$  é o vetor descritor (um vetor com uma quantidade fixa  $K$  de elementos que representa as características visuais da vizinhança do ponto saliente). Diversos algoritmos para detecção de pontos salientes são capazes de retornar esse conjunto de informações, dentre os quais SIFT [Lowe, 1999, 2004] e SURF [Bay et al., 2006].

**Detecção de obstáculos.** Este módulo é responsável pela identificação das coordenadas de obstáculos próximos ou em colisão com o robô. Ele não é essencial para a metodologia e serve apenas para prover informações adicionais para a camada de execução, de modo que a geração de trajetórias possa ser realizada por algoritmos capazes de evitar colisões.

### 3.3.3 Camada de estimação de estado

Esta camada (Figura 3.4) compreende os módulos responsáveis por integrar as informações fornecidas pelos módulos de extração de pontos salientes e de sensores proprioceptivos, com o objetivo de fornecer estimativas e respectivas incertezas acerca da pose da câmera e das coordenadas dos marcos da cena. Esta camada incorpora uma das contribuições científicas propostas por este trabalho e é discutida em detalhes no Capítulo 5.

O objetivo da estimação de estado é prover, a cada instante de tempo  $t$ , estimações para as seguintes informações para as próximas camadas:

- a pose da câmera do robô,  $\mathbf{r}_t$ ; e
- algumas informações sobre cada marco da cena, em particular a sua posição tridimensional,  $\mathbf{m}_t^m$ .

Para estabelecer uma notação clara sobre a “posição de um marco”, é importante lembrar que este trabalho adota uma abordagem multi-hipotética para resolver o problema da triangulação atrasada. Em vez de criar uma única estimativa para a posição de um marco, várias hipóteses são lançadas para cada nova observação de um possível marco (Seção 3.1). Novas observações do mesmo marco eliminam hipóteses menos prováveis até que uma única hipótese remanesça. Até isto ocorrer, a “posição do marco” não é definida (já que cada hipótese possui suas próprias coordenadas tridimensionais).

Desta forma, um *marco* será formalmente definido como o conjunto das coordenadas de suas respectivas hipóteses:

$$\mathbf{m}_t^m \triangleq \{\mathbf{h}_t^{m,1}, \mathbf{h}_t^{m,2}, \dots, \mathbf{h}_t^{m,H_t^m}\}, \quad (3.4)$$

onde  $\mathbf{m}_t^m$  é o  $m$ -ésimo marco conhecido até o momento,  $\mathbf{h}_t^{m,h}$  é a  $h$ -ésima hipótese associada a  $\mathbf{m}_t^m$  e  $H_t^m$  é a quantidade atual de hipóteses associadas ao marco  $\mathbf{m}_t^m$ . Com base nesta formalização, duas definições serão adotadas:

- Um *marco não resolvido* é aquele que possui mais de uma hipótese associada, ou seja,  $H_t^m > 1$ ;
- Um *marco resolvido* é aquele que possui uma única hipótese associada, ou seja,  $H_t^m = 1$ . Por extensão, a “posição de um marco conhecido” refere-se às coordenadas dessa hipótese.

Com base nestes conceitos, os módulos da camada de estimação de estados são apresentados a seguir:

**Correspondência de pontos salientes.** Este módulo tem por objetivo triar e corresponder o conjunto de pontos salientes  $F_t^f$  (Eq. (3.3)) fornecidos pelo módulo de extração de pontos salientes. Em outras palavras, este módulo determina se cada ponto saliente corresponde a um marco conhecido (resolvido ou não) ou se ele representa um novo marco, sem histórico de observações.

A correspondência entre pontos salientes de duas imagens é uma das principais contribuições científicas deste trabalho e será tratada em detalhes no **Capítulo 4**.

**Manutenção de marcos.** Este módulo mantém um banco de dados contendo todos os marcos rastreados. Três são as tarefas realizadas por este módulo:

- adicionar ao banco de dados os novos marcos (e respectivas hipóteses) para os pontos salientes não correspondidos;

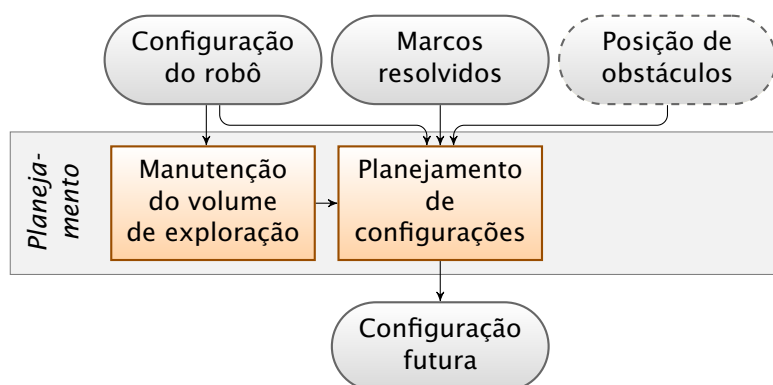


Figura 3.5. Diagrama da camada de planejamento.

- eliminar as hipóteses menos prováveis dos marcos não resolvidos; e
- eliminar os marcos resolvidos quando as suas coordenadas forem inconsistentes com as observações (eliminação de *outliers*).

**SLAM.** Este módulo utiliza um banco de filtros de Kalman *Unscented* para estimar a pose da câmera do robô e a posição de todas as hipóteses existentes. Para a etapa de predição são utilizadas as informações sobre os deslocamentos relativos (fornecidos pelo módulo de sensores proprioceptivos); para a etapa de correção, os pontos salientes correspondidos (triados pelo módulo de correspondência de pontos salientes) são utilizados.

O funcionamento do banco de filtros de Kalman e a sua integração com o módulo de manutenção de marcos compõem o cerne de toda a metodologia exposta neste trabalho. O assunto será tratado em detalhes no [Capítulo 5](#).

### 3.3.4 Camada de planejamento

Esta camada (Figura 3.5) tem por objetivo analisar o conjunto de informações disponíveis para determinar uma pose futura para a câmera. Essa configuração futura é estabelecida pela busca simultânea de três objetivos principais:

1. *Melhoria da localização*, pela observação de marcos resolvidos. A observação de tais marcos permite corrigir a estimativa da pose da câmera e consequentemente reduzir a sua incerteza;
2. *Melhoria do mapa*, pela observação de marcos não resolvidos ou de marcos resolvidos mas pouco observados. De maneira análoga à melhoria da localização, a observação desses marcos permite reduzir a incerteza da estimativa de suas posições e, no caso de marcos não resolvidos, promovê-los a resolvidos;



3. *Exploração*, pela observação de áreas não exploradas, de modo a expandir a fronteira de conhecimento acerca da geometria do objeto de interesse.

O planejamento de poses futuras é primordialmente baseado em duas decisões: (i) qual região do espaço deve ser observada e (ii) a partir de que ponto de vista essa região deve ser observada. Naturalmente essas duas decisões são fortemente acopladas e intuitivamente se traduzem na determinação da posição da câmera (2ª questão) e de sua orientação (1ª questão). Ainda seguindo uma argumentação intuitiva, os três comportamentos descritos acima podem ser descritos como segue:

1. A melhoria da localização é atingida ao se levar o robô para uma pose previamente visitada, ou seja, quando se enquadra um conjunto de marcos resolvidos a partir de pontos de vista que propiciam a sua reobservação;
2. A melhoria do mapa é atingida da mesma maneira, porém buscando enquadrar marcos pouco observados e a partir de novos pontos de vista, propiciando assim a redução de incertezas por triangulação;
3. A exploração pode ser buscada quando se posiciona a câmera de modo a observar regiões desconhecidas da cena. No entanto, a exploração não pode ser conduzida pelo posicionamento arbitrário da câmera: Como não há um sistema global de localização, é necessário corroborar constantemente estimação da pose da câmera por meio da observação de regiões conhecidas (marcos resolvidos). Portanto, é melhor afirmar que a exploração deve ser buscada quando se posiciona a câmera de modo a enquadrar tanto uma região conhecida (isto é, um conjunto de marcos resolvidos) quanto outra desconhecida (permitindo a observação de novos pontos salientes e conseqüentemente a criação de novos marcos).

Uma questão importante (ainda que sutil) da descrição exposta se refere à classificação das regiões da cena. Em alto nível, pode-se dizer que o espaço tridimensional pode ser segmentado em três categorias: (i) *regiões ocupadas* (contendo os objetos reconstruídos), (ii) *regiões livres* (aquelas dentro das quais o robô pode se movimentar livremente) e (iii) *regiões inexploradas* (sobre as quais não foram coletadas evidências sobre a sua ocupação). No que diz respeito ao planejamento das poses futuras, fica claro que os marcos resolvidos determinam as regiões ocupadas; no entanto, não há nada no processo de estimação (Subseção 3.3.3) que permita identificar as regiões livres e, por exclusão, as inexploradas. De fato, essa informação é irrelevante para o processo de reconstrução dos objetos da cena, ainda que essencial para o processo de planejamento.

Portanto, a camada de planejamento deve ser responsável, antes de mais nada, por manter uma estrutura de dados capaz de permitir essa classificação. Para este fim, o espaço tridimensional será discretizado em células tridimensionais igualmente espaçadas, de maneira análoga à representação por voxels. O valor atribuído a cada célula, designado por  $o_t^{x,y,z}$ , representa o conhecimento acumulado até o momento  $t$  sobre a ocupação do correspondente espaço geométrico, em forma probabilística. Assim:

$$o_t^{x,y,z} \triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \mathcal{I}_{0\dots t}). \quad (3.5)$$

Informalmente, um alto valor (próximo de 1) para uma célula  $o_t^{x,y,z}$  indica uma alta probabilidade de haver um objeto na região ocupada pela célula centralizada nas coordenadas  $(x, y, z)$ , enquanto um baixo valor (próximo de 0) sinaliza alta probabilidade de que essa célula esteja localizada em espaço livre.

Deve-se ter em mente que, embora essa estrutura seja praticamente idêntica a uma **OGM** adaptada para o espaço tridimensional, ela não será utilizada para inferir a forma dos objetos construídos, apenas para prover informações para os módulos da camada de planejamento. Em particular, essa discretização deve ser ajustada para prover uma representação de baixa resolução da cena, em contraposição ao uso tradicional da **OGM**, onde se espera a melhor resolução possível dentro das limitações computacionais.<sup>e</sup>

Os módulos desta camada são apresentados a seguir:

**Manutenção dos volumes de exploração.** Este módulo é responsável por manter a estrutura da **OGM** tridimensional e atualizá-la com base nos dados fornecidos pela camada de estimação de estado.

**Planejamento de configurações.** Este módulo é responsável pela determinação da configuração futura que deve ser adotada pelo robô, chamada de  $\mathbf{r}_{\text{fut}}$ , com base na classificação das regiões espaciais em livres ou ocupadas fornecida pelo módulo anterior.

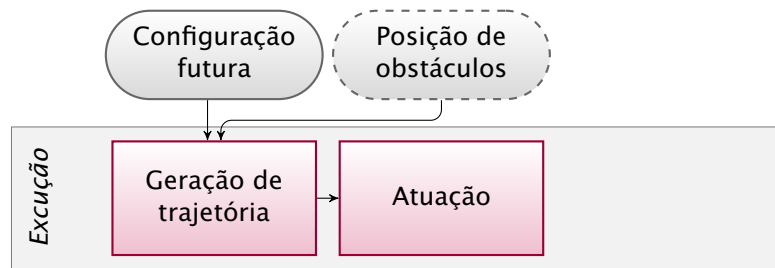


Figura 3.6. Diagrama da camada de execução.

### 3.3.5 Camada de execução

Esta camada (Figura 3.6) tem por objetivo estabelecer e executar o conjunto de ações necessárias para levar o robô da configuração corrente,  $\mathbf{r}_t$ , até a configuração planejada,  $\mathbf{r}_{\text{fut}}$ .

A geração da trajetória é efetuada de modo a evitar as seguintes classes de obstáculos:

- os marcos resolvidos (definidos na Subseção 3.3.3),  $\{\mathbf{m}_t^m\}$ ; e
- os obstáculos fornecidos pelo módulo de detecção de obstáculos (Subseção 3.3.2).

Dois são os módulos que compõem esta camada:

**Geração de trajetória.** Este módulo é responsável por estabelecer o caminho a ser seguido pelo robô a fim de atingir a pose desejada, evitando a colisão com os obstáculos conhecidos (marcos definitivos e aqueles eventualmente descobertos pelo módulo de detecção de obstáculos) e respeitando as restrições holonômicas do robô, se existirem.

**Atuação.** Este módulo é responsável pela execução da trajetória determinada no módulo anterior pela comunicação direta com os atuadores do robô.

É importante notar que o presente trabalho não pretende apresentar inovações científicas nas áreas de geração e execução de trajetórias.

---

<sup>e</sup>Para que se tenha uma ideia das ordens de grandeza dos parâmetros aqui discutidos, uma “representação de baixa resolução” significa que, para uma cena com um objeto com dimensões próximas a 1 m, células medindo 10 cm de lado são suficientes para realizar o planejamento. Por outro lado, note que uma reconstrução geométrica com essa resolução seria considerada grosseira para a maioria das aplicações.



## Capítulo 4

# Correspondência entre pontos salientes de duas imagens

ESTE CAPÍTULO DESCREVE O MÓDULO responsável por manter um banco de dados dos pontos salientes observados pela câmera e de estabelecer a correspondência entre pares de pontos de imagens distintas, fornecendo dados que permitem o processo de triangulação que ocorre nos módulos de manutenção de marcos e de SLAM (Capítulo 5).

### 4.1 Definições preliminares

Tipicamente, a etapa de correspondência realizada pelos algoritmos de detecção e correspondência de pontos salientes (SIFT, SURF, etc.) pode ser resumida como segue:

- Dadas duas imagens  $\mathcal{I}_p$  e  $\mathcal{I}_q$  e o conjunto dos descritores de todos os pontos salientes encontrados em cada imagem,  $\mathcal{D}_p = \{\mathbf{d}_p^1, \dots, \mathbf{d}_p^{N_p}\}$  e  $\mathcal{D}_q = \{\mathbf{d}_q^1, \dots, \mathbf{d}_q^{N_q}\}$ , onde  $\mathbf{d}_*^*$  são vetores com  $K$  elementos e  $N_p$  e  $N_q$  são o número de pontos salientes detectados respectivamente nas imagens  $\mathcal{I}_p$  e  $\mathcal{I}_q$  (ver Eq. (3.3));
- dada uma *função de distância entre descritores*,  $\text{dist} : \mathbb{R}^K \times \mathbb{R}^K \rightarrow \mathbb{R}^+$  (possivelmente, mas não necessariamente, a distância Euclideana), onde  $\mathbb{R}^+$  é o conjunto de números reais não negativos;
- para cada descritor  $\mathbf{d}_p^i$ , encontrar o descritor  $\mathbf{d}_q^j$  que minimiza a distância  $\text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j)$ .

O resultado é um conjunto de pares  $\langle i, j \rangle$  de pontos correspondentes. Formalmente, o conjunto  $C$  de todos os pares correspondidos pode ser descrito por:

$$C(\mathcal{D}_p, \mathcal{D}_q) = \{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) \rangle \} \quad \forall 1 \leq i \leq N_p. \quad (4.1)$$

Esta abordagem é obviamente sujeita a erros, já que *todos* os pontos salientes da primeira imagem serão correspondidos com algum ponto saliente da segunda imagem, o que pode gerar falsas correspondências — principalmente porque algumas regiões da cena observadas na primeira imagem podem não ser observadas na segunda, entre outros motivos). Para contornar este problema, uma solução típica é aceitar uma correspondência apenas se a distância de  $\mathbf{d}_p^i$  para a segunda melhor correspondência em  $\mathbf{d}_q^*$  for significativamente maior. Formalmente, um par  $\langle i, j \rangle$  é rejeitado se houver um par  $\langle i, k \rangle$ , onde  $1 \leq k \leq N_q$  e  $k \neq j$ , para o qual  $\text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^k) < \tau_M \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j)$  para um dado *limiar de distinguibilidade*  $\tau_M \geq 1$ , ou:

$$C(\mathcal{D}_p, \mathcal{D}_q, \tau_M) = \left\{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) \rangle : \forall k \neq j [ \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^k) \geq \tau_M \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) ] \right\}. \quad (4.2)$$

A Eq. (4.1) representa o caso particular sem restrição de distinguibilidade, onde  $\tau_M = 1$ . Entretanto, a solução da Eq. (4.2) impede a correspondência entre pontos salientes com características visuais semelhantes, como padrões recorrentes na cena ou texturas repetitivas, mesmo que esses pontos salientes possuam características particularmente distintivas. Em outras palavras, o uso do limiar de distinguibilidade causa o descarte de informações possivelmente importantes.

Uma abordagem melhor consiste em adotar outras restrições e manter  $\tau_M = 1$ , efetivamente permitindo correspondências de pontos com características repetidas. Isto pode ser obtido pela procura de um consenso geral de transformações geométricas sofridas pelos pontos salientes de uma imagem para outra, antes de avaliar as correspondências individuais. Para buscar este consenso, um fato conveniente é que diversos algoritmos de detecção de pontos salientes retornam não somente o descritor associado a cada ponto, mas também um conjunto de informações geométricas. Transcrevendo a Eq. (3.3) por conveniência, um ponto saliente  $F_t^f$  pode ser descrito como uma tupla com pelo menos os seguintes dados:

$$F_t^f = \langle x_t^f, y_t^f, s_t^f, \phi_t^f, \mathbf{d}_t^f \rangle, \quad (4.3)$$

onde  $x_t^f$  e  $y_t^f$  são as coordenadas do centroide do ponto saliente, em pixels;  $s_t^f$  é um fator de escala;  $\phi_t^f$  é a orientação de alguma característica particular da vizinhança do ponto saliente; e  $\mathbf{d}_t^f$  é o vetor descritor. Assim, o conjunto de pontos salientes detectados em uma imagem é representado por:

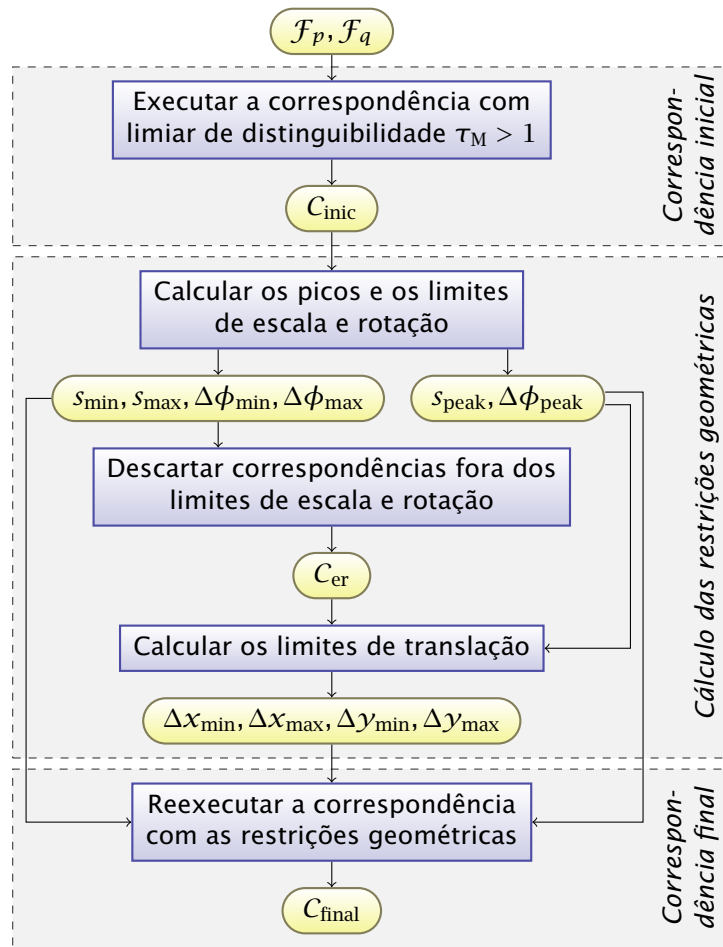
$$\mathcal{F}_t = \{F_t^1, \dots, F_t^N\}. \quad (4.4)$$

Em geral, o fator de escala  $s_t^f$  e o ângulo de orientação  $\phi_t^f$  não possuem nenhum significado no espaço da imagem, e servem apenas para comparar a geometria da região vizinha de  $(x_t^f, y_t^f)$  entre as duas imagens. Por exemplo, dado um conjunto de  $\{\langle i, j \rangle\}$  correspondências corretas entre as duas imagens,  $\mathcal{I}_p$  e  $\mathcal{I}_q$ , onde  $\mathcal{I}_q$  é uma vista aproximada (*zoom*) de uma região qualquer de  $\mathcal{I}_p$ , então todas as razões  $s_q^j/s_p^i$  devem manter uma consistência global sobre a transformação de escala, ou seja,  $s_q^j/s_p^i \sim \text{constante}$ . O mesmo se aplica à diferença entre as orientações,  $\phi_q^j - \phi_p^i$ , se a segunda imagem é uma vista rotacionada (em relação ao eixo principal da câmera) da primeira imagem.

No caso geral, não se pode esperar a mesma razão de escala  $s_q^j/s_p^i$  ou a mesma mudança de orientação  $\phi_q^j - \phi_p^i$  para todas as correspondências  $\langle i, j \rangle$ . No entanto, se a cena for estática e o movimento da câmera não causar uma transformação perspectiva significativa entre duas imagens consecutivas — duas restrições compatíveis com a proposição deste trabalho —, então pode-se esperar que tanto a rotação quanto a translação sejam confinadas a um intervalo bem definido.

A metodologia proposta para o processo de correspondência é apresentada na Figura 4.1: Dado o conjunto de correspondências  $C_{\text{inic}} = \{\langle i, j \rangle\}$  gerado por qualquer método (por exemplo, pelo proposto na Eq. (4.2)), analisa-se todas as transformações de escala, rotação e translação entre os pontos correspondidos e calcula-se limites aceitáveis para cada uma dessas transformações. Em seguida, uma nova correspondência é executada baseada somente nesses limites (ou seja, adotando o limiar de distinguibilidade  $\tau_M = 1$ ).

Embora a Eq. (4.2) possa ser usada para avaliar a correspondência inicial  $C_{\text{inic}}$ , isto implicaria que o método proposto fosse necessariamente mais lento do que os métodos tradicionais (já que o tempo total de execução inclui o tempo necessário para computar  $C_{\text{inic}}$ ). Um método mais eficiente para avaliar a correspondência inicial é visto a seguir. Detalhes sobre os passos seguintes são apresentados nas Subseções 4.3 e 4.4.



**Figura 4.1.** Visão geral da metodologia de correspondência de pontos salientes. A partir de um conjunto inicial de correspondências, as transformações de escala, rotação e translação são analisadas para calcular as restrições geométricas. A correspondência é novamente executada sob essas restrições.

## 4.2 Correspondência inicial

A avaliação da correspondência de acordo com a Eq. (4.2) possui custo assintótico de tempo da ordem de  $O(K N_p N_q)$ . Reduzir a complexidade assintótica do processo não é uma tarefa trivial; porém, é possível reduzir o custo computacional ao utilizar somente um subconjunto de  $\mathcal{F}_p$  para avaliar  $C_{inic}$ . Embora o número de correspondências resultantes seja em geral menor do que o obtido pela Eq. (4.2), na prática isso não compromete o restante da metodologia, já que  $C_{inic}$  é utilizado apenas para estimar os valores numéricos dos limites utilizados na etapa de correspondência final.

Desta forma, define-se  $\mathcal{Z}$  como sendo um subconjunto aleatório de  $\mathcal{F}_p$ , onde



a cardinalidade de  $Z$  é fixada em

$$|Z| = \left\lfloor \frac{|\mathcal{F}_p|}{f_a} \right\rfloor \quad (4.5)$$

para um dado *fator de amostragem*  $f_a \geq 1$ . A correspondência inicial  $C_{\text{inic}}$  é definida como segue, aplicando a Eq. (4.5) como entrada para a Eq. (4.2):

$$C_{\text{inic}} = \left\{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) \rangle : \right. \\ \left. \nexists k \neq j [ \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^k) < \tau_M \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) ] \right\} \\ \forall \mathbf{d}_p^i \in Z. \quad (4.6)$$

Apesar de o custo de tempo assintótico permanecer o mesmo (já que  $O(K \frac{N_p}{f_a} N_q) = O(K N_p N_q)$ ), na prática o tempo consumido para computar  $C_{\text{inic}}$ , para valores altos de  $f_a$ , é significativamente menor quando comparado com a proposição original da Eq. (4.2). O valor a ser adotado para  $f_a$  será empiricamente determinado como parte dos experimentos apresentados neste trabalho, no [Capítulo 7](#).

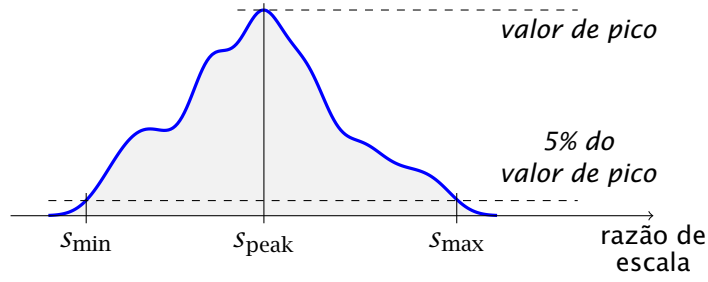
### 4.3 Determinação dos limites geométricos

O objetivo deste passo é definir os limites aceitáveis para todas as transformações geométricas consideradas: escala, rotação e translação. As duas primeiras — escala e rotação — podem ser diretamente estimadas com base nos fatores de escala  $s_t^*$  e nos ângulos de orientação  $\phi_t^*$ , fornecidos pelo detector de pontos salientes, conforme a Eq. (4.3). Por outro lado, as translações dos pontos somente podem ser consistentemente analisadas após a determinação das demais transformações. Portanto, a determinação dos limites geométricos será executada em dois passos: (i) escala e rotação e (ii) translação.

#### 4.3.1 Determinação dos limites de escala e rotação

Neste passo, as razões de escala e as diferenças de orientação são analisadas entre os pontos correspondidos. Em seguida, limites para os dois casos são determinados e todas as correspondências que não obedecem a esses limites são descartados.

Para a transformação de escala, o seguinte procedimento é adotado: Dado o



**Figura 4.2.** Determinação dos limites aceitáveis para a razão de escala. O mesmo procedimento é aplicado para a determinação dos limites aceitáveis para as diferenças de orientação.

conjunto de todas as razões de escala, representado por  $S_M$  e definido por:

$$S_M \triangleq \{s_q^j / s_p^i\} \quad \forall \langle i, j \rangle \in C_{\text{inic}}, \quad (4.7)$$

primeiro estima-se a PDF de  $S_M$  por meio de janelas de Parzen [Parzen, 1962] com núcleo gaussiano. Para a estimação da largura do núcleo será utilizado o método proposto por Botev [2006]. A partir da PDF estimada, os seguintes valores são extraídos (veja Figura 4.2):

- a moda da razão de escala (o valor correspondente ao pico da PDF),  $s_{\text{peak}}$ ;
- a maior razão de escala menor do que  $s_{\text{peak}}$  cuja verossimilhança seja igual a 5% da moda,  $s_{\text{min}}$ ; e
- a menor razão de escala maior do que  $s_{\text{peak}}$  cuja verossimilhança seja igual a 5% da moda,  $s_{\text{max}}$ .

O mesmo procedimento se aplica para avaliar o pico e os limites do ângulo de rotação (diferença de orientações). Dado o conjunto de todas as rotações,  $\mathcal{R}_M$ , definido como:

$$\mathcal{R}_M \triangleq \{\phi_q^j - \phi_p^i\} \quad \forall \langle i, j \rangle \in C_{\text{inic}}, \quad (4.8)$$

estima-se a PDF de  $\mathcal{R}_M$ , novamente por meio de janelas de Parzen. Como os ângulos de rotação formam um espaço cíclico, a PDF estimada deve ser confinada a um intervalo com largura de  $2\pi$  rad e centrado no pico da verossimilhança  $\Delta\phi_{\text{peak}}$ , ou seja, a PDF é definida sobre o domínio  $(\Delta\phi_{\text{peak}} - \pi \text{ rad}, \Delta\phi_{\text{peak}} + \pi \text{ rad}]$ . Os limites  $\Delta\phi_{\text{min}}$  and  $\Delta\phi_{\text{max}}$  são definidos a partir dessa PDF exatamente como feito para  $s_{\text{min}}$  e  $s_{\text{max}}$ .

Finalmente, constroi-se o conjunto  $C_{\text{er}} \subseteq C_{\text{inic}}$  contendo somente as correspondências cujas transformações de escala e rotação estejam dentro dos limites

previamente definidos:

$$C_{er} \triangleq \{ \langle i, j \rangle \in C_{inic} : \\ (s_{\min} \leq s_q^j / s_p^i \leq s_{\max}) \wedge \\ \wedge (\Delta\phi_{\min} \leq \phi_q^j - \phi_p^i \leq \Delta\phi_{\max}) \}, \quad (4.9)$$

onde as diferenças  $\phi_q^j - \phi_p^i$  são normalizadas para o intervalo  $(\Delta\phi_{\text{peak}} - \pi \text{ rad}, \Delta\phi_{\text{peak}} + \pi \text{ rad}]$ .

### 4.3.2 Determinação dos limites de translação

Conforme explicado anteriormente, a translação dos pontos salientes é analisada sobre as suas coordenadas transformadas (escaladas e rotacionadas). Para cada par de correspondências  $\langle i, j \rangle \in C_{er}$ , define-se o *vetor de deslocamento transformado*  $\vec{v}_{i,j}$  como:

$$\vec{v}_{i,j} \triangleq \begin{bmatrix} \Delta x_{i,j} \\ \Delta y_{i,j} \end{bmatrix} \triangleq \begin{bmatrix} x_q^j \\ y_q^j \end{bmatrix} - s_{\text{peak}} \mathbf{R}_M \begin{bmatrix} x_p^i \\ y_p^i \end{bmatrix}, \quad (4.10)$$

onde  $\mathbf{R}_M$  é a matriz de rotação da moda das diferenças de rotação,  $\Delta\phi_{\text{peak}}$ :

$$\mathbf{R}_M \triangleq \begin{bmatrix} \cos \Delta\phi_{\text{peak}} & -\sin \Delta\phi_{\text{peak}} \\ \sin \Delta\phi_{\text{peak}} & \cos \Delta\phi_{\text{peak}} \end{bmatrix}. \quad (4.11)$$

Os limites de translação são assim definidos: Dado um histograma bidimensional sobre as projeções dos vetores  $\vec{v}_{i,j}$  sobre os eixos  $x$  e  $y$ , o retângulo que cobre a vizinhança conexa que contém o pico do histograma define os limites aceitáveis para ambos os eixos:  $\Delta x_{\min}$ ,  $\Delta x_{\max}$ ,  $\Delta y_{\min}$  e  $\Delta y_{\max}$ .

## 4.4 Correspondência final

A partir das restrições avaliadas no passo anterior, constroi-se o conjunto final de pares de correspondências,  $C_{\text{final}}$ :

$$\begin{aligned}
 C_{\text{final}} \triangleq \{ \langle i, \arg \min_j \text{dist}(\mathbf{d}_p^i, \mathbf{d}_q^j) \rangle : \\
 (s_{\min} \leq s_q^j / s_p^i \leq s_{\max}) \wedge \\
 \wedge (\Delta\phi_{\min} \leq \phi_q^j - \phi_p^i \leq \Delta\phi_{\max}) \wedge \\
 \wedge (\Delta x_{\min} \leq \Delta x_{i,j} \leq \Delta x_{\max}) \wedge \\
 \wedge (\Delta y_{\min} \leq \Delta y_{i,j} \leq \Delta y_{\max}) \}, \quad (4.12)
 \end{aligned}$$

novamente com as diferenças de orientação  $\phi_q^j - \phi_p^i$  normalizadas para o intervalo  $(\Delta\phi_{\text{peak}} - \pi \text{ rad}, \Delta\phi_{\text{peak}} + \pi \text{ rad}]$ . Note-se que, ao contrário das Eqs. (4.2) e (4.6), o limiar de distinguibilidade,  $\tau_M$ , não é usado — permitindo, por exemplo, a correspondência entre pontos salientes em regiões com texturas repetitivas nas imagens.

# Capítulo 5

## Estimação de estados

ESTE CAPÍTULO TEM POR OBJETIVO apresentar em detalhes a estrutura dos dois módulos que constituem a espinha dorsal de toda a metodologia: o *módulo de manutenção de marcos* e o *módulo de SLAM*. Esses módulos trabalham em estreita cooperação para realizar as seguintes tarefas:

1. manter um banco de dados de todos os marcos (resolvidos e não resolvidos)<sup>a</sup> e respectivas hipóteses, utilizando as informações fornecidas pelo módulo de correspondência de pontos salientes para criar novos marcos, reduzir as hipóteses dos marcos não resolvidos e eliminar os marcos resolvidos espúrios (*outliers*);
2. fundir as informações provenientes desse banco de dados, dos sensores proprioceptivos e do módulo de correspondência de pontos salientes para estimar recursivamente (isto é, a cada instante de tempo  $t$ ) a pose da câmera do robô,  $\mathbf{r}_t$ , e a posição de todas as hipóteses dos marcos<sup>b</sup>,  $\mathbf{h}_t^{m,h}$ , além das incertezas associadas a cada uma dessas estimações.

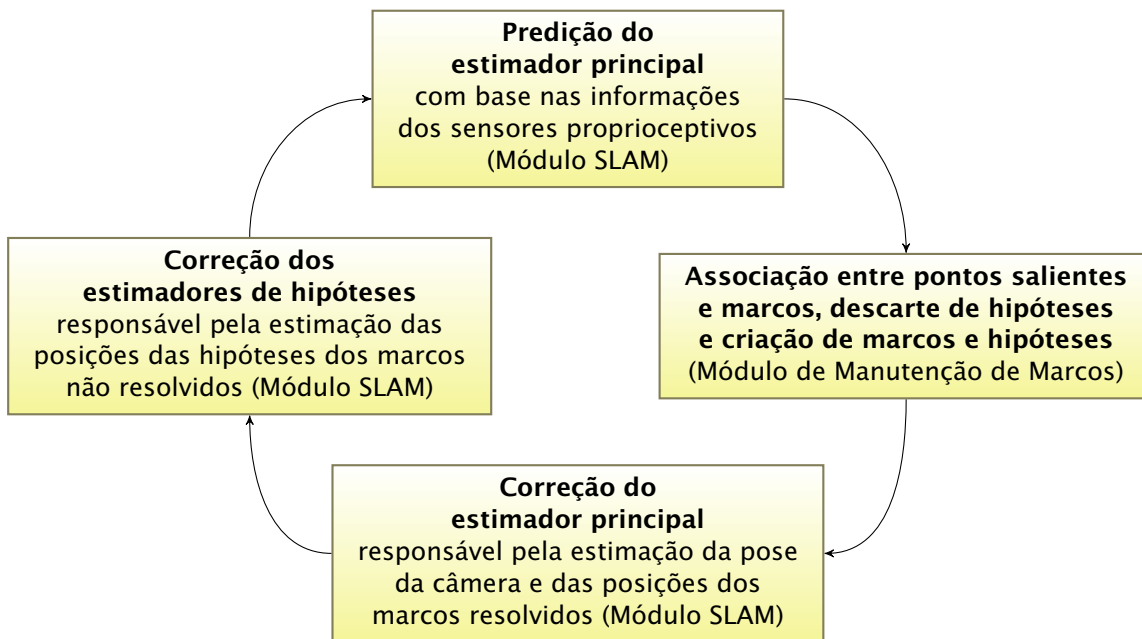
Os dois módulos em questão são responsáveis pelo ciclo principal de estimação de todas as variáveis relevantes deste trabalho (pose da câmera e posição dos marcos e hipóteses). Este ciclo é ilustrado na [Figura 5.1](#), que também apresenta as tarefas atribuídas a cada módulo.

Este capítulo está assim organizado: As definições matemáticas necessárias para o detalhamento dos módulos serão apresentadas na [Seção 5.1](#); a [Seção 5.2](#) descreve o módulo de manutenção de marcos; e a [Seção 5.3](#) detalha o módulo *SLAM*.

---

<sup>a</sup>“Marcos resolvidos” e “marcos não resolvidos”: definidos na [Subseção 3.3.3](#).

<sup>b</sup>Como consequência da definição de marco conhecido, apresentada na [Subseção 3.3.3](#), a estimação da posição das hipóteses implica a estimação da posição dos marcos conhecidos.



**Figura 5.1.** Ciclo principal de estimação de estados adotado neste trabalho, evidenciando a estreita cooperação entre o módulo de manutenção de marcos e o módulo SLAM. O fato de haver duas etapas de correção (do estimador principal e dos estimadores de hipóteses) refere-se ao uso de um banco de filtros de Kalman, cuja necessidade e implementação são discutidas na Seção 5.3.

## 5.1 Definições preliminares

No que diz respeito às variáveis de estado estimadas (pose da câmera e posição das hipóteses dos marcos), convém lembrar que neste trabalho supõe-se que todas as incertezas podem ser aproximadas por PDFs gaussianas multivariadas (vide Seção 3.2). Em outras palavras, as variáveis de estado e respectivas incertezas são representadas pelos seus dois primeiros momentos (média e covariância).

Assim,  $\hat{\mathbf{h}}_t^{m,h}$  refere-se ao valor esperado (ou seja, o valor de máxima verossimilhança) para a posição da  $h$ -ésima hipótese do  $m$ -ésimo marco, considerando-se toda a informação disponível até o instante  $t$ . A incerteza deste valor é representada por uma matriz de covariâncias  $\mathbf{H}_t^{m,h}$  de  $3 \times 3$  elementos, contendo as covariâncias entre as ordenadas de  $\hat{\mathbf{h}}_t^{m,h}$ . De maneira análoga, a estimação da pose da câmera compõe-se do vetor  $\hat{\mathbf{r}}_t$  (com 7 elementos: posição tridimensional e rotação representada por um quatérnio) e da respectiva matriz de covariâncias,  $\mathbf{R}_t$ , com  $7 \times 7$  elementos.

Dada uma pose arbitrária  $\mathbf{r}_t$  para uma câmera, conforme definida na Eq. (3.1), algumas funções são definidas:

**Função de projeção de um ponto da cena.** É uma função que, dadas as coordenadas de um ponto da cena,  $\mathbf{p}_{\text{cena}}$ , determina as coordenadas de sua projeção na imagem,  $\mathbf{p}_{\text{img}}$ , em pixels:

$$\mathbf{p}_{\text{img}} = \text{proj}(\mathbf{r}_t, \mathbf{p}_{\text{cena}}). \quad (5.1)$$

**Função de direção de um ponto da imagem.** Pode ser entendida como o dual da função de projeção. Dadas as coordenadas de um ponto da imagem em pixels,  $\mathbf{p}_{\text{img}}$ , esta função determina um vetor unitário (em unidades da cena) que define a direção do correspondente ponto da cena a partir do centro de projeção,  $\vec{\mathbf{v}}_{\text{cena}}$ :

$$\vec{\mathbf{v}}_{\text{cena}} = \text{diri}(\mathbf{r}_t, \mathbf{p}_{\text{img}}). \quad (5.2)$$

Pelas definições, segue-se que a seguinte relação é sempre verdadeira:

$$\mathbf{p}_{\text{img}} = \text{proj}(\mathbf{r}_t, \mathbf{p}_{\text{cena}}) \Leftrightarrow \mathbf{p}_{\text{cena}} \in \mathbf{c}_t + \lambda \text{diri}(\mathbf{r}_t, \mathbf{p}_{\text{img}}), \quad \lambda \in \mathbb{R}^+, \quad (5.3)$$

que pode ser lido como: “O ponto da imagem  $\mathbf{p}_{\text{img}}$  é a projeção do ponto da cena  $\mathbf{p}_{\text{cena}}$  se e somente se  $\mathbf{p}_{\text{cena}}$  encontra-se na semirreta cuja origem é o centro de projeção da câmera,  $\mathbf{c}_t$ , e que segue na direção do vetor  $\text{diri}(\mathbf{r}_t, \mathbf{p}_{\text{img}})$ .” Essa semirreta será chamada de *eixo de projeção* do ponto  $\mathbf{p}_{\text{img}}$ .

**Função de direção de um ponto da cena.** Esta função avalia o vetor unitário que indica a direção de um ponto da cena a partir do centro de projeção da câmera:

$$\vec{\mathbf{v}}_{\text{cena}} = \text{dirc}(\mathbf{r}_t, \mathbf{p}_{\text{cena}}). \quad (5.4)$$

Esta função depende somente das coordenadas do centro de projeção e é independente de sua orientação e dos parâmetros intrínsecos da câmera. De fato, ela pode ser avaliada trivialmente como segue:

$$\text{dirc}(\mathbf{r}_t, \mathbf{p}_{\text{cena}}) \triangleq \frac{\mathbf{p}_{\text{cena}} - \mathbf{c}_t}{\|\mathbf{p}_{\text{cena}} - \mathbf{c}_t\|} \quad (5.5)$$

e guarda a seguinte relação com as funções definidas nas Eqs. (5.1) e (5.2):

$$\text{dirc}(\mathbf{r}_t, \mathbf{p}_{\text{cena}}) = \text{diri}[\mathbf{r}_t, \text{proj}(\mathbf{r}_t, \mathbf{p}_{\text{cena}})]. \quad (5.6)$$

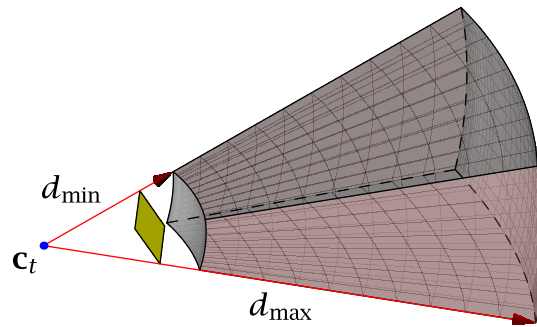


Figura 5.2. Espaço observável por uma câmera.

**Espaço observável por uma câmera.** Uma câmera com uma determinada configuração  $\mathbf{r}_t$  define uma região do espaço tridimensional que contém pontos que podem ser imageados pela câmera. Essa região será chamada de *espaço observável* e é delimitada pelas seguintes características da câmera (Figura 5.2):

- o **campo de visada** (*field-of-view*, ou **FoV**), uma restrição geométrica que representa os intervalos angulares horizontal e vertical além dos quais a projeção de um ponto da cena se encontra fora dos limites físicos do sensor. Considerando um sensor retangular e sem levar em consideração efeitos de distorções na imagem, o **FoV** delimita um espaço piramidal de altura infinita, com ápice no centro de projeção;
- a **profundidade de campo** (*depth-of-field*), uma restrição óptica que define um intervalo de distâncias, a partir do centro de projeção da câmera,  $[d_{\min}, d_{\max}]$  além e aquém do qual os pontos da cena são projetados fora de foco (borrados).

A interseção entre a pirâmide definida pelo **FoV** e a região limitada pela profundidade de campo definem uma região fechada no espaço tridimensional (Figura 5.2). Esta região será determinada pela *função de avaliação do espaço observável*,  $V(\mathbf{r}_t)$ .<sup>c</sup>

## 5.2 O módulo de manutenção de marcos

Conforme apresentado na introdução deste capítulo, o módulo de manutenção de marcos é responsável pelo gerenciamento de um banco de dados que contém informações sobre todos os marcos conhecidos e suas respectivas hipóteses. Os

<sup>c</sup>Tecnicamente,  $V(\cdot)$  também é função dos ângulos do **FoV** e das distâncias  $d_{\min}$  e  $d_{\max}$ . No entanto, esses parâmetros não serão explicitados nas equações por questões de clareza notacional.



dados fornecidos pelo módulo de correspondência de pontos salientes, assim como a interação com o módulo **SLAM**, fornecem as bases para que este módulo realize três operações principais: (i) Criação de marcos e respectivas hipóteses; (ii) eliminação das hipóteses menos prováveis de um marco não resolvido; e (iii) eliminação de marcos espúrios (*outliers*).

Para efetuar essas operações, este módulo e o módulo **SLAM** são fortemente interdependentes. Esta reciprocidade fica clara em face dos seguintes fatos:

- Para realizar a correspondência entre hipóteses (expressas no sistema de coordenadas da cena) e pontos salientes (expressos no sistema de coordenadas da imagem), é necessário que se disponha de uma estimativa *a priori* para a pose da câmera. Essa estimativa é fornecida pelo módulo **SLAM** — mais especificamente, pelo resultado da etapa de predição da filtragem de Kalman que constitui o cerne do processo de estimação (que será discutido posteriormente na [Seção 5.3](#)). Essa estimativa será chamada de  $\mathbf{r}_t^-$ ;
- Para a etapa de correção da filtragem de Kalman no módulo **SLAM**, o modelo de observação é baseado na projeção das hipóteses associadas aos pontos salientes da imagem corrente. Essa associação é produzida pelo módulo de manutenção de marcos.

### 5.2.1 Marcos resolvidos reobserváveis

A fim de reduzir os custos computacionais durante o processo de associação entre marcos resolvidos e pontos salientes, o módulo de manutenção de marcos leva em consideração somente o subconjunto de marcos resolvidos que possuem probabilidade significativa de serem reobservados pela câmera em sua pose correntemente estimada. Tais marcos são chamados de *marcos resolvidos reobserváveis* e obedecem simultaneamente aos seguintes critérios:

1. Um marco resolvido  $m$  deve estar dentro do espaço observável da câmera (ou seja, o marco deve estar enquadrado pela câmera). Lembrando que a posição de um marco resolvido  $m$  é representado pela posição de sua única hipótese,  $\mathbf{h}_t^{m,1}$ , temos:

$$\text{O marco } m \text{ é reobservável} \quad \Rightarrow \quad \mathbf{h}_t^{m,1} \in V(\mathbf{r}_t^-); \quad (5.7)$$

2. O ponto de vista atual (representado pela estimação *a priori* da pose da câmera,  $\mathbf{r}_t^-$ ) não pode ser significativamente diferente dos pontos de vista anteriores nos quais o marco foi previamente observado. Isto é consequência

do fato de em geral os algoritmos de descritores de características visuais (SIFT, SURF, etc.) não são robustos a mudanças significativas do ponto de vista (isto é, do ângulo de observação da cena), por causa dos efeitos de distorção perspectiva.<sup>d</sup>

Para isto, o módulo de manutenção de marcos mantém um vetor associado a cada marco  $m$ , chamado de *vetor de reobservação*, ou  $\vec{\mathbf{r}}^m$ . Este vetor (não necessariamente unitário) representa a direção de um ponto de vista que, se adotado por uma câmera, indica uma alta probabilidade de reobservação do respectivo marco. Formalmente:

$$\text{O marco } m \text{ é reobservável} \Rightarrow \text{dirc}(\mathbf{r}_t^-, \mathbf{h}_t^{m,1}) \cdot \frac{\vec{\mathbf{r}}^m}{\|\vec{\mathbf{r}}^m\|} \geq \cos(\tau_{\text{vis}}), \quad (5.8)$$

onde  $\tau_{\text{vis}}$  representa o limite máximo de diferença angular além do qual a probabilidade de correspondência é considerada insuficiente para o processo de correspondência de pontos salientes.

Os mecanismos de inicialização e atualização dos vetores de reobservação serão apresentados posteriormente.

Juntando as restrições apresentadas nas Eqs. (5.7) e (5.8), a definição formal de marco resolvido reobservável é vista a seguir:

$$\begin{aligned} \text{Marco } m \text{ é reobservável} \Leftrightarrow & H_t^m = 1 \wedge \\ & \wedge \mathbf{h}_t^{m,1} \in V(\mathbf{r}_t^-) \wedge \\ & \wedge \text{dirc}(\mathbf{r}_t^-, \mathbf{h}_t^{m,1}) \cdot \frac{\vec{\mathbf{r}}^m}{\|\vec{\mathbf{r}}^m\|} \geq \cos(\tau_{\text{vis}}). \end{aligned} \quad (5.9)$$

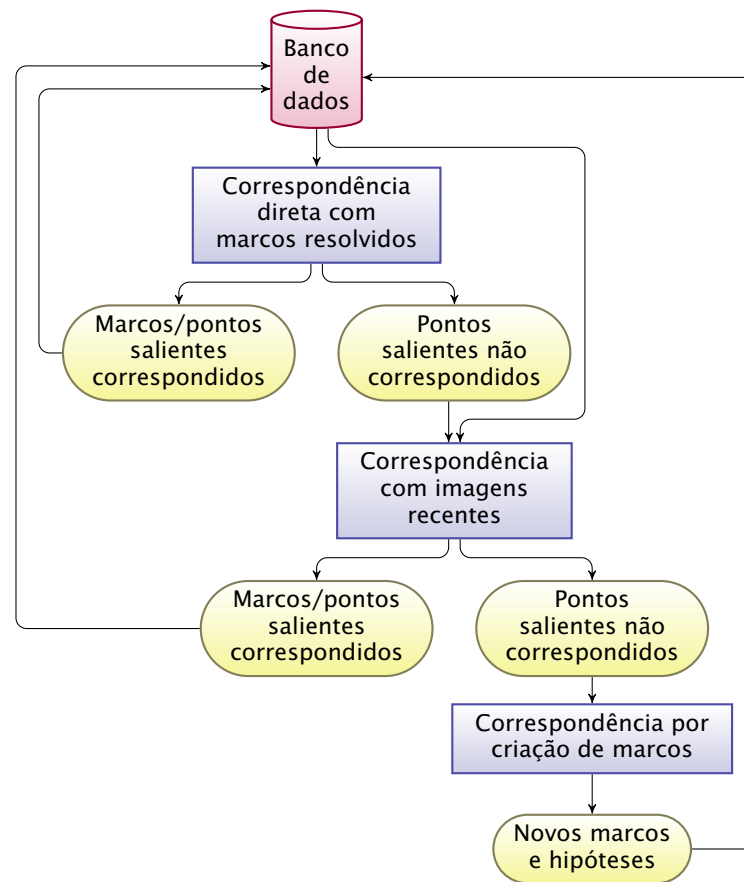
### 5.2.2 Associação entre pontos salientes e marcos

A cada novo conjunto de pontos salientes  $\mathcal{F}_t = \{F_t^1, \dots, F_t^N\}$ , a pergunta primordial a ser respondida por este módulo é: “A qual marco  $m$  corresponde cada ponto saliente  $F_t^f$ ?” O objetivo é tentar estabelecer uma associação biunívoca entre cada ponto saliente  $F_t^f$  e algum marco  $m$ .

Há três formas de se buscar essa associação, representadas pelos seguintes passos de execução deste módulo e ilustradas na [Figura 5.3](#):

1. *Associação por correspondência direta com um marco resolvido do banco de dados*: Dada uma estimativa *a priori* para a pose da câmera,  $\mathbf{r}_t^-$ , projetam-

<sup>d</sup>Morel & Yu [2009] demonstram experimentalmente que o SIFT pode falhar completamente com mudanças de ponto de vista a partir de de 45°.



**Figura 5.3.** As três formas de associação entre pontos salientes e marcos: por correspondência direta com marcos resolvidos extraídos do banco de dados; por correspondência com marcos (resolvidos ou não) associados a pontos salientes de imagens recentes; e por criação de novos marcos.

-se os marcos resolvidos reobserváveis no plano de imagem e realiza-se a correspondência baseada em descritores, segundo a Eq. (4.2), com  $\mathcal{F}_t$ .

Nota-se que a correspondência não é feita entre duas imagens, e sim entre a imagem  $\mathcal{I}_t$  e uma “imagem virtual” formada pelas projeções dos marcos resolvidos reobserváveis. Tecnicamente não há problemas em adotar uma imagem virtual para realizar o procedimento de correspondência. No entanto, uma vez que esta imagem não é formada por pixels, é impossível utilizar algum procedimento para calcular os descritores desses “pontos salientes” da imagem virtual. Este problema é resolvido da seguinte maneira:

- Sempre que um marco  $m$  for criado, armazena-se no banco de dados o descritor  $\mathbf{d}_t^f$  do ponto saliente que gerou o marco (como será visto no Passo 3, um marco é sempre criado a partir de algum ponto saliente). O descritor associado a um marco é chamado de  $\mathbf{d}^m$ ;

- Sempre que um marco  $m$  for reobservado (por este Passo 1 ou pelo Passo 2, apresentado a seguir), o descritor da imagem corrente que foi associado a esse marco substitui o valor corrente de  $\mathbf{d}^m$ .

Desta forma, o banco de dados armazena o vetor descritor mais recente  $\mathbf{d}^m$  de qualquer marco  $m$ . São esses os vetores descritores utilizados na correspondência realizada por meio da Eq. (4.2).

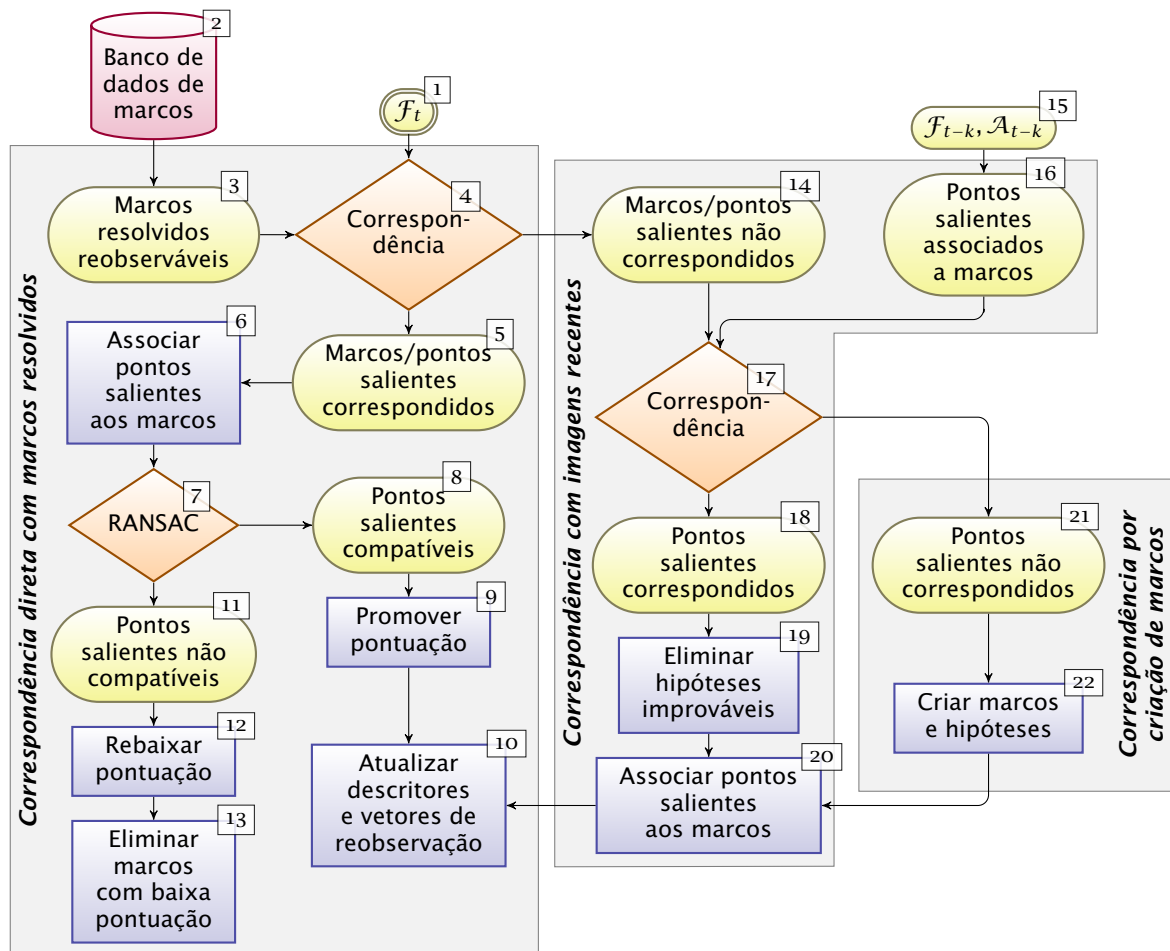
No entanto, essa correspondência não leva em consideração qualquer consistência geométrica, portanto é possível que falsos positivos contaminem o processo de estimação da geometria da cena. Para resolver este problema, a consistência geométrica é realizada utilizando-se **RANSAC**, com base na homografia entre as coordenadas dos pontos salientes da imagem de entrada e as coordenadas projetadas dos marcos na imagem virtual. Um sistema de pontuação de marcos regula a aprovação ou rejeição final dos marcos (o que ocorre apenas após várias observações sucessivas): Os marcos cujas projeções forem consistentes com a homografia ganham um ponto, enquanto os demais perdem um ponto; os que atingirem um limite inferior de pontos são classificados como espúrios e excluídos do banco de dados.

A pontuação associada a um marco  $m$  é designada por  $s^m$ . O sistema de pontuações será detalhado posteriormente;

2. *Associação por correspondência com uma imagem recente:* Os pontos salientes que ainda não foram associados a algum marco são correspondidos com a imagem anterior,  $\mathcal{I}_{t-1}$ . Para cada par correspondido  $\langle F_t^i, F_{t-1}^j \rangle$ , se o ponto saliente  $F_{t-1}^j$  estiver associado a algum marco  $m$ , então estabelece-se a associação entre  $F_t^i$  e  $m$ . O processo é repetido para um conjunto pequeno de imagens anteriores,  $\mathcal{I}_{t-2}, \dots, \mathcal{I}_{t-K}$  (o valor  $K = 3$  foi adotado durante as experimentações).

Este passo não deve ser considerado meramente um “reaproveitamento” dos pontos salientes não correspondidos no Passo 1. É importante perceber que apenas os marcos *resolvidos* são levados em consideração no passo anterior, portanto este passo é a única forma de estabelecer associações entre pontos salientes e marcos não resolvidos;

3. *Associação por criação de marcos:* Finalmente, considera-se que os pontos salientes que não foram associados nos passos anteriores simplesmente não correspondem a nenhum marco conhecido. Neste caso, para cada ponto saliente não associado, um marco é criado e um número finito de hipóte-



**Figura 5.4.** Fluxograma representativo das operações realizadas no módulo de manutenção de marcos. Os valores numéricos no canto superior direito de cada etapa não representam necessariamente a ordem de execução: servem para referência (entre colchetes) no texto.

ses é lançado ao longo do eixo de projeção do ponto saliente (a semirreta apresentada na Eq. (5.3)).

O processo de criação das hipóteses será posteriormente discutido na Subseção 5.2.4.

### 5.2.3 Execução do módulo de manutenção de marcos

A execução deste módulo será detalhada com o auxílio do fluxograma da Figura 5.4 e de alguns algoritmos para esclarecer as operações mais complexas. As etapas do fluxograma serão referenciadas entre colchetes: por exemplo, a “Etapa [3]” refere-se à classificação pelo número de hipóteses, identificado por [3] na Figura 5.4.

A relação entre este fluxograma e os passos de associação entre pontos

---

```

1 procedure CORRESPONDE_IMAGEM_ATUAL_COM_BANCO_DE_DADOS ( $\mathbf{r}_t^-$ )
2   ▷ Mapeamento entre pontos salientes e marcos
3    $\mathcal{A}_t \leftarrow []$ 
4   ▷ Etapas [2]-[3]
5    $\mathcal{M}_{\text{res}} \leftarrow \text{Seleciona\_Marcos\_Resolvidos\_Observáveis}(\mathbf{r}_t^-)$ 
6   ▷ Etapas [4]-[5]
7    $C_{\text{descri t}} \leftarrow \text{Correspondência\_por\_Descritores}(\mathcal{D}_t, \{\mathbf{d}^m \mid \forall m \in \mathcal{M}_{\text{res}}\})$ 
8   ▷ Etapa [6]
9   for all  $\langle f, m \rangle \in C_{\text{descri t}}$  do
10      $\mathcal{A}_t[f] \leftarrow m$ 
11   end for
12   ▷ Etapa [7]
13    $\{(x^m, y^m)\} \leftarrow \{\text{proj}(\mathbf{r}_t^-, \mathbf{h}_t^{m,1}) \mid \forall \langle f, m \rangle \in C_{\text{descri t}}\}$ 
14    $C_{\text{ransac}} \leftarrow \text{Homografia\_por\_RANSAC}(\{(x_t^f, y_t^f)\}, \{(x^m, y^m)\})$ 
15   ▷ Etapa [8]
16   for all  $\langle f, m \rangle \in C_{\text{ransac}}$  do
17     ▷ Etapa [9]
18      $\text{Incrementa\_Pontuação\_de\_Marco}(m)$ 
19     ▷ Etapa [10]
20      $\text{Atualiza\_Descritor\_e\_Vetor\_de\_Reobservação}(m, f, \mathbf{r}_t^-)$ 
21   end for
22   ▷ Etapa [11]
23   for all  $\langle f, m \rangle \in (C_{\text{descri t}} - C_{\text{ransac}})$  do
24     ▷ Etapas [12]-[13]
25      $\text{Decrementa\_Pontuação\_de\_Marco}(m)$ 
26   end for
27 end procedure

```

---

**Algoritmo 5.1.** Procedimento Corresponde\_Imagem\_Atual\_com\_Banco\_de\_Dados.

salientes e marcos apresentados na Subseção 5.2.2 e na Figura 5.3 é vista a seguir:

1. Associação por correspondência direta com um marco resolvido do banco de dados: Etapas [1]-[13];
2. Associação por correspondência com uma imagem recente: Etapas [14]-[20];
3. Associação por criação de marcos: Etapas [21]-[22].

Para melhor compreender a execução dessas etapas, cada uma delas será discutida algorítmicamente nas subseções seguintes.

### 5.2.3.1 Associação de pontos salientes por correspondência direta com marcos resolvidos

Esta tarefa é representada pelo Algoritmo 5.1, cujos passos são detalhados a seguir:

---

```

1 procedure SELECIONA_MARCOS_RESOLVIDOS_OBSERVÁVEIS ( $\mathbf{r}_t^-$ )
2   ▷ Subconjunto de marcos que obedecem à definição da Eq. (5.9)
3   return  $\{m : [H_t^m = 1] \wedge [\mathbf{h}_t^{m,1} \in V(\mathbf{r}_t^-)] \wedge [\text{dirc}(\mathbf{r}_t^-, \mathbf{h}_t^{m,1}) \cdot \frac{\mathbf{r}_t^m}{\|\mathbf{r}_t^m\|} \geq \cos(\tau_{\text{vis}})]\}$ 
4 end procedure

```

---

**Algoritmo 5.2.** Procedimento Selecciona\_Marcos\_Resolvidos\_Observáveis.

---

```

1 procedure CORRESPONDÊNCIA_POR_DESCRITORES ( $\mathcal{D}_p, \mathcal{D}_q$ )
2   return Correspondência realizada entre  $\mathcal{D}_p$  e  $\mathcal{D}_q$  conforme Eq. (4.2)
3 end procedure

```

---

**Algoritmo 5.3.** Procedimento Correspondência\_por\_Descriptores.

- *Linha 3:* Inicializa o mapeamento  $\mathcal{A}_t$ , que contém a associação entre cada ponto saliente  $f$  e o respectivo marco  $m$ . Inicialmente vazio, este mapeamento é populado à medida que pontos salientes e marcos são posteriormente associados.
- *Linha 5:* Selecciona os marcos resolvidos reobserváveis. O procedimento Selecciona\_Marcos\_Resolvidos\_Observáveis (Algoritmo 5.2) implementa diretamente as restrições definidas na Eq. (5.9);
- *Linha 7:* Realiza a correspondência entre os pontos salientes da imagem corrente,  $\mathcal{I}_t$ , e os marcos resolvidos reobserváveis. Esta correspondência (Algoritmo 5.3) é realizada conforme a Eq. (4.2);
- *Laço das linhas 9-11:* Registra em  $\mathcal{A}_t$  as associações entre pontos salientes e marcos;
- *Linhas 13 e 14:* Projetam as coordenadas dos marcos resolvidos reobserváveis em uma imagem virtual e realiza a correspondência por RANSAC, utilizando o modelo de homografia como modelo de transformação. O resultado é um conjunto de pares  $C_{\text{ransac}} = \{\langle f, m \rangle\}$ , onde  $C_{\text{ransac}} \subseteq C_{\text{descri}};$
- *Laço das linhas 16-21:* Processa os marcos que foram considerados *inliners* durante a classificação por RANSAC. A pontuação desses marcos é incrementada, possivelmente atingindo o limite de pontuação para o qual o marco é considerado definitivo (Linha 18, detalhada no Algoritmo 5.5). Em seguida, os descritores correspondidos são armazenados e o vetor de reobservação é atualizado (Linha 20, detalhada no Algoritmo 5.5);
- *Laço das linhas 23-26:* Processa os marcos que foram considerados *outliers* durante a classificação por RANSAC. Os marcos que ainda não foram consi-

---

```

1 procedure INCREMENTA_PONTUAÇÃO_DE_MARCO ( $m$ )
2   ▷ Etapa [9]
3    $s^m \leftarrow s^m + 1$ 
4   if  $s^m \geq s_{\text{ratif}}$  then
5     ▷ Ratificação do marco
6      $s^m \leftarrow \infty$ 
7   end if
8 end procedure

```

---

**Algoritmo 5.4.** Procedimento Incrementa\_Pontuação\_de\_Marco.

---

```

1 procedure ATUALIZA_DESCRITOR_E_VETOR_DE_REOBSERVAÇÃO ( $m, f, \mathbf{r}_t^-$ )
2   ▷ Etapa [10]
3    $\mathbf{d}^m \leftarrow \mathbf{d}_t^f$ 
4    $\tilde{\mathbf{r}}^m \leftarrow \tilde{\mathbf{r}}^m + \text{diri}[\mathbf{r}_t^-, (x_t^f, y_t^f)]$ 
5 end procedure

```

---

**Algoritmo 5.5.** Procedimento Atualiza\_Descriptor\_e\_Vetor\_de\_Reobservação.

---

```

1 procedure DECREMENTA_PONTUAÇÃO_DE_MARCO ( $m$ )
2   ▷ Etapa [12]
3    $s^m \leftarrow s^m - 1$ 
4   ▷ Etapa [13]
5   if  $s^m \leq s_{\text{corte}}$  then
6     Remove todas as referências ao marco  $m$  do banco de dados
7   end if
8 end procedure

```

---

**Algoritmo 5.6.** Procedimento Decrementa\_Pontuação\_de\_Marco.

derados definitivos (isto é, que atingiram uma pontuação mínima  $s_{\text{ratif}}$ ) têm sua pontuação decrementada, possivelmente atingindo o limite mínimo  $s_{\text{corte}}$  que dispara a exclusão de marcos do banco de dados (Linha 25, detalhada no Algoritmo 5.6).

### 5.2.3.2 Associação de pontos salientes por correspondência com uma imagem recente

Esta tarefa é representada pelo Algoritmo 5.7, cujos passos são detalhados a seguir:

- *Linha 3:* Seleciona o subconjunto de pontos salientes  $\mathcal{F}_{\text{rest}} \subseteq \mathcal{F}_t$  que não foram correspondidos com o banco de dados;



---

```

1 procedure CORRESPONDE_IMAGEM_ATUAL_COM_IMAGEM_RECENTE ( $\mathbf{r}_t^-$ )
2   ▷ Etapa [14]: Conjunto de pontos salientes pendentes
3    $\mathcal{F}_{\text{rest}} \leftarrow \mathcal{F}_t - \{F_t^f \mid \forall \langle f, m \rangle \in C_{\text{descri}}\}$ 
4   ▷ Etapa [15]
5   for all  $u \in \{t-1, t-2, \dots, t-K\}$  do
6     ▷ Etapa [16]
7      $\mathcal{F}_{\text{assoc}} \leftarrow \{F_u^f \in \mathcal{F}_u : f \in \mathcal{A}_u\}$ 
8     ▷ Etapa [17]
9      $C_{\text{geom}} \leftarrow \text{Correspondência\_por\_Consistência\_Geométrica}(\mathcal{F}_{\text{rest}}, \mathcal{F}_{\text{assoc}})$ 
10    ▷ Etapa [18]
11    for all  $\langle f_t, f_u \rangle \in C_{\text{geom}}$  do
12      ▷ Marcos associados ao ponto saliente da imagem anterior
13       $m \leftarrow \mathcal{A}_u[f_u]$ 
14      ▷ Etapa [19]
15      Elimina as hipóteses mais improváveis de  $m$  (Subseção 5.2.5)
16      ▷ Etapa [20]
17       $\mathcal{A}_t[f_t] \leftarrow m$ 
18      ▷ Etapa [10]
19       $\text{Atualiza\_Descritor\_e\_Vetor\_de\_Reobservação}(m, f_t, \mathbf{r}_t^-)$ 
20      ▷ Retira o ponto saliente do conjunto de pontos salientes pendentes
21       $\mathcal{F}_{\text{rest}} \leftarrow \mathcal{F}_{\text{rest}} - \{F_t^{f_t}\}$ 
22    end for
23  end for
24 end procedure

```

---

**Algoritmo 5.7.** Procedimento `Corresponde_Imagem_Atual_com_Imagem_Recente`.

---

```

1 procedure CORRESPONDÊNCIA_POR_CONSISTÊNCIA_GEOMÉTRICA ( $\mathcal{F}_p, \mathcal{F}_q$ )
2   return Correspondência realizada entre  $\mathcal{F}_p$  e  $\mathcal{F}_q$  conforme Eq. (4.12)
3 end procedure

```

---

**Algoritmo 5.8.** Procedimento `Correspondência_por_Consistência_Geométrica`.

- *Laço das linhas 5-23:* Corresponde os pontos salientes em  $\mathcal{I}_t$  sucessivamente com os das imagens  $\mathcal{I}_{t-1}, \mathcal{I}_{t-2}, \dots, \mathcal{I}_{t-K}$ . A cada iteração, os pontos salientes de  $\mathcal{I}_t$  herdam a associação previamente realizada entre marcos e os pontos salientes de  $\mathcal{I}_{t-*}$ ;
- *Linha 7:* Seleciona o subconjunto de pontos salientes de uma imagem anterior  $\mathcal{I}_u, \mathcal{F}_{\text{assoc}} \subseteq \mathcal{F}_u$ , que foram previamente associados a algum marco;
- *Linha 9:* Realiza a correspondência por consistência geométrica (Algoritmo 5.8).

---

```

1 procedure CRIA_MARCOS_E_HIPÓTESES ( $\mathbf{r}_t^-$ )
2   ▷ Etapa [21]
3   for all  $F_t^f \in \mathcal{F}_{\text{rest}}$  do
4     ▷ Etapa [22]
5     Cria um novo marco  $m$  e as respectivas hipóteses (Subseção 5.2.4)
6      $s^m \leftarrow 0$ 
7      $\bar{\mathbf{r}}^m \leftarrow [0 \ 0 \ 0]^\top$ 
8     ▷ Etapa [20]
9      $\mathcal{A}_t[f] \leftarrow m$ 
10    ▷ Etapa [10]
11    Atualiza_Descriptor_e_Vetor_de_Reobservação( $m, f, \mathbf{r}_t^-$ )
12  end for
13 end procedure

```

---

**Algoritmo 5.9.** Procedimento Cria\_Marcos\_e\_Hipóteses.

- *Laço das linhas 11-22:* Itera sobre as correspondências obtidas, realizando as tarefas descritas a seguir;
- *Linha 13:* Recupera os marcos associados aos pontos salientes das imagens anteriores;
- *Linha 15:* Para os marcos não resolvidos, elimina as hipóteses menos prováveis. Os detalhes deste procedimento serão vistos na Subseção 5.2.5;
- *Linha 17:* Copia as associações entre pontos salientes e marcos para os pontos salientes da imagem corrente;
- *Linha 19:* Armazena os descritores correspondidos e atualiza o vetor de reobservação (Algoritmo 5.5);
- *Linha 21:* Impede que o pontos salientes correspondidos sejam levados em consideração na próxima iteração do laço principal do algoritmo.

### 5.2.3.3 Associação de pontos salientes por criação de marcos

Esta tarefa é representada pelo Algoritmo 5.9, cujos passos são detalhados a seguir:

- *Laço das linhas 3-12:* Itera sobre os pontos salientes restantes, a fim de criar um marco para cada;
- *Linha 5:* Avalia as coordenadas e as incertezas de todas as hipóteses associadas ao novo marco e lança-os no banco de dados. Como esta operação possui certa complexidade, ela será descrita com detalhes na Subseção 5.2.4;

- *Linhas 6 e 7*: Inicializa a pontuação e o vetor de reobservação do marco. O vetor de reobservação será atualizado na etapa seguinte;
- *Linha 11*: Armazena os descritores correspondidos e atualiza o vetor de reobservação (*Algoritmo 5.5*).

#### 5.2.4 Criação de marcos e hipóteses

Quando um ponto saliente é detectado em uma imagem e não tiver sido associado a algum marco nos passos anteriores deste módulo, considera-se que ele corresponde a um marco desconhecido. Assim, um novo marco é inserido no banco de dados e um conjunto de hipóteses é criado para esse marco.

Naturalmente, o problema imediato é estabelecer os valores iniciais para as posições dessas hipóteses e as respectivas incertezas. Para isto, é imprescindível dispor do modelo ideal da PDF da posição de um marco para então discutir a abordagem utilizada.

Se a única informação disponível sobre a posição  $\mathbf{m}_t^m$  de um marco  $m$  é a observação do respectivo ponto saliente, então a PDF tridimensional é assim definida:

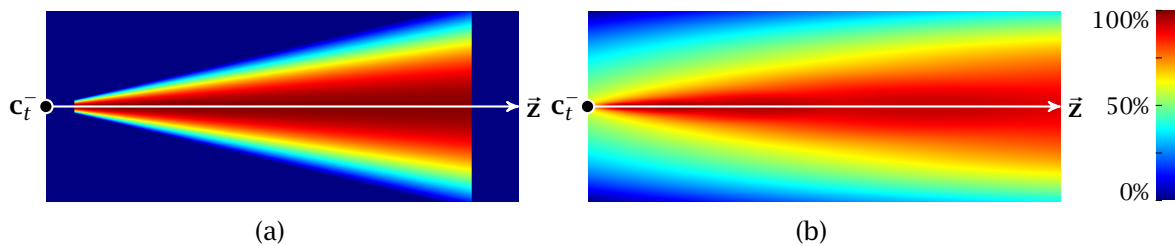
- No que diz respeito à profundidade do marco (isto é, à sua distância a partir do centro de projeção da câmera), a distribuição de probabilidades é uniforme e limitada pela profundidade de campo, isto é, pelos valores  $d_{\min}$  e  $d_{\max}$  (*Seção 5.1*). Esta incerteza se estende ao longo do eixo de projeção do ponto saliente (Eq. (5.3));
- Quanto à distribuição de probabilidades ortogonal ao eixo de projeção, esta é igual à incerteza de localização do ponto saliente na imagem, convertida em unidades da cena e proporcional à profundidade.

A PDF assim definida, representada por  $\mathcal{P}(\mathbf{m}_t^m)$ , forma um tronco de cone, cuja seção longitudinal é ilustrada na Figura 5.5(a).

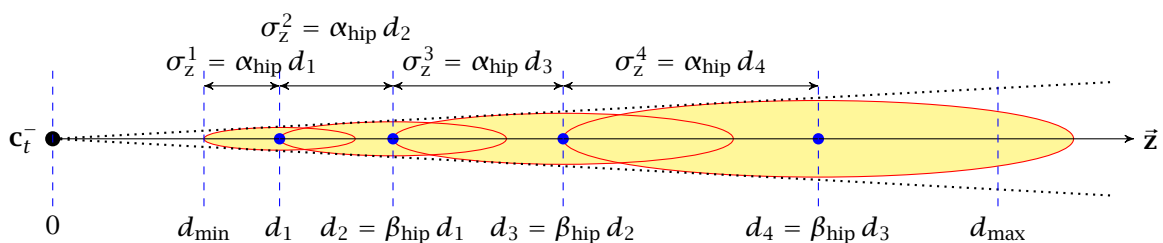
No entanto, essa distribuição ideal não pode ser representada por uma PDF normal (ou mesmo por uma combinação linear de PDFs normais) — um pré-requisito para a adoção de *Mapeamento Estocástico (Stochastic Mapping, ou SM)*. Desta forma, neste trabalho a PDF ideal é aproximada por uma SoG, com base em uma variação da estratégia proposta por Solà et al. [2005, 2008].

Em uma SoG, a distribuição  $\mathcal{P}(\mathbf{m}_t^m)$  é aproximada pelo seguinte somatório:

$$\mathcal{P}(\mathbf{m}_t^m) \approx \sum_h w^h \mathcal{P}(\mathbf{h}_t^{m,h}), \quad (5.10)$$



**Figura 5.5.** Exemplos das PDFs para a localização de um marco no instante de sua criação. O eixo  $\bar{z}$  é a direção das coordenadas do ponto saliente  $f$  que disparou a criação das hipóteses, ou seja,  $\bar{z} = \text{diri}(\mathbf{r}_t^-, (x_t^f, y_t^f))$ . A escala de cores é relativa ao valor máximo da verossimilhança. **(a)** Distribuição ideal (uniforme ao longo do eixo de projeção e proporcional à distância nos demais eixos); **(b)** Distribuição aproximada pela Soma de Gaussianas (SoG) proposta.



**Figura 5.6.** Posição das hipóteses recém-criadas (a partir do centro de projeção da câmera) e respectivas incertezas ao longo do eixo de projeção. O eixo  $\bar{z}$  é a direção das coordenadas do ponto saliente  $f$  que disparou a criação das hipóteses, ou seja,  $\bar{z} = \text{diri}(\mathbf{r}_t^-, (x_t^f, y_t^f))$ . (Adaptado de Solà et al. [2005])

onde  $\mathbf{h}_t^{m,h}$  é a estimativa corrente da  $h$ -ésima hipótese do marco  $m$  e  $w^h$  é o respectivo peso na SoG, inicialmente fixado em

$$w^h = \frac{1}{H_{\text{inic}}}, \quad (5.11)$$

e  $H_{\text{inic}}$  é o número de hipóteses. Como a distribuição de probabilidades de cada hipótese é normal, tem-se

$$\mathcal{P}(\mathbf{h}_t^{m,h}) \sim \mathcal{N}(\hat{\mathbf{h}}_t^{m,h}, \mathbf{H}_t^{m,h}), \quad (5.12)$$

onde  $\hat{\mathbf{h}}_t^{m,h}$  e  $\mathbf{H}_t^{m,h}$  são os dois primeiros momentos (média e covariância) da distribuição de incertezas da hipótese  $\mathbf{h}_t^{m,h}$ .

Na estratégia de inicialização original de Solà et al., as médias  $\hat{\mathbf{h}}_t^{m,h}$  são posicionadas sobre o eixo de projeção com espaçamento progressivamente maior, formando uma progressão geométrica de distâncias a partir do centro de projeção

(Figura 5.6). Formalmente:

$$d_h \triangleq \begin{cases} \frac{d_{\min}}{1 - \alpha_{\text{hip}}} & \text{para } h = 1 \\ \beta_{\text{hip}} d_{h-1} = (\beta_{\text{hip}})^{h-1} d_1 & \text{para } h > 1, \end{cases} \quad (5.13)$$

onde  $\beta_{\text{hip}}$  é a *razão geométrica de distribuição das hipóteses*, que determina o espaçamento inicial entre as hipóteses; e  $\alpha_{\text{hip}}$  é a *razão de incerteza das hipóteses*, que serve de base para determinar a incerteza de cada hipótese na direção do eixo de projeção. O número de hipóteses,  $H_{\text{inic}}$ , é fixado de modo a que a última hipótese não ultrapasse a distância  $d_{\max}$ , de onde:

$$H_{\text{inic}} \triangleq \left\lfloor \frac{\log d_{\max} - \log(d_{\min} + \alpha_{\text{hip}})}{\log \beta_{\text{hip}}} \right\rfloor + 1. \quad (5.14)$$

As matrizes de covariância,  $\mathbf{H}_t^{m,h}$ , são construídas de modo a formar elipsoides alongadas no sentido do eixo de projeção (ver Figura 5.6).<sup>e</sup> O desvio-padrão ao longo do eixo de projeção,  $\sigma_z^h$ , é assim calculado:

$$\sigma_z^h \triangleq \alpha_{\text{hip}} d_h \quad (5.15)$$

e o desvio-padrão ortogonal ao eixo de projeção é definido por:

$$\sigma_{xy}^h \triangleq \sigma_{\text{ps}} \frac{d_h}{z_c}, \quad (5.16)$$

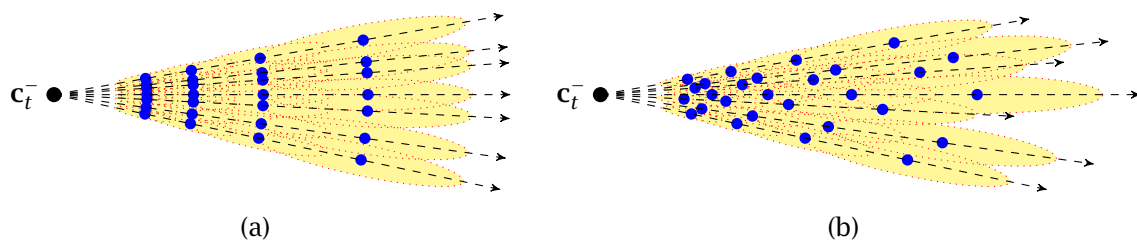
onde  $z_c$  é a distância focal da câmera e  $\sigma_{\text{ps}}$  é o desvio-padrão que representa a incerteza de localização dos pontos salientes na imagem.

No entanto, a formulação original de Solà et al. concentra todas as hipóteses criadas em um conjunto preestabelecido de cascas esféricas concêntricas (Figura (5.7(a))). Essa característica gera uma tendência estatística que foi percebida na fase inicial das experimentações.

Para combater essa tendência, o cálculo das distâncias,  $d_h$ , foi alterado nesta

---

<sup>e</sup>O aspecto alongado é consequência natural do fato de que a incerteza na direção da profundidade é maior do que a dos eixos ortogonais (estes resultados da incerteza de localização do ponto saliente na imagem).



**Figura 5.7.** Análise da tendência de privilégio de distâncias entre a câmera e a cena: **(a)** Na formulação original de Solà et al. [2005, 2008], as distâncias da câmera às hipóteses são fixas, gerando regiões com concentração de hipóteses; **(b)** Na formulação adotada, as profundidades são alteradas por um fator aleatório, evitando a formação de zonas privilegiadas.

tese: As distâncias originais são multiplicadas por um fator aleatório exponencial:

$$d_h \triangleq \begin{cases} \frac{d_{\min}}{1 - \alpha_{\text{hip}}} 2^{(u-1/2)} & \text{para } h = 1 \\ \beta_{\text{hip}} d_{h-1} = (\beta_{\text{hip}})^{h-1} d_1 & \text{para } h > 1, \end{cases} \quad (5.17)$$

onde  $u$  é um valor aleatório com distribuição uniforme no intervalo  $u \in [0, 1)$ . Com esta nova definição, as hipóteses recém-criadas não tendem a formar grupos em distâncias predefinidas, evitando o erro sistemático descrito acima (Figura 5.7(b)). O número de hipóteses,  $H_{\text{inic}}$ , permanece conforme a definição apresentada na Eq. (5.14).

### 5.2.5 Avaliação e descarte de hipóteses

Sempre que um marco não resolvido é observado, as suas hipóteses são avaliadas para que pelo menos uma seja descartada. Com isso, garante-se a convergência de marcos recém-criados em marcos resolvidos em no máximo  $H_{\text{inic}} - 1$  observações.

Para compreender o funcionamento da avaliação das hipóteses, tome-se o evento em que um ponto saliente  $F_t^f$  da imagem corrente é associado a um marco não resolvido  $m$  (Algoritmo 5.7, linha 13). As coordenadas do ponto saliente,  $(x_t^f, y_t^f)$ , representam uma evidência — uma observação — da verdadeira posição tridimensional do marco. Com base nas estimativas correntes das posições tridimensionais das hipóteses ( $\hat{\mathbf{h}}_t^{m,h}$ ) e respectivas incertezas ( $\mathbf{H}_t^{m,h}$ ), é possível determinar a verossimilhança de cada hipótese em relação à observação.

No entanto, o cálculo das verossimilhanças não pode ser feito de maneira direta: Note-se que a observação é realizada no espaço bidimensional, porém as hipóteses são descritas no espaço tridimensional. Com isso, é necessário primeiro

projetar a PDF de cada hipótese sobre o plano de imagem. Como a transformação projetiva não é linear, optou-se pelo uso da *Transformação Unscented (Unscented Transform, ou UT)* (Subseção A.3.1) para realizar essa projeção. Assim, a Eq. (A.14) é reescrita a seguir, instanciada para este caso em particular:

$$\mathcal{N}(\hat{\mathbf{h}}_t^{m,h}, \mathbf{H}_t^{m,h}) \xrightarrow{\text{UT} [\text{proj}(\cdot), \mathbf{P}_{\text{proj}}]} \mathcal{N}(\hat{\mathbf{h}}_{\text{im}}^{m,h}, \mathbf{H}_{\text{im}}^{m,h}), \quad (5.18)$$

onde  $\mathbf{P}_{\text{proj}}$  é a matriz de covariâncias que representa as incertezas (ruídos) do processo projetivo e  $\mathcal{N}(\hat{\mathbf{h}}_{\text{im}}^{m,h}, \mathbf{H}_{\text{im}}^{m,h})$  é a PDF normal (no espaço de imagem) obtida após a transformação.

A verossimilhança de cada hipótese  $h$  no espaço de imagem, designada por  $l_{\text{im}}^{m,h}$ , é obtida pela avaliação da curva normal nas coordenadas  $(x_t^f, y_t^f)$  do ponto saliente  $f$ :

$$l_{\text{im}}^{m,h} \triangleq \frac{1}{|2\pi\mathbf{H}_{\text{im}}^{m,h}|^{1/2}} \exp \left[ -\frac{1}{2} \left( \begin{bmatrix} x_t^f \\ y_t^f \end{bmatrix} - \hat{\mathbf{h}}_{\text{im}}^{m,h} \right)^\top (\mathbf{H}_{\text{im}}^{m,h})^{-1} \left( \begin{bmatrix} x_t^f \\ y_t^f \end{bmatrix} - \hat{\mathbf{h}}_{\text{im}}^{m,h} \right) \right], \quad (5.19)$$

de onde a probabilidade de observação de cada hipótese,  $\mathcal{P}(\mathbf{h}_t^{m,h})$ , pode ser trivialmente calculada pela normalização das verossimilhanças:

$$\mathcal{P}(\mathbf{h}_t^{m,h}) \triangleq \frac{1}{\sum_{h'} l_{\text{im}}^{m,h'}} l_{\text{im}}^{m,h}. \quad (5.20)$$

Finalmente, são descartadas todas as hipóteses  $h$  cujas probabilidades estão abaixo de um limiar  $\tau_{\text{hip}}$ , ou seja:

$$\mathcal{P}(\mathbf{h}_t^{m,h}) < \tau_{\text{hip}} \quad \Rightarrow \quad \text{Hipótese } h \text{ é descartada.} \quad (5.21)$$

O limiar  $\tau_{\text{hip}}$  representa aproximadamente a probabilidade de se descartar a hipótese verdadeira [Solà et al., 2005]. Neste trabalho, adota-se  $\tau_{\text{hip}} = 0,1\%$ . Caso nenhuma hipótese obedeça à condição da Eq. (5.21), então a hipótese de menor probabilidade é descartada.

### 5.2.6 Descarte de marcos não observados

Conforme se pode observar sobre o procedimento apresentado na subseção anterior, a promoção de marcos não resolvidos a resolvidos somente ocorre se o mesmo for observado diversas vezes. A manutenção de marcos não resolvidos

representa um custo computacional significativamente maior do que a manutenção de marcos resolvidos, pelo simples fato de que o sistema deve manter e atualizar as PDFs de cada hipótese do marco. Assim, é interessante manter o menor número possível de marcos não resolvidos no sistema.

No entanto, sempre há a possibilidade de que o número de observações de um marco não seja suficiente para promovê-lo a resolvido. Por exemplo, ele pode sair do enquadramento da câmera. Ainda pior é o caso de que pontos salientes espúrios (causados por ruídos) gerem marcos que não voltarão a ser observados. É importante lembrar que marcos não resolvidos não contribuem para o objetivo final deste trabalho — a estimação da geometria dos objetos da cena; portanto, é importante que se disponha de um mecanismo para o descarte de marcos que apresentam baixa probabilidade de se tornarem resolvidos.

Este problema está implicitamente resolvido no processo de associação de marcos e hipóteses, descrito anteriormente (Subseção 5.2.3). De fato, percebe-se que a única forma possível para associar um marco conhecido não resolvido a um ponto saliente está descrito na Subsubseção 5.2.3.2, que trata da correspondência entre pontos salientes da imagem atual,  $\mathcal{I}_t$ , e uma recente,  $\mathcal{I}_{t-k}$ , para  $k \in \{1, \dots, K\}$  e um limite preestabelecido  $K$  de imagens recentes. Como essas correspondências são limitadas apenas às  $K$  últimas imagens, assume-se que os marcos que não se tornarem resolvidos nessa janela de tempo não poderão jamais ser reobservados, e portanto devem ser descartados.

### 5.2.7 Detecção e eliminação de marcos espúrios

A promoção de um marco não resolvido a resolvido não garante que a hipótese sobrevivente realmente corresponda a um ponto na superfície de um objeto da cena. De fato, mesmo os algoritmos robustos de correspondência de pontos salientes entre imagens, como o apresentado no Capítulo 4, não garantem a remoção de 100% de falsas correspondências. No processo de estimação de estados deste trabalho, as falsas correspondências geram dois efeitos indesejados: podem causar a eliminação das melhores hipóteses e podem corromper criticamente a estimação da posição das hipóteses restantes — gerando, portanto, um marco cuja posição é geometricamente inconsistente com a cena.

Para forçar a consistência geométrica dos marcos, um algoritmo simples foi desenvolvido. Cada marco resolvido  $m$  possui uma pontuação associada, designada por  $s^m$  e inicializada com o valor 0 (Algoritmo 5.9, linha 6). Durante o procedimento de correspondência entre a imagem atual e o banco de dados



(Algoritmo 5.1), busca-se uma homografia entre os pontos salientes da imagem e os falsos pontos salientes projetados utilizando-se o algoritmo *Random Sample Consensus* (RANSAC) [Fischler & Bolles, 1981] (Algoritmo 5.1, linha 14)<sup>f</sup>. Para cada marco  $m$ , um dos dois predicados é verdadeiro:

- O marco  $m$  é aceito pelo RANSAC, indicando consistência geométrica com a cena: Neste caso, a sua pontuação é incrementada (Algoritmo 5.4), possivelmente atingindo um limite de pontuação de ratificação,  $s_{\text{ratif}}$ , que o promove a *marco definitivo* (pontuação infinita). Os marcos definitivos são aqueles que não podem mais ser descartados e compõem a nuvem de pontos do objeto reconstruído;
- O marco  $m$  é rejeitado pelo RANSAC, indicando inconsistência geométrica: Neste caso, a sua pontuação é decrementada (Algoritmo 5.6), possivelmente atingindo um limite de pontuação de corte,  $s_{\text{corte}}$ , que causa a sua eliminação do banco de dados.

A adoção do sistema de pontuações evita que falsos positivos ou falsos negativos gerados pelo RANSAC causem a promoção ou eliminação imediata de um marco.

### 5.3 O módulo SLAM

Conforme apresentado no início deste capítulo, o objetivo principal do módulo SLAM é estimar continuamente a pose da câmera do robô,  $\mathbf{r}_t$ , e a posição de todas as hipóteses dos marcos,  $\mathbf{h}_t^{m,h}$ . Estocasticamente, essas informações são representadas pelas PDFs normais multivariadas  $\mathcal{N}(\hat{\mathbf{r}}_t, \mathbf{R}_t)$  e  $\mathcal{N}(\hat{\mathbf{h}}_t^{m,h}, \mathbf{H}_t^{m,h})$ , respectivamente.

O processo de estimação adotado neste trabalho é baseado em *Mapeamento Estocástico* (*Stochastic Mapping*, ou SM), com algumas adaptações para a adequação ao cenário multi-hipotético proposto. Em sua forma original, apresentada anteriormente na Seção 2.1, o SM condensa a estimação da localização e dos pontos da cena em uma única PDF normal que compreende todas as variáveis estimadas. A Eq. (2.1) apresenta a forma geral do vetor que condensa todas as variáveis estimadas. No entanto, ela terá que ser adaptada para refletir o fato de que a posição individual

<sup>f</sup>Embora a transformação homográfica não modele as distorções perspectivas causadas pela observação da cena em dois pontos de vista distintos, considera-se que a diferença entre a pose da câmera *a priori*,  $\mathbf{r}_t^-$ , e a pose real seja suficientemente pequena para que tais distorções sejam desprezadas.

das hipóteses é que terá que ser estimada, não mais a posição dos marcos<sup>8</sup>:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{r}_t \\ \mathbf{h}_t^{1,1} \\ \vdots \\ \mathbf{h}_t^{M_t, H_t^{M_t}} \end{bmatrix}. \quad (5.22)$$

Por consequência, as Eqs. (2.2a) e (2.2b), que apresentam os termos da PDF normal que descreve todo o estado do sistema, também é apresentada com adaptações:

$$\hat{\mathbf{x}}_t = \begin{bmatrix} \hat{\mathbf{r}}_t \\ \hat{\mathbf{h}}_t^{1,1} \\ \vdots \\ \hat{\mathbf{h}}_t^{M_t, H_t^{M_t}} \end{bmatrix} \quad \text{e} \quad (5.23a)$$

$$\mathbf{X}_t = \begin{bmatrix} \mathbf{R}_t & \mathbf{P}_t^{r;1,1} & \dots & \mathbf{P}_t^{r;M_t, H_t^{M_t}} \\ \mathbf{P}_t^{r;1,1} & \mathbf{H}_t^{1,1} & \dots & \mathbf{H}_t^{1,1;M_t, H_t^{M_t}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_t^{r;M_t, H_t^{M_t}} & \mathbf{H}_t^{M_t, H_t^{M_t};1,1} & \dots & \mathbf{H}_t^{M_t, H_t^{M_t}} \end{bmatrix}, \quad (5.23b)$$

onde  $\mathbf{R}_t$  é a matriz de covariâncias da pose da câmera;  $\mathbf{H}_t^{m,i}$  é a covariância da hipótese  $\mathbf{h}_t^{m,i}$ ;  $\mathbf{H}_t^{m,i;n,j}$  é a covariância cruzada entre a posição das hipóteses  $\mathbf{h}_t^{m,i}$  e  $\mathbf{h}_t^{n,j}$ ; e  $\mathbf{P}_t^{r;m,i}$  é a covariância cruzada entre a pose da câmera e a da hipótese  $\mathbf{h}_t^{m,i}$ .

### 5.3.1 Estimadores de estados

Um interessante subproduto do SM é que as covariâncias entre todas as variáveis de estado (os elementos fora da diagonal de  $\mathbf{X}_t$ ) também são estimadas. Em geral, esse é um efeito desejado, já que tais covariâncias são relevantes para o processo [Leonard & Durrant-Whyte, 1991]. Entretanto, para a presente abordagem multi-hipotética, o SM original apresenta duas grandes desvantagens:

- Apenas uma hipótese sobrevive entre as  $H_{\text{inic}}$  hipóteses criadas para cada marco. Portanto, a covariância cruzada entre as hipóteses de um mesmo marco não só é inútil, como gera custo computacional para armazenamento e avaliação desnecessários;

<sup>8</sup>Ressalta-se que a posição de um marco não é um conceito definido para marcos não resolvidos (Subseção 3.3.3).

- A inclusão de todas as hipóteses no **SM** gera um efeito danoso: a cada passo, o estimador busca um consenso entre as observações disponíveis e as variáveis de estado. Por conseguinte, as piores hipóteses (as que estão mais distantes da posição real do respectivo marco) induzem o estimador a buscar uma solução consensual que na prática pode contaminar a estimação das melhores hipóteses, causando até mesmo o eventual descarte destas. Este efeito foi observado durante a fase inicial de implementação.

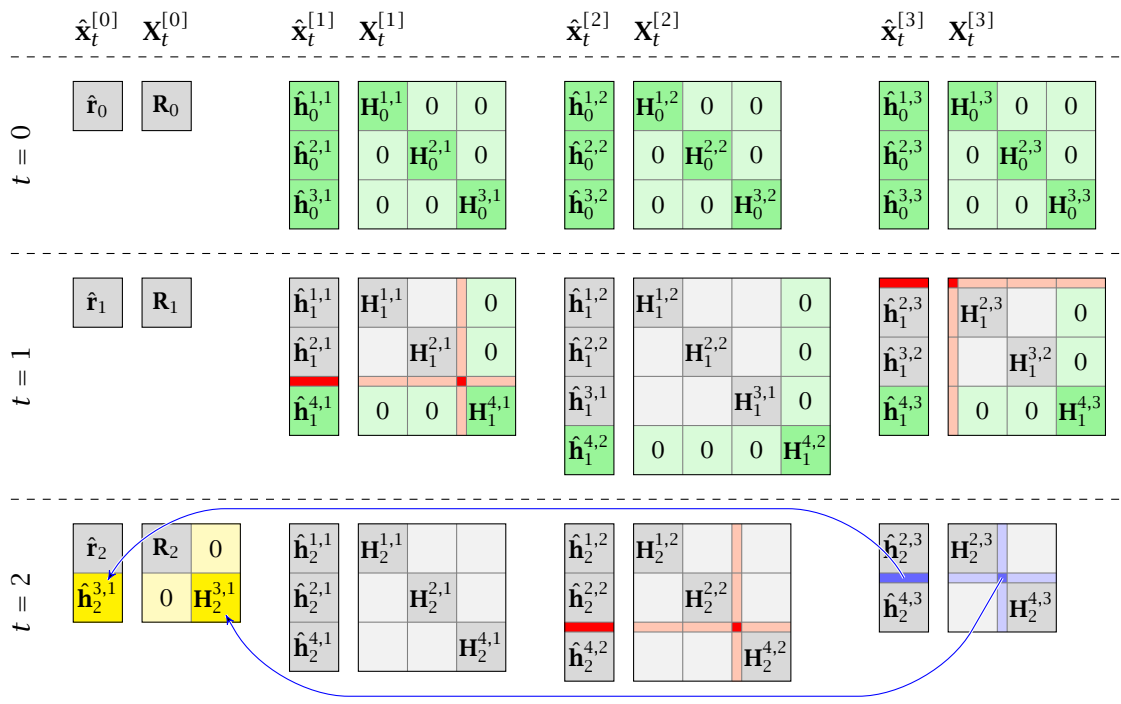
Para contornar ambos os problemas, o estado apresentado na Eq. (5.22) será dividido em um conjunto de estimadores independentes assim definidos:

- o *estimador principal* é responsável pela estimação conjunta da pose da câmera do robô e das posições dos marcos resolvidos. A PDF do estimador principal será representada por  $\mathcal{N}(\hat{\mathbf{x}}_t^{[0]}, \mathbf{X}_t^{[0]})$ ;
- os *estimadores de hipóteses* são responsáveis pela estimação das posições das hipóteses dos marcos não resolvidos. A PDF do  $p$ -ésimo estimador de hipóteses será representado por  $\mathcal{N}(\hat{\mathbf{x}}_t^{[p]}, \mathbf{X}_t^{[p]})$ .

No momento da criação de um marco, as suas respectivas hipóteses são acrescentadas aos estimadores de hipóteses — uma hipótese para cada estimador, de modo que há  $H_{\text{inic}}$  estimadores de hipóteses no total. Em outras palavras, cada hipótese  $\mathbf{h}_t^{m,h}$  de um novo marco  $m$  é acrescentada ao  $h$ -ésimo estimador de hipóteses.

As operações subsequentes de descarte de hipóteses são conduzidas pela remoção dos elementos correspondentes de  $\hat{\mathbf{x}}_t^{[p]}$  e  $\mathbf{X}_t^{[p]}$ . Sempre que um marco se tornar resolvido — ou seja, quando lhe restar uma única hipótese —, as suas médias e covariâncias são transferidas do estimador de hipóteses para o estimador principal. Essa dinâmica de criação, eliminação e transferência de hipóteses entre os estimadores é ilustrada na [Figura 5.8](#).

Observe que esta separação em vários estimadores resolve simultaneamente os dois problemas descritos anteriormente: (i) a covariância entre hipóteses de um mesmo marco não é avaliada, pois essas estão distribuídas entre os diversos estimadores de hipóteses; e (ii) nenhuma hipótese de um marco não resolvido influencia na estimação da pose da câmera, pois esta é tratada pelo estimador principal, para onde são transferidas somente as hipóteses de marcos resolvidos.



**Figura 5.8.** Exemplo ilustrativo da evolução das PDFs dos estimadores diante da criação e descarte de hipóteses, com  $H_{\text{inic}} = 3$  (estimador principal + 3 estimadores de hipóteses). **Primeira linha** ( $t = 0$ ): Criação de três marcos,  $m = 1, \dots, 3$ . As hipóteses recém-criadas,  $(\mathbf{h}_0^{1,*}, \mathbf{h}_0^{2,*}$  e  $\mathbf{h}_0^{3,*})$  são distribuídas entre os estimadores de hipóteses. Note que as hipóteses mais próximas,  $\mathbf{h}_0^{*,1}$ , são acrescentadas ao primeiro estimador de hipóteses; no estimador seguinte, as hipóteses imediatamente mais distantes ( $\mathbf{h}_0^{*,2}$ ); e assim por diante. **Segunda linha** ( $t = 1$ ): Eliminação de duas hipóteses, afetando o primeiro e o terceiro estimador de hipóteses, e criação do marco  $m = 4$  (cujas hipóteses são inseridas nas últimas linhas/colunas de  $\hat{\mathbf{x}}_1^{[p>0]}$  e  $\mathbf{X}_1^{[p>0]}$ ). **Terceira linha** ( $t = 1$ ): Eliminação de outra hipótese do marco  $\mathbf{m}_2^3$ . Como uma única hipótese resta para este marco ( $\mathbf{h}_2^{3,1}$ ), esta é transferida do estimador de hipóteses para o estimador principal.

### 5.3.2 O processo de estimação de estados

A estimação das variáveis do sistema — pose da câmera e posição das hipóteses — é realizada recursivamente por meio de um banco de UKFs, um para cada um dos estimadores descritos na subseção anterior. A modelagem desses UKFs depende do tipo do estimador:

- O estimador principal, responsável pela pose tridimensional da câmera e posição dos marcos resolvidos, é tratada por uma UKF assim modelada:
  - *Modelo de transição de estados:* A pose da câmera é atualizada de acordo com as informações provenientes dos sensores proprioceptivos,

- $\mathbf{u}_t$  (Subseção 3.3.1), enquanto as hipóteses permanecem estacionários durante a transição de estados;
- *Modelo de observação*: Compara as coordenadas dos pontos salientes com as respectivas hipóteses associadas durante a correspondência com o banco de dados (Algoritmo 5.1).
- Os estimadores de hipóteses, responsáveis unicamente pelas posições das hipóteses dos marcos não resolvidos, são tratadas por UKFs assim modeladas:
    - *Modelo de transição de estados*: Como o modelo dinâmico do sistema (deslocamento da câmera) afeta somente o estimador principal, o modelo de transição de estados dos demais estimadores é estático: todas as hipóteses permanecem estacionárias e não há fonte de incertezas durante a etapa de predição;
    - *Modelo de observação*: Compara as coordenadas dos pontos salientes com as respectivas hipóteses associadas durante a correspondência com as imagens recentes (Algoritmo 5.7).

Finalmente, todo o processo integrado de execução dos módulos de manutenção de marcos e de SLAM pode ser resumido na execução cíclica dos seguintes passos, ilustrados no início deste capítulo (Figura 5.1):

1. Predição do estimador principal, com base nas informações dos sensores proprioceptivos;
2. Associação entre pontos salientes e marcos, descarte de hipóteses e criação de marcos e hipóteses (operações a cargo do módulo de manutenção de marcos, detalhadas na Seção 5.2);
3. Execução da etapa de correção do UKF do estimador principal;
4. Execução da etapa de correção dos UKFs dos estimadores de hipóteses.

Note-se que os passos de correção dos vários estimadores podem ser executados em paralelo.



# Capítulo 6

## Planejamento

**E**STE CAPÍTULO TEM POR OBJETIVO detalhar a estrutura dos dois módulos que representam o caráter autônomo deste trabalho, apresentados previamente na [Subseção 3.3.4](#) (pág. 44): o *módulo de manutenção do volume de exploração* e o *módulo de planejamento de configurações*.

O primeiro módulo, discutido na [Seção 6.1](#), é responsável pela classificação do espaço tridimensional em uma de três categorias: *regiões ocupadas* (na borda ou no interior dos objetos), *regiões livres* (que podem ser ocupadas pelo robô durante a navegação) e *regiões inexploradas* (que não foram suficientemente observadas para que se conclua que sejam ocupadas ou livres).

Por sua vez, o segundo módulo se baseia na classificação disponível do espaço tridimensional e nas informações fornecidas pelo módulo de estimação de estados para identificar regiões propícias para a observação de novos dados sobre a cena e para avaliar poses da câmera capazes de efetuar essas observações. O módulo de planejamento de configurações é discutido na [Seção 6.2](#).

### 6.1 O módulo de manutenção do volume de exploração

A [Subseção 3.3.4](#) apresenta, em alto nível, o objetivo geral deste módulo e a estrutura de dados adotada para tal: Trata-se de representar o espaço tridimensional por meio de células discretas (voxels) que serão individualmente classificadas de acordo com a sua ocupação: *ocupadas*, *livres* ou *inexploradas*.

Essa classificação não será determinística (no sentido de atribuir uma dessas categorias discretas a cada célula), mas sim probabilística: A cada célula é atribuído

um valor escalar,  $o_t^{x,y,z}$ , que representa a probabilidade de sua ocupação (ou a probabilidade de que pertença ao volume de um objeto da cena), dada a sequência de observações visuais obtida até o instante de tempo  $t$ . A Eq. (3.5) (pág. 46), transcrita a seguir por conveniência, formaliza essa definição:

$$o_t^{x,y,z} \triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \mathcal{I}_{0..t}), \quad (6.1)$$

de onde  $o_t^{x,y,z} \approx 1$  indica que a célula  $(x, y, z)$  é *ocupada* e  $o_t^{x,y,z} \approx 0$  indica que é *livre*. Um valor intermediário para  $o_t^{x,y,z}$  indica que a célula não foi suficientemente observada para que seja classificada como ocupada ou livre (portanto ela está *inexplorada*), ou então foram coletadas evidências contraditórias e, portanto, a célula requer novas observações para que seja definitivamente classificada — em outras palavras, para todos os efeitos ela também é considerada *inexplorada*.

Intuitivamente, todas as células são inicializadas com um valor intermediário e devem gradativamente convergir para 0 ou 1. De fato, como consequência direta da Eq. (6.1), pode-se representar a probabilidade *a priori* (ou *probabilidade incondicional*) de ocupação de uma célula como sendo o seu valor em um instante de tempo  $t = -1$  (isto é, antes de qualquer observação) como:

$$o_{\text{incond}}^{x,y,z} = \mathcal{P}(\text{célula } x, y, z \text{ está ocupada}). \quad (6.2)$$

### 6.1.1 Os valores armazenados na estrutura de dados

A descrição apresentada nesta seção até o momento dá a entender que a estrutura de dados armazena os valores de  $o_t^{x,y,z}$ . No entanto, trabalhar diretamente com os valores de probabilidade gera instabilidades numéricas justamente quando os valores convergem para os limites 0 ou 1 [Thrun et al., 2005].

Uma maneira elegante de contornar esse problema é adotar a representação de probabilidades por meio do *logaritmo da razão de chances* (*log-odds ratio*). Essencialmente, este método consiste em mapear os valores de probabilidade para o espaço de números reais por meio da função *logit* :  $\mathbb{P} \rightarrow \mathbb{R}$  e realizar as incorporações de evidência neste espaço. O mapeamento simétrico é feito pela função sigmoide (*sigm* :  $\mathbb{R} \rightarrow \mathbb{P}$ ).

Os detalhes e a formalização matemática do uso de logaritmo da razão de chances, além de uma discussão sobre as vantagens da adoção deste método, são apresentados no [Apêndice B](#). Por questões de brevidade, neste capítulo apenas será apresentada a forma alternativa de representação da probabilidade de ocupação



de uma célula, cuja relação matemática pode ser vista a seguir:

$$l_t^{x,y,z} = \text{logit}(o_t^{x,y,z}) \quad \text{ou} \quad o_t^{x,y,z} = \text{sigm}(l_t^{x,y,z}), \quad (6.3)$$

onde  $l_t^{x,y,z}$  é o logaritmo da razão de chances de ocupação da célula  $(x, y, z)$  com base nas informações disponíveis até o instante  $t$ . Apenas os valores  $l_t^{x,y,z}$  serão armazenados (isto é, associados às células tridimensionais) e atualizados a cada observação.

### 6.1.2 Inicialização e atualização das células: Evidências de ocupação e de não ocupação

A cada instante de tempo  $t$ , a observação de um marco definitivo  $\mathbf{m}_t^m$  (a partir da câmera posicionada nas coordenadas  $\mathbf{c}_t$ ) fornece duas evidências sobre a ocupação de determinadas células:

1. A célula onde o marco  $\mathbf{m}_t^m$  está posicionada está ocupada. Neste caso, diz-se que *há evidência de ocupação dessa célula*; e
2. As células que formam o segmento de reta a partir do centro de projeção da câmera até o marco  $\mathbf{m}_t^m$  estão desocupadas (pois promovem uma vista sem obstáculos do marco), desde que não contenham outros marcos definitivos. Neste caso, diz-se que *há evidência de não ocupação dessas células*.<sup>a</sup>

Obedecendo ao caráter probabilístico do presente trabalho, nenhuma dessas evidências é definitiva: Apenas indica que esta observação fornece uma probabilidade de ocupação, a ser incorporada na estimação recursiva. Adaptando a Eq. (B.14) para as variáveis utilizadas neste módulo, tem-se que a estimação das probabilidades de ocupação de cada célula obedece à seguinte equação recursiva:

$$l_t^{x,y,z} = l_{t-1}^{x,y,z} + \text{logit}[\mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \mathcal{I}_t)] - l_{\text{incond}}^{x,y,z}, \quad (6.4)$$

cujo caso-base (antes da incorporação das primeiras evidências de observação) é visto a seguir:

$$l_{-1}^{x,y,z} = l_{\text{incond}}^{x,y,z} = \text{logit}(o_{\text{incond}}^{x,y,z}). \quad (6.5)$$

A Eq. (6.4) — em particular, o termo  $\mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \mathcal{I}_t)$  — recai em um de três casos, detalhados a seguir:

<sup>a</sup>Note que esta assertiva é diferente de dizer que “não há evidência acerca da ocupação dessas células”, o que será tratado como um terceiro caso.

1. *Quando há evidência de ocupação de uma célula:* Neste caso, a probabilidade a ser incorporada na Eq. (6.4) se refere ao seguinte valor:

$$o_{\text{ocup}} \triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \text{evidência de ocupação}), \quad (6.6)$$

de onde o passo de atualização é realizado como segue:

$$l_t^{x,y,z} = l_{t-1}^{x,y,z} + \text{logit}(o_{\text{ocup}}) - l_{\text{incond}}^{x,y,z} \quad (6.7)$$

O valor de  $o_{\text{ocup}}$  é uma constante que deve ser estipulada antes da execução da metodologia.

2. *Quando há evidência de não ocupação de uma célula:* Neste caso, a probabilidade a ser incorporada é análoga ao caso anterior e se refere ao seguinte valor:

$$o_{\text{livre}} \triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \text{evidência de não ocupação}), \quad (6.8)$$

de onde o passo de atualização é realizado como segue:

$$l_t^{x,y,z} = l_{t-1}^{x,y,z} + \text{logit}(o_{\text{livre}}) - l_{\text{incond}}^{x,y,z} \quad (6.9)$$

O valor de  $o_{\text{livre}}$  também é uma constante que deve ser estipulada antes da execução da metodologia.

3. *Quando não há evidência acerca da ocupação de uma célula:* Este é o caso mais simples, pois a probabilidade é representada pelo seguinte valor:

$$\begin{aligned} o_{\text{desc}} &\triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada} \mid \text{não há evidência sobre ocupação}) \\ &\triangleq \mathcal{P}(\text{célula } x, y, z \text{ está ocupada}) \\ &\triangleq o_{\text{incond}}^{x,y,z} \end{aligned} \quad (6.10)$$

de onde o passo de atualização é realizado como segue:

$$\begin{aligned} l_t^{x,y,z} &= l_{t-1}^{x,y,z} + \text{logit}(o_{\text{desc}}) - l_{\text{incond}}^{x,y,z} \\ &= l_{t-1}^{x,y,z}. \end{aligned} \quad (6.11)$$

Em outras palavras, na falta de evidências durante a observação no instante  $t$  o valor é replicado a partir do avaliado no passo anterior.

### 6.1.3 O algoritmo de atualização do volume de ocupação

Retornando à descrição dos dois tipos de evidências descrito no início da [Subseção 6.1.2](#), a identificação das células que caracterizam o segmento de linha reta no espaço tridimensional pode ser feita pela generalização de técnicas tradicionalmente utilizadas em duas dimensões, chamadas em inglês de *rasterisation* ou *rasterization*. Trata-se da determinação dos pixels que representam um segmento de reta no espaço bidimensional, comumente utilizada para traçar segmentos em imagens *bitmap*.

Neste trabalho, adotaremos o neologismo “rasterização” para identificar este processo, cuja adaptação para o espaço tridimensional é apresentado no [Algoritmo 6.1](#). Sem perda de generalização, adota-se no algoritmo que o espaço é discretizado em células cujas dimensões coincidem com a unidade da cena e cujos centros coincidem com as coordenadas compostas por valores inteiros.<sup>b</sup> O conjunto  $P$  retornado pelo algoritmo contém as coordenadas de todas as células que representam o segmento de reta tridimensional fornecido.

O passo seguinte é identificar todas as células que fazem parte de pelo menos uma rasterização dos segmentos que partem do centro de projeção da câmera até cada um dos marcos definitivos observados. O [Algoritmo 6.2](#) realiza esta tarefa: O laço das linhas 7-14 coleta todas as células sobre as quais há evidência de ocupação, com o cuidado de registrar aquelas que contêm os marcos observados (linha 13).

Finalmente, as equações apresentadas anteriormente para a avaliação dos valores de probabilidade são aplicados. Note-se, em particular, o uso implícito da Eq. (6.11) na linha 17 e as Eqs. (6.7) e (6.9) respectivamente nas linhas 20 e 24.

### 6.1.4 O fator de ocupação

Para facilitar o entendimento do estado do conhecimento sobre a ocupação espacial e o desenvolvimento dos algoritmos do Módulo de Planejamento de Configurações, é necessário dispor de uma ferramenta para analisar e classificar os valores de  $o_t^{x,y,z}$  de maneira mais direta. Para tanto, será apresentado a seguir o *fator de ocupação de uma célula*, que nada mais é do que a probabilidade de ocupação da

---

<sup>b</sup>Esta simplificação é evidenciada pelo uso, nos algoritmos, da função  $\text{round}(\cdot)$  para determinar as coordenadas discretas das células.

---

```

1 procedure RASTERIZA_RET_A_EM_3D ( $x_1, y_1, z_1, x_2, y_2, z_2$ )
2    $\triangleright$  Se o segmento de reta inicia e termina no mesmo ponto,
3    $\triangleright$  então retorna apenas a célula correspondente a esse ponto
4   if  $x_1 = x_2$  and  $y_1 = y_2$  and  $z_1 = z_2$  then
5     return  $\{(\text{round}(x_1), \text{round}(y_1), \text{round}(z_1))\}$ 
6   end if
7    $\triangleright$   $P$  é o conjunto de coordenadas de todas as células rasterizadas
8    $P \leftarrow \emptyset$ 
9   if  $|x_2 - x_1| \geq |y_2 - y_1|$  and  $|x_2 - x_1| \geq |z_2 - z_1|$  then
10     $\triangleright$  Se a projeção do segmento sobre o eixo  $x$  for maior do que a projeção
11     $\triangleright$  sobre os demais eixos, então o eixo  $x$  domina a rasterização
12    for  $x \leftarrow \text{round}[\min(x_1, x_2)], \dots, \text{round}[\max(x_1, x_2)]$  do
13       $y \leftarrow (x - x_1) \frac{y_2 - y_1}{x_2 - x_1}$ 
14       $z \leftarrow (x - x_1) \frac{z_2 - z_1}{x_2 - x_1}$ 
15       $P \leftarrow P \cup \{(x, \text{round}(y), \text{round}(z))\}$ 
16    end for
17  else if  $|y_2 - y_1| \geq |x_2 - x_1|$  and  $|y_2 - y_1| \geq |z_2 - z_1|$  then
18     $\triangleright$  Se a projeção do segmento sobre o eixo  $y$  for maior do que a projeção
19     $\triangleright$  sobre os demais eixos, então o eixo  $y$  domina a rasterização
20    for  $y \leftarrow \text{round}[\min(y_1, y_2)], \dots, \text{round}[\max(y_1, y_2)]$  do
21       $x \leftarrow (y - y_1) \frac{x_2 - x_1}{y_2 - y_1}$ 
22       $z \leftarrow (y - y_1) \frac{z_2 - z_1}{y_2 - y_1}$ 
23       $P \leftarrow P \cup \{(\text{round}(x), y, \text{round}(z))\}$ 
24    end for
25  else
26     $\triangleright$  Se a projeção do segmento sobre o eixo  $z$  for maior do que a projeção
27     $\triangleright$  sobre os demais eixos, então o eixo  $z$  domina a rasterização
28    for  $z \leftarrow \text{round}[\min(z_1, z_2)], \dots, \text{round}[\max(z_1, z_2)]$  do
29       $x \leftarrow (z - z_1) \frac{x_2 - x_1}{z_2 - z_1}$ 
30       $y \leftarrow (z - z_1) \frac{y_2 - y_1}{z_2 - z_1}$ 
31       $P \leftarrow P \cup \{(\text{round}(x), \text{round}(y), z)\}$ 
32    end for
33  end if
34  return  $P$ 
35 end procedure

```

---

**Algoritmo 6.1.** Procedimento Rasteriza\_Reta\_em\_3D: Rasteriza um segmento de reta no espaço tridimensional, retornando as coordenadas dos voxels sobre os quais o segmento passa.

célula,  $o_t^{x,y,z}$ , normalizada para um espaço mais conveniente, assim definido:

$$k_t^{x,y,z} \triangleq \begin{cases} \frac{o_t^{x,y,z} - o_{\text{desc}}}{1 - o_{\text{desc}}} & \text{se } o_t^{x,y,z} \geq o_{\text{desc}}, \\ \frac{o_t^{x,y,z} - o_{\text{desc}}}{o_{\text{desc}}} & \text{caso contrário.} \end{cases} \quad (6.12)$$

---

```

1 procedure INCORPORA_EVIDÊNCIAS_DE_OBSERVAÇÃO ( $\mathbf{c}_t$ )
2   ▷  $P$  é o conjunto de coordenadas de todas as células rasterizadas
3    $P \leftarrow \emptyset$ 
4   ▷  $O$  é o conjunto de coordenadas das células ocupadas
5    $O \leftarrow \emptyset$ 
6    $(x_c, y_c, z_c) \leftarrow \mathbf{c}_t$ 
7   for each marco resolvido observado  $m$  do
8     ▷ Primeiro momento das coordenadas estimadas de  $m$ 
9      $(x_m, y_m, z_m) \leftarrow \hat{\mathbf{m}}_t^m$ 
10    ▷ Inclui em  $P$  todas as células do centro de projeção da câmera até o marco  $m$ 
11     $P \leftarrow P \cup \text{Rasteriza\_Reta\_em\_3D}(x_c, y_c, z_c, x_m, y_m, z_m)$ 
12    ▷ Inclui em  $O$  a célula que contém o marco  $m$ 
13     $O \leftarrow O \cup \{(\text{round}(x_m), \text{round}(y_m), \text{round}(z_m))\}$ 
14  end for
15  ▷ Assume-se que, por padrão, os valores das células em  $t$  são idênticos
16  ▷ aos do passo anterior (Eq. (6.11))
17   $l_t^{x,y,z} \leftarrow l_{t-1}^{x,y,z} \quad \forall x, y, z$ 
18  ▷ Atualização das probabilidades das células ocupadas (Eq. (6.7))
19  for each  $(x, y, z) \in O$  do
20     $l_t^{x,y,z} \leftarrow l_{t-1}^{x,y,z} + \text{logit}(o_{\text{ocup}}) - l_{\text{incond}}^{x,y,z}$ 
21  end for
22  ▷ Atualização das probabilidades das células livres (Eq. (6.9))
23  for each  $(x, y, z) \in P - O$  do
24     $l_t^{x,y,z} \leftarrow l_{t-1}^{x,y,z} + \text{logit}(o_{\text{livre}}) - l_{\text{incond}}^{x,y,z}$ 
25  end for
26 end procedure

```

---

**Algoritmo 6.2.** Procedimento Incorpora\_Evidências\_De\_Observação: Coleta todas as evidências de espaços ocupados e livres e incorpora-as no volume de exploração.

Segundo esta definição, a classificação de uma célula pode ser assim analisada:

- Se  $k_t^{x,y,z} \approx 1$ , então a célula encontra-se ocupada;
- Se  $k_t^{x,y,z} \approx -1$ , então a célula encontra-se livre;
- Se  $k_t^{x,y,z} \approx 0$ , então não há evidências sobre a ocupação da célula.

Valores intermediários indicam que há evidências não conclusivas sobre a ocupação da célula. Analisando de outra maneira, o módulo do fator de planejamento,  $|k_t^{x,y,z}|$ , pode ser percebido como o percentual de conhecimento sobre a ocupação da célula — uma conveniência que será utilizada na seção seguinte para o processo de planejamento.

## 6.2 O módulo de planejamento de configurações

O objetivo deste módulo é determinar uma configuração futura para o robô,  $\mathbf{r}_{\text{fut}}$ , de modo a tentar alcançar os objetivos discutidos na [Subseção 3.3.4](#): Melhoria da localização, melhoria do mapa e exploração.

O conceito do fator de ocupação ([Subseção 6.1.4](#)) permite perceber o processo de exploração sob um ponto de vista menos filosófico e mais quantitativo:

- A melhoria de localização e do mapa estão relacionadas ao enquadramento de células com  $k_t^{x,y,z} > 0$ . Em particular, a melhoria de localização pode ser alcançada com a observação de células com  $k_t^{x,y,z} \approx 1$ , enquanto o refinamento sobre o conhecimento da geometria dos objetos pode ser alcançado pela observação de células com valores menores;
- A exploração é facilmente alcançada pela observação de células com  $k_t^{x,y,z} = 0$ .

Embora a descrição apresentada estabeleça certas diretivas para o funcionamento autônomo da reconstrução, ela não permite criar um plano de ação, já que não estabelece critérios de comportamento ou prioridades na observação das células. Na próxima subseção será apresentada a ideia que norteia o processo de planejamento esperado por este módulo.

### 6.2.1 Visão geral

Este módulo desmembra a tarefa de determinar um valor para  $\mathbf{r}_{\text{fut}}$  na forma de três subtarefas, que são resumidas a seguir:

1. determinação da distância desejada entre a câmera e o alvo,  $d_{\text{fut}}$ ;
2. determinação da região do espaço que deve ser enquadrada pela câmera, representada por um *ponto-alvo*,  $\mathbf{t}_{\text{fut}} \in \mathbb{R}^3$ ;
3. determinação da orientação que a câmera deve adotar para observar o ponto-alvo, representada pelo quatérnio  $\mathbf{q}_{\text{fut}}$ .

Essas três informações em conjunto determinam univocamente a configuração futura a ser adotada pela câmera,  $\mathbf{r}_{\text{fut}}$ .

A distância da câmera ao ponto-alvo,  $d_{\text{fut}}$ , é um valor constante calculado como a média da distância da câmera até os marcos resolvidos mais próximos, no momento em que estes são criados pelo processo de estimação. Opcionalmente, essa distância pode ser predeterminada pelo usuário no início do processo de reconstrução. A existência de uma distância desejada invariável tende a reduzir

a variação de escala na reobservação das características visuais, facilitando o processo de correspondência de pontos salientes entre imagens.

O ponto-alvo,  $\mathbf{t}_{\text{fut}}$ , é um ponto da superfície do objeto de interesse e que esteja no limite do que se conhece sobre o objeto (ou seja, na *borda de exploração dos objetos*, um conceito que será formalizado na [Subseção 6.2.2](#)). Este critério permite abordar simultaneamente dois objetivos: (i) A câmera enquadra diversos marcos resolvidos, melhorando assim a estimativa sobre as suas posições (melhoria do mapa); e (ii) Como o alvo da câmera se encontra na borda de exploração dos objetos, parte da imagem coleta informações sobre uma região desconhecida, promovendo assim a exploração.

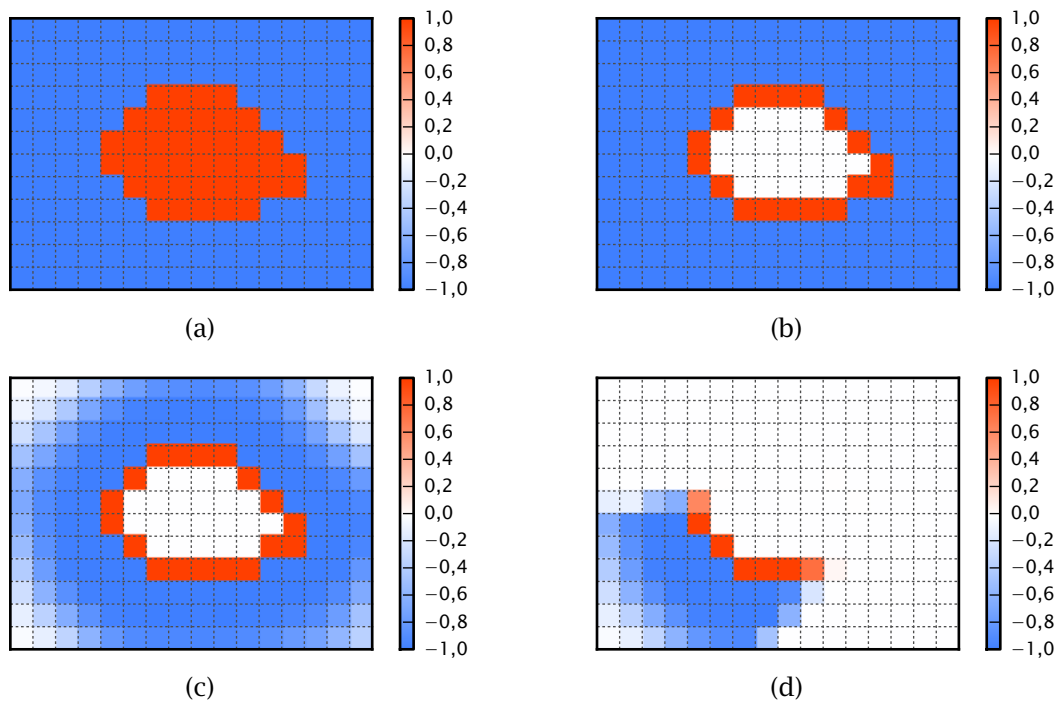
Finalmente, a orientação de observação,  $\mathbf{q}_{\text{fut}}$ , é determinada de modo a ficar aproximadamente alinhada com a normal da superfície do objeto em torno do ponto-alvo. A metodologia para estimar a normal será apresentada posteriormente na [Subseção 6.2.3](#).

## 6.2.2 Bordas de exploração

No caso ideal em que todo o espaço tridimensional é completamente conhecido em um determinado instante  $t$ , tem-se que  $k_t^{x,y,z} = \pm 1$  para todas as células  $(x, y, z)$  (Figura 6.1(a)). Na prática, é muito provável que existam diversas células no interior dos objetos reconstruídos que jamais poderão ser observados, já que os objetos são opacos (Figura 6.1(b)). Além disto, já que a metodologia não estabelece limites para o espaço a ser explorado, o domínio de  $(x, y, z)$  é infinito, o que garante que a qualquer momento existe uma quantidade infinita de células para as quais  $k_t^{x,y,z} = 0$ .

No entanto, é importante lembrar que o objetivo deste trabalho não é a exploração exaustiva do ambiente (mesmo que o seu domínio fosse limitado), e sim a recuperação da geometria de determinados objetos de interesse. Portanto, é esperado que o espaço explorado não se estenda para regiões distantes do objeto reconstruído, de modo que o conhecimento sobre a ocupação deve permanecer em  $k_t^{x,y,z} \approx 0$  para regiões não interessantes do espaço (Figura 6.1(c)).

Toda esta argumentação apresentada tem por objetivo chamar a atenção do leitor para um aspecto crucial do processo exploratório: Enquanto a reconstrução estiver em andamento (Figura 6.1(d)), as células desconhecidas (isto é, com  $k_t^{x,y,z} \approx 0$ ) dominam o espaço. Cabe a este módulo, portanto, determinar quais dessas células devem ser exploradas e quais devem ser ignoradas (por pertencerem ao interior de um objeto ou por estarem distantes deste).

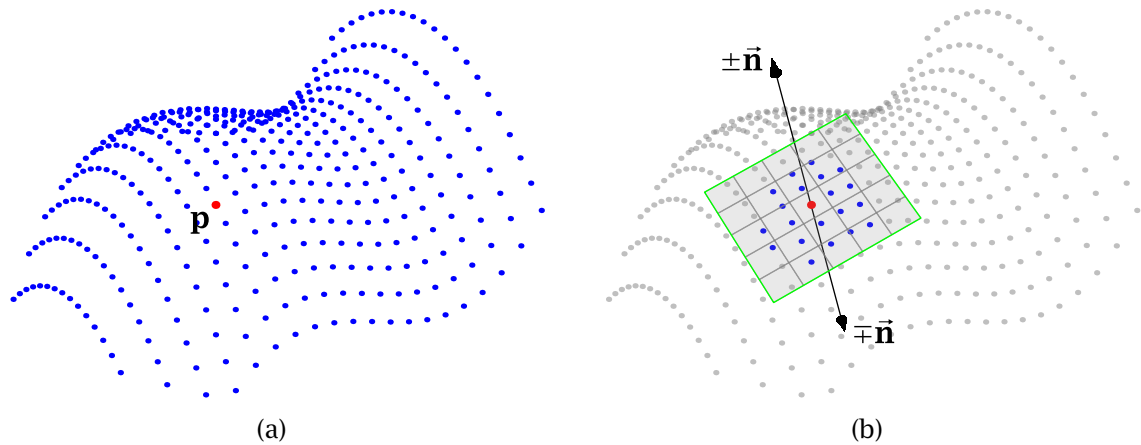


**Figura 6.1.** Representação gráfica dos fatores de ocupação de algumas instâncias de volumes de exploração (aqui representados em duas dimensões para facilitar a visualização). **(a)** Caso ideal: O conhecimento é completo acerca de todas as células ( $k_t^{x,y,z} = \pm 1$ ); **(b)** Na prática, não há como coletar conhecimento sobre o interior dos objetos, pois este não pode ser observado; **(c)** O desconhecimento da ocupação de células distantes do objeto de interesse não prejudica a metodologia deste trabalho; **(d)** Caso incompleto: As informações sobre ocupação de células importantes ainda estão sendo coletadas.

Para atingir tal objetivo, a metodologia utilizada analisa a relação entre os fatores de ocupação de células vizinhas. Em termos gerais, pode-se perceber que certas transições fornecem indícios importantes para nortear o processo exploratório:

- Transições entre  $k_t^{x,y,z} \approx 1$  e  $k_t^{x',y',z'} \approx -1$  representam a superfície dos objetos. Essas transições são fortemente desejadas e representam sucesso na exploração de determinadas regiões do objeto;
- Transições entre  $|k_t^{x,y,z}| \approx 1$  e  $k_t^{x',y',z'} \approx 0$  sinalizam as *bordas de exploração*, ou seja, os limites entre regiões conhecidas e inexploradas. De particular interesse são as *bordas de exploração dos objetos*, que são as transições entre  $k_t^{x,y,z} \approx 1$  e  $k_t^{x',y',z'} \approx 0$ . Essas transições devem ser investigadas a fim de determinar se apenas delimitam regiões não exploráveis (como o interior da Figura 6.1(c)) ou se representam de fato alvos para o prosseguir-





**Figura 6.2.** Estimação da direção do vetor normal de um ponto  $\mathbf{p}$  da superfície de um objeto, dada uma nuvem de pontos que representa essa superfície. **(a)** Nuvem de pontos de entrada; **(b)** Estimação do vetor normal da superfície no ponto  $\mathbf{p}$ , destacando os vizinhos mais próximos de  $\mathbf{p}$  (em azul) e as duas possíveis soluções estimadas pelo método descrito,  $\pm\vec{\mathbf{n}}$  e  $\mp\vec{\mathbf{n}}$ . Note-se que o sentido não pode ser inferido pela análise da vizinhança.

mento da exploração (como os extremos da sequência de células ocupadas da Figura 6.1(d)).

### 6.2.3 Estimação do vetor normal de um ponto da superfície

Conforme estabelecido anteriormente, a orientação da observação,  $\mathbf{q}_{\text{fut}}$ , será alinhada com a normal da superfície do objeto em torno do ponto-alvo. Esta subseção apresenta a metodologia para estimar o vetor normal  $\vec{\mathbf{n}}$  em um ponto qualquer  $\mathbf{p}$  da superfície do objeto, quando este é representado por uma nuvem de pontos (ou por um conjunto de marcos resolvidos, conforme as convenções adotadas neste trabalho) (Figura 6.2(a)). Convencionou-se que a orientação do vetor normal é para fora do objeto.

Uma maneira simples e muito utilizada para estimar a direção (não o sentido) de  $\vec{\mathbf{n}}$  é selecionar os marcos mais próximos de  $\mathbf{p}$  e considerar que localmente estes representam um plano ortogonal ao vetor normal (Figura 6.2(b)). Para tanto, estabelece-se o número de vizinhos que será considerado,  $N_V$ , e define-se um conjunto composto pelos  $N_V$  marcos mais próximos de  $\mathbf{p}$ :

$$\mathcal{V}_t^m \triangleq \{m \mid \mathbf{m}_t^m \text{ é um dos } N_V \text{ marcos mais próximos de } \mathbf{p}\}. \quad (6.13)$$

Naturalmente, é de se esperar que esses pontos em geral não sejam coplana-

res. Aqui será adotado um método para a regressão de um plano que minimiza a soma de suas distâncias quadráticas aos marcos  $\mathcal{V}_t^m$ . Visto de outra maneira, este método determina o plano cuja normal é a direção que captura a menor variação de alinhamento dos pontos. Tal plano passa necessariamente pelo centroide das coordenadas desses marcos, calculado como segue:

$$\bar{\mathbf{m}} \triangleq \frac{1}{N_V + 1} \sum_{m \in \mathcal{V}_t^m} \hat{\mathbf{m}}_t^m. \quad (6.14)$$

O passo seguinte é determinar a seguinte matriz positiva semidefinida:

$$\mathbf{M} \triangleq \sum_{m \in \mathcal{V}_t^m} (\hat{\mathbf{m}}_t^m - \bar{\mathbf{m}})^\top (\hat{\mathbf{m}}_t^m - \bar{\mathbf{m}}). \quad (6.15)$$

Assume-se que a direção da normal da superfície do objeto no ponto  $\mathbf{p}$ , designada por  $|\bar{\mathbf{n}}|$ , é o autovetor correspondente ao menor autovalor de  $\mathbf{M}$  — o que corresponde à direção de menor variação espacial dos marcos selecionados.

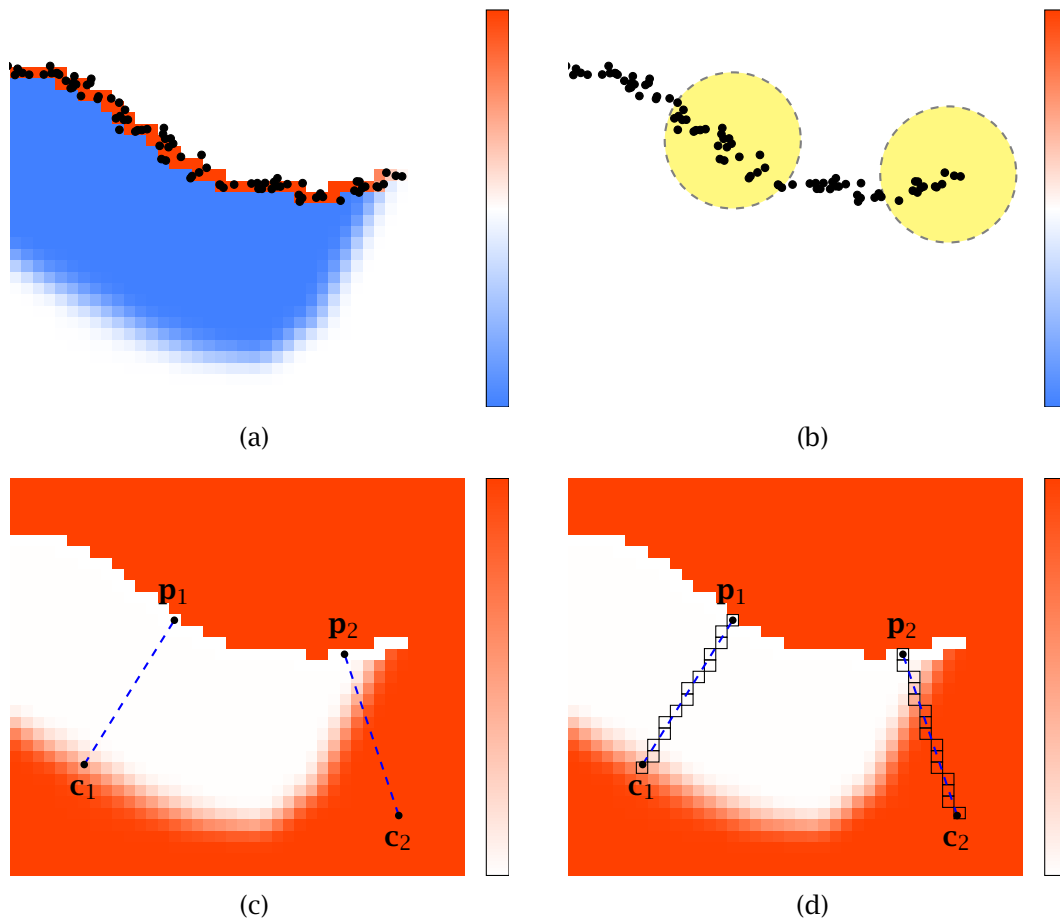
Nota-se, porém, que o sentido de  $\bar{\mathbf{n}}$  não pode ser trivialmente determinado pela análise dos pontos vizinhos: de fato, este é um tema ativo de desenvolvimento científico.

No entanto, este problema é simplificado nesta metodologia pelo fato de se dispor de uma informação histórica acerca dos pontos de vista de observação de cada marco. Trata-se do *vetor de reobservação* (ver [Algoritmo 5.5](#)). Conforme a conceituação apresentada no [Subseção 5.2.1](#), este vetor “representa a direção de um ponto de vista que, se adotado por uma câmera, indica uma alta probabilidade de reobservação do respectivo marco”. Embora a direção do vetor de reobservação não seja necessariamente coincidente com o vetor normal, espera-se que os sentidos desses dois vetores sejam contrários entre si. Desta forma, ajusta-se o sentido de  $\bar{\mathbf{n}}$  (isto é, multiplica-se o seu valor por  $\pm 1$ ) de modo que  $\bar{\mathbf{n}} \cdot \bar{\mathbf{r}}^m \leq 0$ , dado o vetor de reobservação  $\bar{\mathbf{r}}^m$  do marco mais próximo de  $\mathbf{p}$ .

Nas seções seguintes, o vetor normal avaliado no instante  $t$  para uma célula posicionada em  $(x, y, z)$  será designado por  $\bar{\mathbf{n}}_t^{x,y,z}$ .

#### 6.2.4 Determinação do ponto-alvo

O ponto-alvo será determinado a partir do conjunto de células que forma a borda de exploração dos objetos (conceituada na [Subseção 6.2.2](#)). O processo de seleção se baseia na atribuição de um valor de utilidade exploratória a cada uma dessas



**Figura 6.3.** Avaliação da utilidade das células e dos pontos de vista (aqui representada em duas dimensões para facilitar a visualização). **(a)** Dados de entrada: A escala de cores representa o fator de ocupação das células,  $o_t^{x,y,z}$ , enquanto os pontos representam os marcos definitivos; **(b)** Análise de vizinhança: A vizinhança da célula da direita é menos populosa do que a vizinhança da célula da esquerda, indicando que a primeira alternativa é mais útil no que diz respeito à expansão da fronteira de exploração; **(c)–(d)** Análise do ponto de vista: A observação da célula  $p_1$  por uma câmera posicionada em  $c_1$  cruza diversas células sobre as quais o conhecimento está consolidado (sabe-se que estão desocupadas), enquanto a observação da célula  $p_2$  por uma câmera em  $c_2$  cruza diversas células cujo fator de ocupação é próximo ou igual a zero. Assim, a segunda alternativa é mais útil no que diz respeito à exploração de novas regiões do espaço.

células. As coordenadas do ponto-alvo,  $t_{fut}$ , são então fixadas nas coordenadas do centro da célula com o maior valor de utilidade exploratória.

Essa função de utilidade exploratória leva os seguintes fatores em consideração, ilustradas na Figura 6.3:

1. O isolamento da célula, isto é, a quantidade de marcos próximos a ela. A busca

de células mais isoladas (ou seja, com poucos marcos em sua vizinhança) promove a busca por regiões pouco exploradas (Figura 6.3(b)).

A *utilidade do isolamento da célula*,  $U_N^{x,y,z}$ , é avaliada como segue:

$$U_N^{x,y,z} \triangleq 1 - \alpha_N \text{count}(\|\hat{\mathbf{m}}_t^m - (x, y, z)\| \leq r), \quad (6.16)$$

onde  $\alpha_N$  é um fator de normalização definido como o inverso do maior valor de  $U_N^{x,y,z}$  no instante de tempo atual:

$$\alpha_N \triangleq \frac{1}{\max_{\forall (x,y,z)} U_N^{x,y,z}}, \quad (6.17)$$

que garante que  $0 \leq U_N^{x,y,z} \leq 1$ , e  $r$  é o raio da esfera que contém todos os marcos que estão dentro do campo de visada da câmera, independentemente de seu ponto de vista, ou:

$$r \triangleq d_{\text{fut}} \tan(\theta_{\text{min fov}}), \quad (6.18)$$

onde  $\theta_{\text{min fov}}$  é o menor ângulo de visada da câmera.

2. A medida do conhecimento acumulado acerca das células que estão na direção do vetor normal da superfície. Se essas células forem pouco exploradas (ou seja, se várias apresentam  $k_t^{x,y,z} \approx 0$ ), então a observação ao longo dessa direção promoverá a expansão do conhecimento da cena.

Para se chegar a esta medida, será definida a seguir a *medida de desconhecimento exploratório de uma célula*,  $u_E^{x,y,z}$ , com base no complemento da medida de exploração dessa célula:

$$u_E^{x,y,z} \triangleq \begin{cases} 1 - |k_t^{x,y,z}| & \text{se } k_t^{x,y,z} \leq 0 \\ 0 & \text{caso contrário.} \end{cases} \quad (6.19)$$

Esta medida é zero caso haja evidência de ocupação da célula correspondente. Caso contrário, a utilidade é tanto mais positiva quanto menor for o módulo do seu fator de ocupação, isto é, quanto menos a célula for explorada. O valor máximo de  $u_E^{x,y,z}$  é obtido para as células nunca observadas.

A *utilidade exploratória de uma célula*,  $U_E^{x,y,z}$ , é avaliada como a média das medidas de desconhecimento exploratório de todas as células que estão entre o centro de projeção da câmera candidata e a célula em questão. Em outras palavras, esta função de utilidade captura o quanto um determinado ponto

---

```

1 procedure AVALIA_UTILIDADE_EXPLORATÓRIA_DA_CÉLULA ( $x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}}, \vec{n}_t^{x,y,z}$ )
2   ▷ ( $x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}}$ ) são as coordenadas do centro da célula investigada
3   ▷  $\vec{n}_t^{x,y,z}$  é o vetor normal estimado conforme explicado na Subseção 6.2.3
4   ▷ Posição para o centro de projeção da câmera,
5   ▷ de modo a observar o marco  $m$  ortogonalmente ao seu vetor normal:
6   ( $x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}$ ) ← ( $x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}}$ ) +  $d_{\text{fut}} \vec{n}_t^{x,y,z}$ 
7   ▷ Rasterização do segmento de reta que liga os dois pontos (ver Algoritmo 6.1):
8    $P$  ← Rasteriza_Reta_em_3D( $x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}}, x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}$ )
9   ▷ Avaliação da utilidade do marco  $m$ , baseada na Eq. (6.19):
10  return  $\frac{1}{|P|} \sum_{(x,y,z) \in P} u_E^{x,y,z}$ 
11 end procedure

```

---

**Algoritmo 6.3.** Procedimento Avalia\_Utilidade\_Exploratória\_da\_Célula: Se a célula  $(x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}})$  for observada a partir do ponto de vista alinhado com o vetor normal (representado por  $\vec{n}_t^{x,y,z}$ ) e distante  $d_{\text{fut}}$  da célula, este algoritmo avalia um valor entre 0 e 1 que representa o quanto se desconhece sobre as células que estão ao longo da reta de visada. Um valor alto de retorno indica que diversas células na reta de visada são inexploradas ou pouco exploradas, sinalizando uma alta utilidade exploratória para este ponto de vista.

de vista será capaz de observar células inexploradas (Figuras 6.3(c)–(d)). O procedimento é detalhado no [Algoritmo 6.3](#).

3. A mudança do ponto de vista. Como a orientação da câmera é determinada a partir do vetor normal da célula selecionada,  $\vec{n}_t^{x,y,z}$ , é importante descartar a seleção de pontos-alvo candidatos que causem uma mudança muito grande na direção de visada da câmera. Isto não é desejado porque faria a câmera rodar o objeto, gerando uma trajetória comprida (e portanto custosa) e possivelmente levando a câmera a transitar por uma grande extensão desconhecida, gerando distorções no processo de estimação causadas pelo efeito de *dead-reckoning*.

Esta limitação é formalizada pelo *fator de corte de mudanças de ponto de vista*,  $\kappa_{\text{PoV}}$ , assim definido:

$$\kappa_{\text{PoV}} \triangleq \begin{cases} 1 & \text{se } -\vec{n}_t^{x,y,z} \cdot \vec{z}_t \geq \cos \theta_{\text{PoV}}, \\ 0 & \text{caso contrário,} \end{cases} \quad (6.20)$$

onde  $\theta_{\text{PoV}}$  é o ângulo que representa a mudança máxima do ponto de vista e  $\vec{z}_t$  é o vetor-direção da câmera no instante  $t$ .

Finalmente, a *utilidade de uma célula* é definida como a combinação linear

das duas utilidades definidas acima:

$$U_t^{x,y,z} \triangleq \kappa_{\text{PoV}} (w_N U_N^{x,y,z} + w_E U_E^{x,y,z}), \quad (6.21)$$

onde  $w_N$  e  $w_E$  são os pesos que definem o impacto de cada função de utilidade.

O ponto-alvo é definido como o centro da célula que maximiza a função de utilidade:

$$\mathbf{t}_{\text{fut}} \triangleq \arg \max_{(x,y,z)} U_t^{x,y,z}. \quad (6.22)$$

### 6.2.5 Determinação da orientação da câmera

No procedimento para a determinação do ponto-alvo detalhado na subseção anterior, pode-se perceber a avaliação de diversos pontos de vista para a avaliação da utilidade exploratória das células ([Algoritmo 6.3](#), linha 6). Esta avaliação fixa implicitamente duas informações a respeito da pose planejada para a câmera:

- a sua posição, definida como segue:

$$(x_{\text{cam}}, y_{\text{cam}}, z_{\text{cam}}) \triangleq (x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}}) + d_{\text{fut}} \vec{\mathbf{n}}_t^{x,y,z}; \quad (6.23)$$

- a direção de visada, fixada em  $-\vec{\mathbf{n}}_t^{x,y,z}$ , de modo a observar  $(x_{\text{cel}}, y_{\text{cel}}, z_{\text{cel}})$  (isto é, o centro da célula) no centro da imagem.

Uma vez que o ponto-alvo é fixado no centro da célula de maior utilidade (Eq. (6.22)), também serão adotadas a posição e direção da câmera utilizadas para a avaliação dessa utilidade.

Entretanto, esses valores não são suficientes para fixar a orientação da câmera de maneira não ambígua, já que a rotação da câmera em torno do eixo principal permanece indeterminada. Desta forma, para os efeitos deste trabalho considera-se que essa variável deve ser definida de acordo com a conveniência das restrições holonômicas do robô adotado na prática. Nos casos em que essas restrições não se aplicam, considera-se que a orientação da câmera é fixada de modo a alinhar o eixo vertical da imagem com o eixo ortogonal ao plano do solo da cena, de modo que todas as imagens são obtidas de modo a obedecer ao conceito fotográfico de imagens panorâmicas (*landscape orientation*).

### 6.2.6 Desistência do plano atual

A cada iteração de coleta de evidências sobre o ambiente, o módulo de planejamento de configurações determina se o plano de ação atualmente em execução deve ser abandonado e substituído. A decisão de desistência do plano atual baseia-se em dois critérios, ambos geométricos:

- Caso o robô se aproxime demasiadamente de algum marco resolvido, o que sinaliza colisão iminente com o objeto em exploração;
- Caso o robô se aproxime demasiadamente dos obstáculos identificados pelo módulo de detecção de obstáculos (que faz parte da camada de processamento, ilustrada na Figura 3.3), se este módulo for implementado.

Em ambos os casos, a decisão depende da modelagem do espaço físico ocupado pelo robô. Para simplificar o algoritmo e reduzir a carga computacional deste processo de decisão, pode-se alternativamente adotar que nenhum marco ou obstáculo deve entrar na esfera centrada no centro de projeção da câmera e cujo raio engloba todo o espaço ocupado pelo robô.





# Capítulo 7

## Experimentos

**E**STE CAPÍTULO TEM POR OBJETIVO apresentar os resultados da avaliação das metodologias desenvolvidas neste trabalho, por meio de experimentos realizados em ambientes simulados e reais.

Este capítulo está dividido nas seguintes seções:

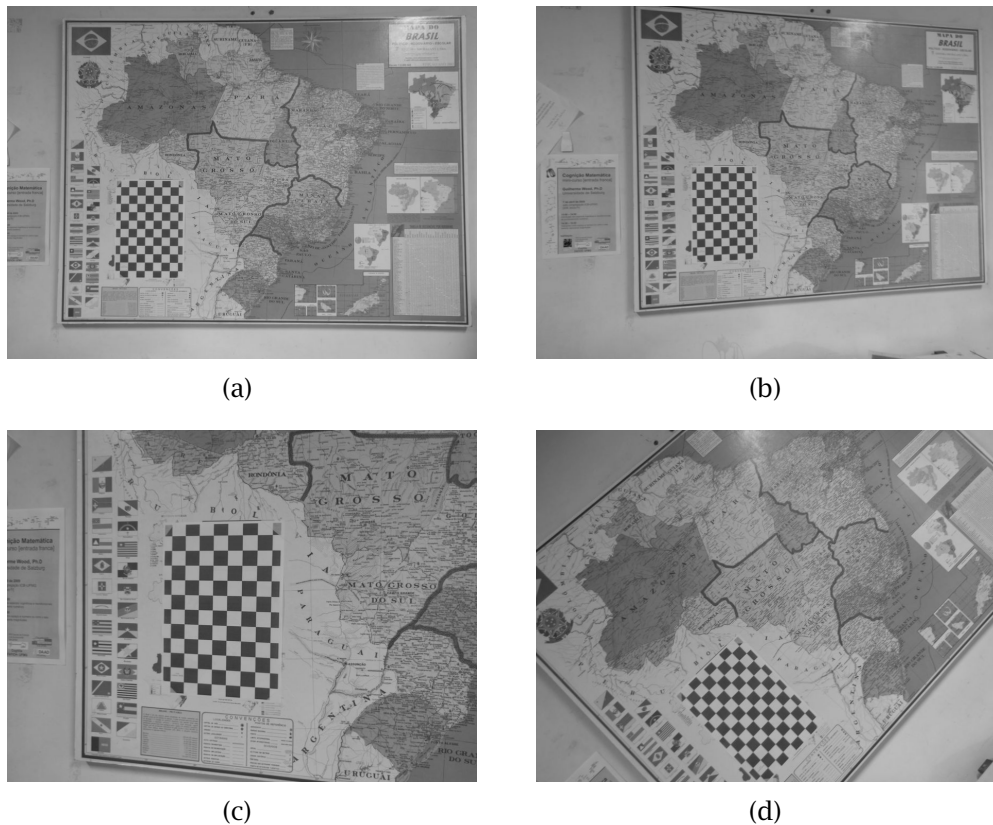
- A avaliação do *módulo de correspondência de pontos salientes*, realizada em cenas reais e apresentada na [Seção 7.1](#);
- A avaliação conjunta do *módulo de manutenção de marcos* e do *módulo SLAM*, realizada em cenas reais e apresentada na [Seção 7.2](#);
- A avaliação do sistema como um todo, realizada em cenas simuladas e reais e apresentada na [Seção 7.3](#).

### 7.1 Módulo de correspondência de pontos salientes

Nesta seção será apresentada uma análise comparativa entre o método clássico de correspondência de marcos visuais, representado pela Eq. (4.2), e a metodologia proposta no [Capítulo 4](#), baseada na busca da consistência geométrica global das correspondências.

Ambos os algoritmos foram implementados em Matlab, com exceção da rotina de comparação entre os vetores descritores, feita em C++. Nenhum aspecto da implementação contempla o paralelismo de processamento (*multi-threading*).

A plataforma utilizada nos testes consiste em um processador Intel® Core™2 Duo de 2 GHz com 4 GB de RAM, executando um sistema operacional GNU/Linux Ubuntu de 64 bits (com *kernel* de versão 2.6.28). As imagens foram obtidas por uma máquina fotográfica digital *Canon PowerShot SX10 IS* com 10 megapixels e



**Figura 7.1.** Conjunto de imagens usadas nos experimentos do módulo de correspondência de pontos salientes: **(a)** imagem-base, **(b)** imagem depois da mudança do ponto de vista, **(c)** imagem depois da aproximação da câmera e **(d)** imagem depois da rotação da câmera.

foram reduzidas para 1/16 de sua resolução original, resultando em imagens de  $912 \times 684$  pixels.

Para a avaliação da metodologia proposta, foram obtidas algumas imagens de uma cena rica em características visuais, vistas na Figura 7.1. A metodologia proposta foi avaliada pela análise das correspondências dos pontos salientes entre a imagem-base (Figura 7.1(a)) e as demais, cobrindo, portanto, três transformações importantes: mudança do ponto de vista (Figura 7.1(b)), aproximação da câmera (Figura 7.1(c)) e rotação da câmera (Figura 7.1(d)). Os conjuntos de pontos salientes  $\mathcal{F}_p$  e  $\mathcal{F}_q$  foram obtidos pelo uso do algoritmo SIFT [Lowe, 1999, 2004].

Embora a metodologia não seja limitada a cenas com regiões planares, esta avaliação foi propositalmente realizada com base na observação de um objeto planar. Esta configuração de testes permite uma avaliação qualitativa direta e simples dos resultados obtidos por meio da homografia entre as imagens. Neste trabalho, a homografia é descrita como sendo uma função  $H : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  que mapeia

coordenadas bidimensionais  $(x_p^i, y_p^i)$  no espaço da imagem, em pixels, da primeira para a segunda imagem:

$$\begin{bmatrix} x_p'^i \\ y_p'^i \end{bmatrix} = H \left( \begin{bmatrix} x_p^i \\ y_p^i \end{bmatrix} \right). \quad (7.1)$$

A homografia entre pares de imagens é estimada pelo uso da ferramenta *Camera Calibration Toolbox for Matlab* [Bouguet, 2008], baseada no padrão de calibração xadrez visível em todas as imagens.

Em condições ideais, dada uma correspondência qualquer  $\langle i, j \rangle$  entre duas imagens, as coordenadas do ponto saliente da segunda imagem,  $(x_q^j, y_q^j)$ , são iguais às coordenadas do correspondente ponto saliente da primeira imagem,  $(x_p^i, y_p^i)$ , transformadas pela função de homografia. Na prática, ruídos provenientes de várias fontes afetam essa igualdade: falsas correspondências, incertezas de calibração e do modelo projetivo, determinação das coordenadas dos pontos salientes, etc. Para mensurar esses ruídos, define-se o *vetor de erro de homografia*,  $\vec{\varepsilon}_{i,j}$ , como a diferença entre as coordenadas preditas (a partir da homografia) e as observadas (a partir do detector de pontos salientes e do processo de correspondência):

$$\vec{\varepsilon}_{i,j} \triangleq \begin{bmatrix} x_q^j \\ y_q^j \end{bmatrix} - H \left( \begin{bmatrix} x_p^i \\ y_p^i \end{bmatrix} \right). \quad (7.2)$$

A partir dos vetores de erro calculados sobre um conjunto de correspondências  $C$ , serão analisadas duas medidas globais de erro: o *Erro Quadrático Médio*,  $\text{RMSE}(C)$ , definido como:

$$\text{RMSE}(C) = \frac{1}{|C|} \sqrt{\sum_{\langle i,j \rangle \in C} \|\vec{\varepsilon}_{i,j}\|^2}, \quad (7.3)$$

e o *Erro Absoluto Médio*,  $\text{MAE}(C)$ , definido como:

$$\text{MAE}(C) = \frac{1}{|C|} \sum_{\langle i,j \rangle \in C} \|\vec{\varepsilon}_{i,j}\|. \quad (7.4)$$

Valores menores calculados para essas medidas de erro indicam melhores resultados obtidos com um determinado conjunto  $C$ . Como o  $\text{RMSE}(C)$  é mais sensível a valores espúrios (*outliers* — valores particularmente altos de  $\vec{\varepsilon}_{i,j}$ ), uma redução significativa das medidas de  $\text{RMSE}(C)$  possivelmente indica uma redução na ocorrência desses valores.

**Tabela 7.1.** Número de correspondências e das medidas de erro obtidas com o método clássico (com  $\tau_M = 1,5$ ) e com o método proposto. Nota-se o aumento consistente do número de correspondências e o decréscimo significativo de ambas as medidas de erro (RMSE e MAE).

(a) Mudança do ponto de vista			
<i>Método</i>	<i>Correspondências</i>	<i>RMSE [px]</i>	<i>MAE [px]</i>
Método clássico	2 887	90,50	22,70
Método proposto	3 565	14,10	8,40
Melhoria (%)	23,5%	541,8%	170,2%

(b) Aproximação da câmera			
<i>Método</i>	<i>Correspondências</i>	<i>RMSE [px]</i>	<i>MAE [px]</i>
Método clássico	1 571	129,54	41,60
Método proposto	1 968	9,33	6,18
Melhoria (%)	25,3%	1 288,4%	573,1%

(c) Rotação da câmera			
<i>Método</i>	<i>Correspondências</i>	<i>RMSE [px]</i>	<i>MAE [px]</i>
Método clássico	3 242	73,65	16,81
Método proposto	4 245	11,24	7,16
Melhoria (%)	30,9%	555,2%	134,8%

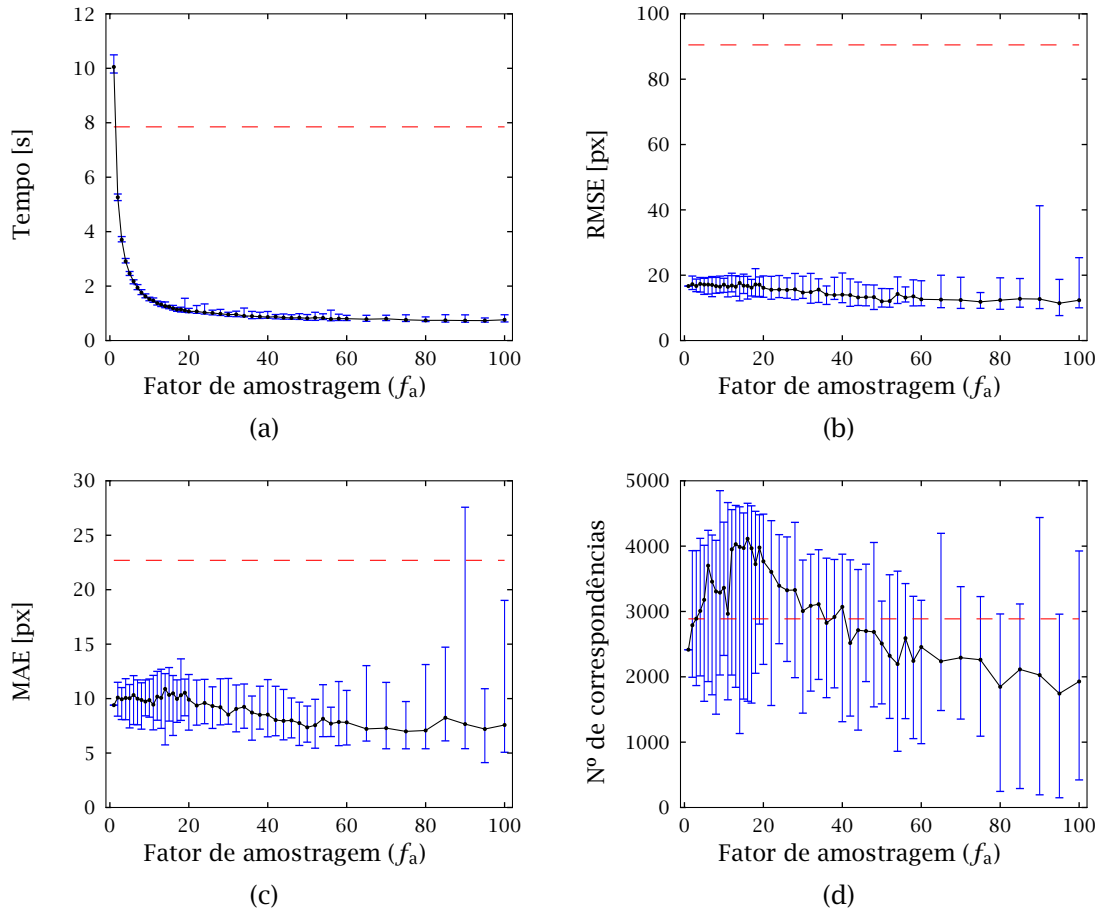
Para comparar os resultados obtidos com o método tradicional (Eq. (4.2)) — aqui referenciados como  $C_{\text{trad}}$  — com os obtidos pelo método proposto (Eq. (4.12)), os experimentos foram conduzidos a fim de levantar as seguintes medidas de desempenho:

- o número de correspondências obtidas por cada método,  $|C_{\text{trad}}|$  vs.  $|C_{\text{final}}|$ ;
- as medidas de erro, RMSE e MAE, tanto para  $C_{\text{trad}}$  quanto para  $C_{\text{final}}$ ; e
- o tempo gasto para a execução de cada método.

Os experimentos foram divididos em duas partes. Na **Subseção 7.1.1**, diversos testes foram conduzidos para a escolha de um valor adequado para o fator de amostragem,  $f_a$ . Na **Subseção 7.1.2**, a metodologia clássica e a proposta são comparadas e analisadas segundo os critérios de desempenho descritos.

### 7.1.1 Escolha do fator de amostragem

Para estes experimentos, diversos valores para o fator de amostragem, no intervalo  $1 \leq f_a \leq 100$ , foram utilizados para a execução do método proposto. Para cada

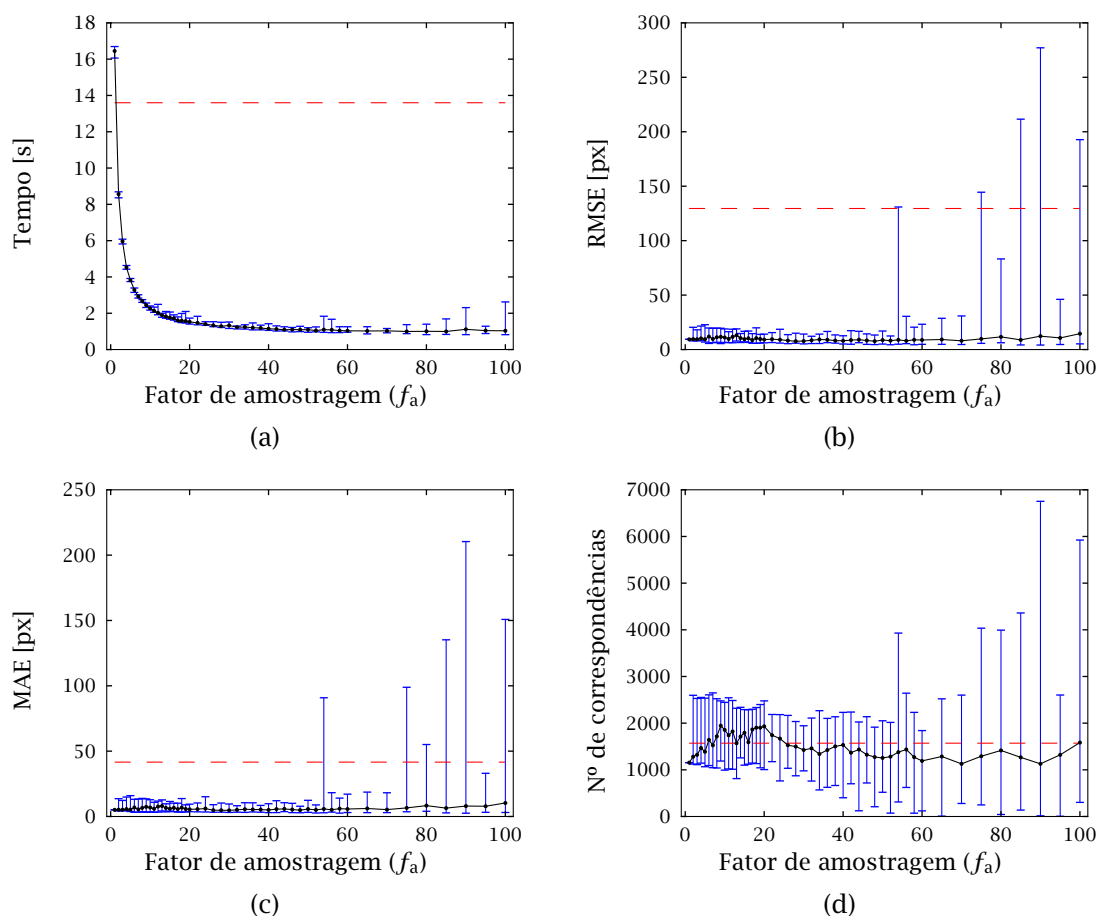


**Figura 7.2.** Resultados obtidos com valores diferentes para o fator de amostragem,  $f_a$ , para o caso de mudança de ponto de vista, apresentando estatísticas após 30 execuções: mediana (em preto) e os valores mínimos e máximos (barras de erro). A reta horizontal (tracejada em vermelho) é o resultado obtido com o método clássico. **(a)** Tempo de execução, **(b)** RMSE, **(c)** MAE e **(d)** número de correspondências ( $|C_{\text{final}}|$ ).

valor, o método foi executado 30 vezes, cada um com uma seleção aleatória do subconjunto  $Z \subseteq \mathcal{F}_p$  (Eq. (4.5)). Os resultados obtidos são apresentados nas Figuras 7.2, 7.3 e 7.4.

Nas Figuras 7.2(a), 7.3(a) e 7.4(a), pode-se observar que o tempo necessário para a execução do algoritmo decresce monotonicamente à medida que  $f_a$  cresce. Nos três casos de teste, a única circunstância em que o algoritmo proposto consumiu mais tempo do que o clássico foi com  $f_a = 1$ , ou seja, quando não houve subamostragem ( $Z = \mathcal{F}_p$ ). Este efeito é esperado e foi previamente detalhado na Seção 4.2.

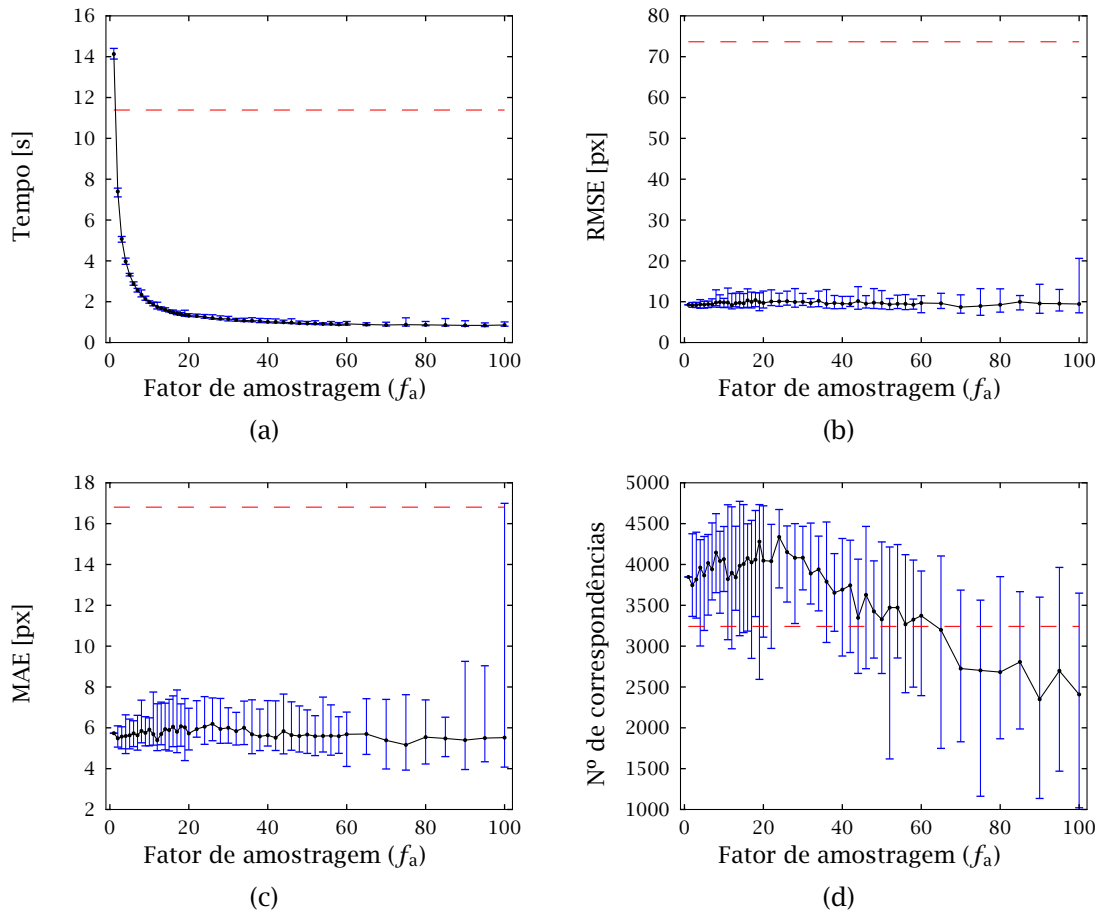
A mediana das medidas de erro, tanto do RMSE (Figuras 7.2(b), 7.3(b) e 7.4(b)) quanto do MAE (Figuras 7.2(c), 7.3(c) e 7.4(c)), apresentaram comportamentos



**Figura 7.3.** Resultados obtidos com valores diferentes para o fator de amostragem,  $f_a$ , para o caso de aproximação da câmera, apresentando estatísticas após 30 execuções: mediana (em preto) e os valores mínimos e máximos (barras de erro). A reta horizontal (tracejada em vermelho) é o resultado obtido com o método clássico. **(a)** Tempo de execução, **(b)** RMSE, **(c)** MAE e **(d)** número de correspondências ( $|C_{\text{final}}|$ ).

semelhantes: em geral, os resultados são estáveis e significativamente menores do que os obtidos com o método clássico. No entanto, para valores altos do fator de amostragem — cerca de  $f_a \geq 90$  nos casos de mudança do ponto de vista (Figuras 7.2(b)-7.2(c)) e de rotação da câmera (Figuras 7.4(b)-7.4(c)) e cerca de  $f_a \geq 50$  no caso de aproximação da câmera (Figuras 7.3(b)-7.3(c)) —, nota-se que a seleção aleatória de  $Z$  pode comprometer a qualidade dos resultados obtidos. Este fato é observado pelos picos de valores máximos tanto do RMSE quanto do MAE, gerando por vezes resultados qualitativamente piores do que os do método clássico.

Quanto ao número de correspondências (Figuras 7.2(d), 7.3(d) e 7.4(d)), em geral observa-se valores melhores em relação ao método clássico, em especial no



**Figura 7.4.** Resultados obtidos com valores diferentes para o fator de amostragem,  $f_a$ , para o caso de rotação da câmera, apresentando estatísticas após 30 execuções: mediana (em preto) e os valores mínimos e máximos (barras de erro). A reta horizontal (tracejada em vermelho) é o resultado obtido com o método clássico. **(a)** Tempo de execução, **(b)** RMSE, **(c)** MAE e **(d)** número de correspondências ( $|C_{\text{final}}|$ ).

intervalo entre  $15 \leq f_a \leq 30$ .

A partir dos resultados apresentados, foi adotado o valor  $f_a = 20$  para os demais experimentos realizados. Este valor representa um ponto de equilíbrio entre o tempo de execução, o número de correspondências e a estabilidade das medidas de erro em relação à seleção aleatória do subconjunto  $\mathcal{Z}$ .

### 7.1.2 Análise de qualidade e performance

Para o fator de subamostragem adotado,  $f_a = 20$ , os resultados comparativos obtidos tanto para o número de correspondências ( $|C_{\text{trad}}|$  vs.  $|C_{\text{final}}|$ ) e para as medidas de erro (RMSE e MAE) são apresentados na [Tabela 7.1](#). Todos os resultados

**Tabela 7.2.** Comparações de tempo entre o método clássico (com  $\tau_M = 1,5$ ) e o método proposto. O tempo consumido por cada passo do método proposto também é apresentado.

<i>Método / passo</i>	<i>Caso de teste</i>	<i>Tempo de execução [s]</i>		
		<i>Mudança do ponto de vista</i>	<i>Aproximação da câmera</i>	<i>Rotação da câmera</i>
Método clássico		7,86	13,49	11,20
Método proposto		0,86	1,06	0,98
	Correspondência inicial	0,32	0,54	0,46
	Cálculo das restrições geométricas	0,28	0,23	0,21
	Correspondência final	0,27	0,29	0,31
	Comparação entre os métodos	9,1×	12,7×	11,4×

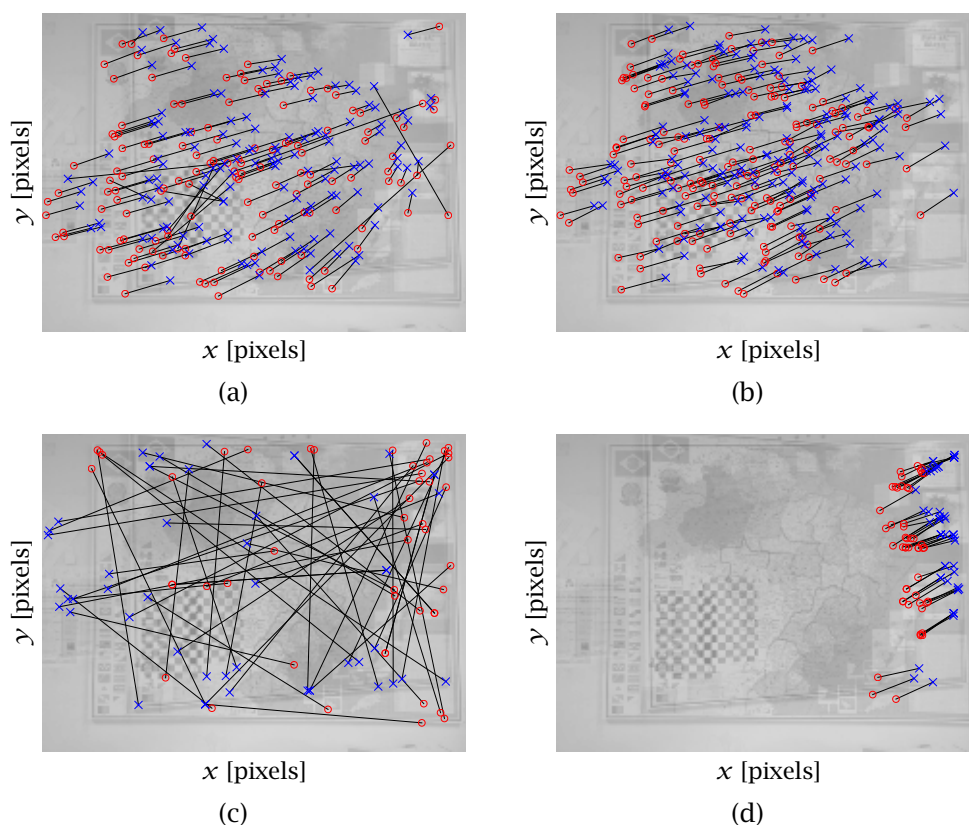
demonstram aumento sistemático no número de correspondências (cerca de 13%-31%) e um decréscimo expressivo tanto nas medidas de RMSE quanto de MAE (para menos de 1/6 no caso do RMSE e para menos de 1/2 no caso do MAE).

A Tabela 7.2 apresenta o tempo necessário para calcular as correspondências nos casos de teste estudados. Os resultados demonstram que o método proposto é aproximadamente 10 vezes mais rápido do que o método clássico. O passo mais demorado é a avaliação da correspondência inicial, responsável por cerca de 37%-51% do tempo total nos testes realizados. Esse passo seria muito mais demorado se a otimização proposta na Eq. (4.6) não tivesse sido usada.

A melhoria na qualidade da correspondência também pode ser visualmente percebida nas Figuras 7.5, 7.6 e 7.7. Para cada uma dessas figuras, foram plotadas 5% das correspondências (primeira linha das imagens) e as 50 piores correspondências, isto é, aquelas com os 50 valores mais altos para  $\|\vec{\epsilon}_{i,j}\|$  (segunda linha). No primeiro caso, foram plotadas somente 5% das correspondências para evitar a sobrecarga visual que seria causada pela plotagem de todos os milhares de correspondências.

Algumas correspondências espúrias podem ser observadas nas Figuras 7.5(a), 7.6(a) e 7.7(a). Por outro lado, nenhuma correspondência incoerente pode ser observada nas Figuras 7.5(b), 7.6(b) e 7.7(b). Além disso, todas as 50 piores correspondências observadas nas Figuras 7.5(c), 7.6(c) e 7.7(c) são claramente falsas correspondências, enquanto as apresentadas nas Figuras 7.5(d), 7.6(d) e 7.7(d) ainda seguem o padrão geral de transformação geométrica. Mesmo assim, é importante notar que todas as piores correspondências do método proposto ocorrem consistentemente em regiões distantes do padrão de calibração xadrez — o que pode indicar que a homografia não representa corretamente a transformação ocorrida nessas regiões, por causa das distorções das lentes da câmera.



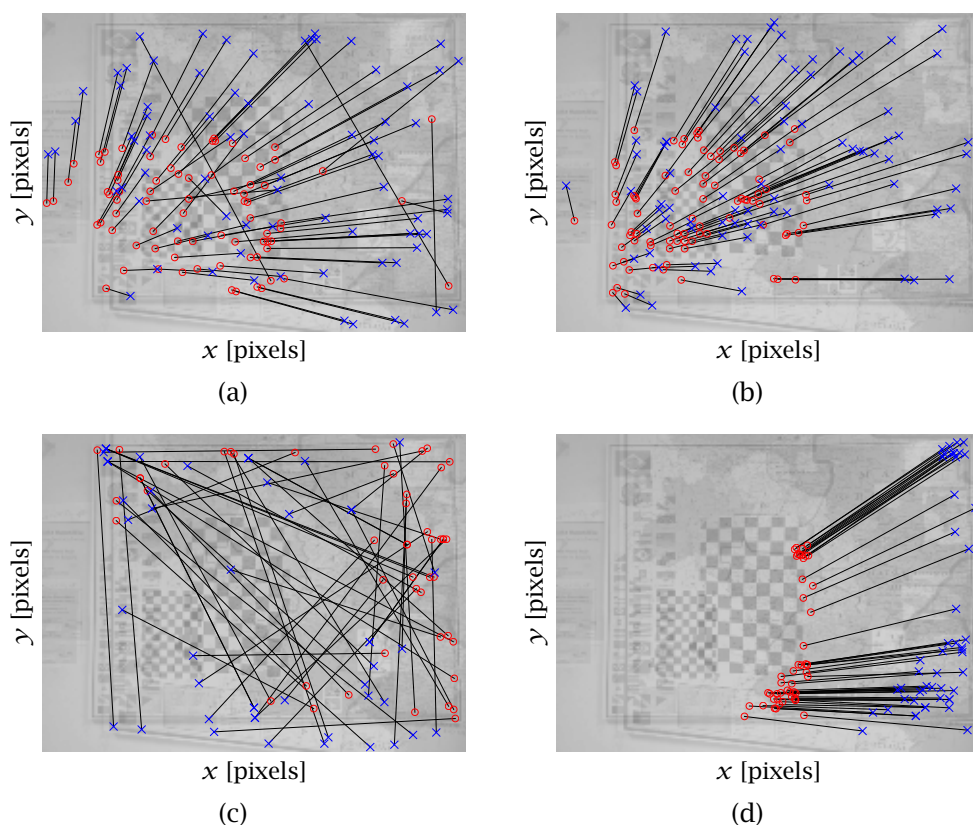


**Figura 7.5.** Correspondência para o caso de mudança de ponto de vista (da Figura 7.1(a) para a 7.1(b)): 5% das correspondências obtidas (a) com o método tradicional com  $\tau_M = 1,5$  e (b) com o método proposto; 50 piores correspondências obtidas (c) com o método tradicional com  $\tau_M = 1,5$  e (d) com o método proposto.

A Figura 7.8 apresenta a distribuição dos vetores de erro,  $\vec{\epsilon}_{i,j}$  para todos os casos de teste. Todos os experimentos demonstram que o método proposto confina os vetores de erro em intervalos bem menores do que os do método tradicional: cerca de uma ordem de magnitude menor em todos os casos. (Note-se a mudança de escala nas figuras.)

### 7.1.3 Conclusões

O método proposto para a correspondência de pontos salientes permite realizar a tarefa proposta de maneira mais acurada e com um custo computacional consideravelmente menor. Em comparação com os métodos tradicionais baseados somente na similaridade dos vetores descritores, a abordagem apresentada neste trabalho é capaz de reduzir o número de falsas correspondências, o que foi evidenciado pela redução significativa das medidas de erro avaliadas (RMSE e MAE). Como o RMSE é

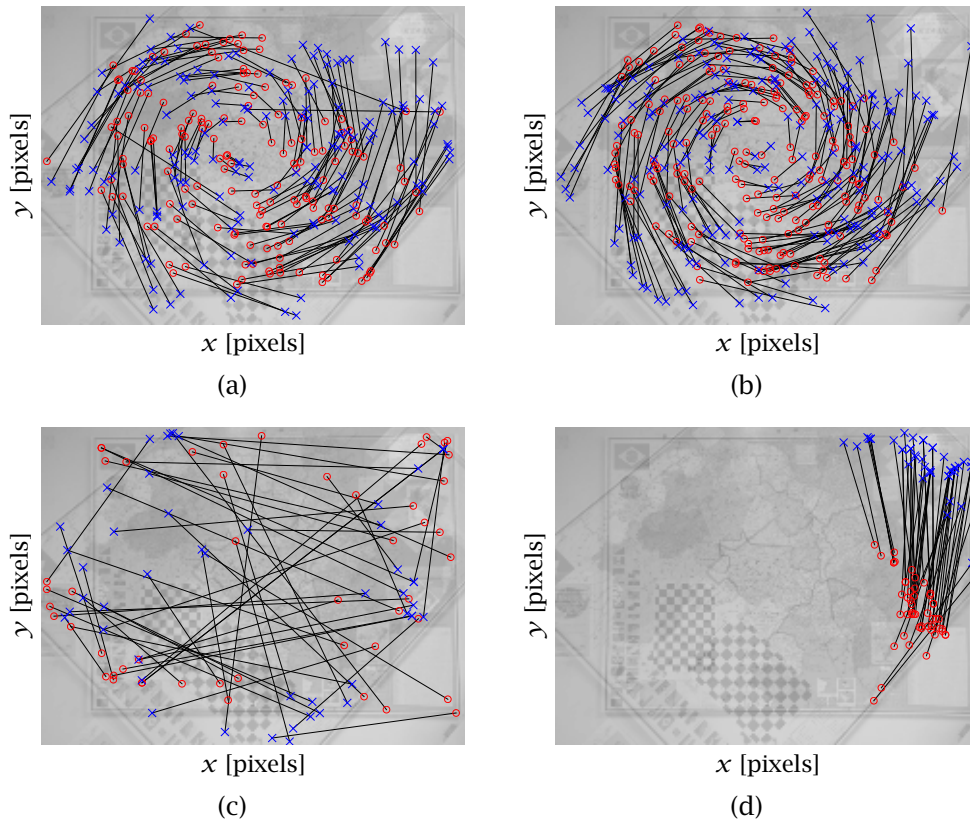


**Figura 7.6.** Correspondência para o caso de aproximação da câmera (da Figura 7.1(a) para a Figura 7.1(c)): 5% das correspondências obtidas **(a)** com o método tradicional com  $\tau_M = 1,5$  e **(b)** com o método proposto; 50 piores correspondências obtidas **(c)** com o método tradicional com  $\tau_M = 1,5$  e **(d)** com o método proposto.

particularmente sensível a valores espúrios, a redução expressiva na magnitude de seus valores calculados, quando comparados com os obtidos com os métodos tradicionais, é uma clara indicação de que a ocorrência de falsas correspondências é bem menos frequente com o método proposto.

Por outro lado, o aumento observado no número de correspondências sugere que os métodos clássicos descartam várias associações válidas entre pares de pontos são descartadas como consequência do uso de um limiar de distinguibilidade. Esse efeito indesejado não é evidenciado no método proposto, já que tal limiar não é aplicado.

Finalmente, observa-se que é possível a avaliação do conjunto de correspondências utilizando apenas cerca de um décimo do tempo consumido pelos métodos clássicos. No que diz respeito a futuras abordagens para otimização e redução do tempo de execução, nota-se que todas as etapas do método proposto são parale-

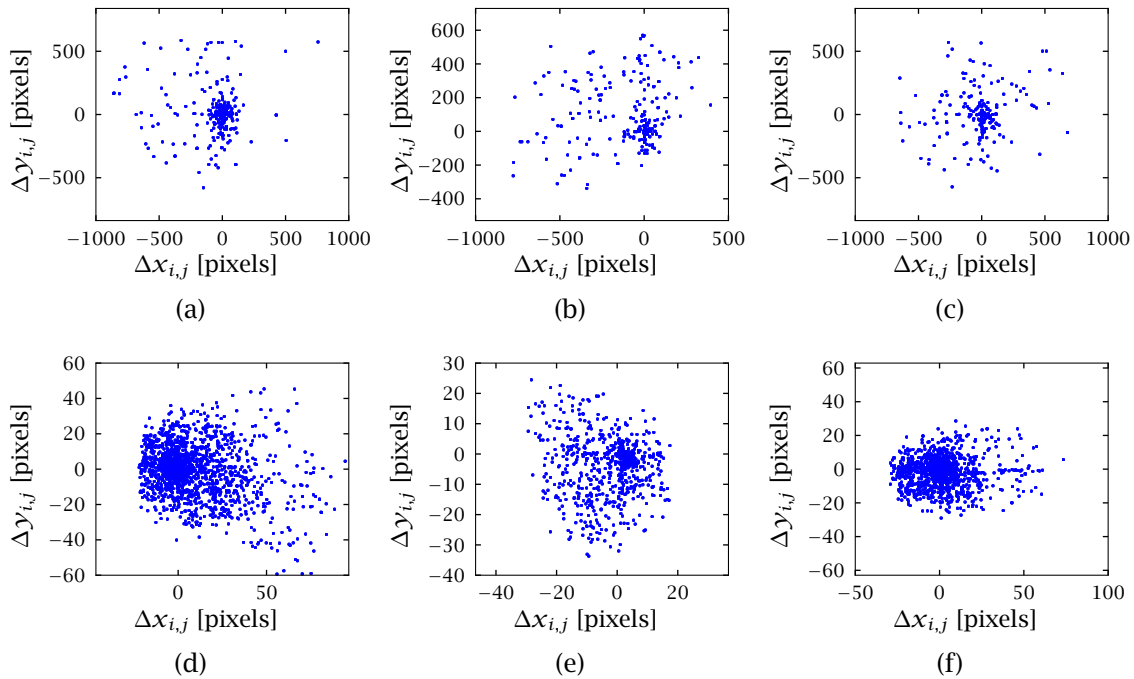


**Figura 7.7.** Correspondência para o caso de rotação da câmera (da Figura 7.1(a) para a Figura 7.1(d)): 5% das correspondências obtidas **(a)** com o método tradicional com  $\tau_M = 1,5$  e **(b)** com o método proposto; 50 piores correspondências obtidas **(c)** com o método tradicional com  $\tau_M = 1,5$  e **(d)** com o método proposto.

lizáveis e facilmente implementáveis em arquiteturas *multi-core*, a exemplo das GPUs, permitindo a execução do algoritmo sobre sequências de imagens em tempo real.

A principal limitação do método proposto encontra-se no fato de que todas as transformações consideradas (escala, rotação e translação) são confinadas a um único intervalo. Essa restrição não se aplica ao caso geral em que as cenas não são estáticas ou quando ocorre uma mudança significativa do ponto de vista da câmera entre a aquisição do par de imagens. Em ambos os casos, é possível melhorar o método proposto pela aceitação de mais de um intervalo para cada transformação geométrica, efetivamente segmentando a imagem em regiões nas quais as transformações geométricas são localmente consistentes.

Embora esta generalização esteja sendo atualmente em desenvolvimento, é importante notar que ela não é essencial para a validade da metodologia geral



**Figura 7.8.** Vetores de erro de homografia ( $\vec{\varepsilon}_{i,j}$ ) para a mudança de ponto de vista (primeira coluna), para a aproximação da câmera (segunda coluna) e para a rotação da câmera (terceira coluna). Primeira linha: método tradicional com  $\tau_M = 1,5$ ; segunda linha: método proposto. Deve-se observar a mudança de cerca de uma ordem de magnitude nas escalas dos eixos dos gráficos da primeira coluna (método tradicional) para a segunda coluna (método proposto).

proposta neste trabalho, já que se considera que a cena é estática e que as imagens são obtidas a intervalos pequenos de movimentação do robô.

## 7.2 Módulos de manutenção de marcos e de SLAM

Os módulos de manutenção de marcos e de SLAM são fortemente interligados, já que o primeiro mantém a lista de marcos resolvidos e não resolvidos (Subseção 5.2.4) e o segundo provê os dados necessários para a avaliação e o descarte de hipóteses (Subseção 5.2.5) e marcos (Subseções 5.2.6 e 5.2.7). Desta maneira, a implementação dos dois módulos foi efetuada como um conjunto de rotinas integradas.

Os algoritmos foram implementados em Matlab, com algumas rotinas auxiliares escritas em C++. A plataforma utilizada nos testes consiste em um processador Intel® Core™ i7 920 de 2,67 GHz com 12 GB de RAM, executando um sistema operacional GNU/Linux Ubuntu de 64 bits (com *kernel* de versão 2.6.28).

### 7.2.1 Nota a respeito da exibição das nuvens de pontos

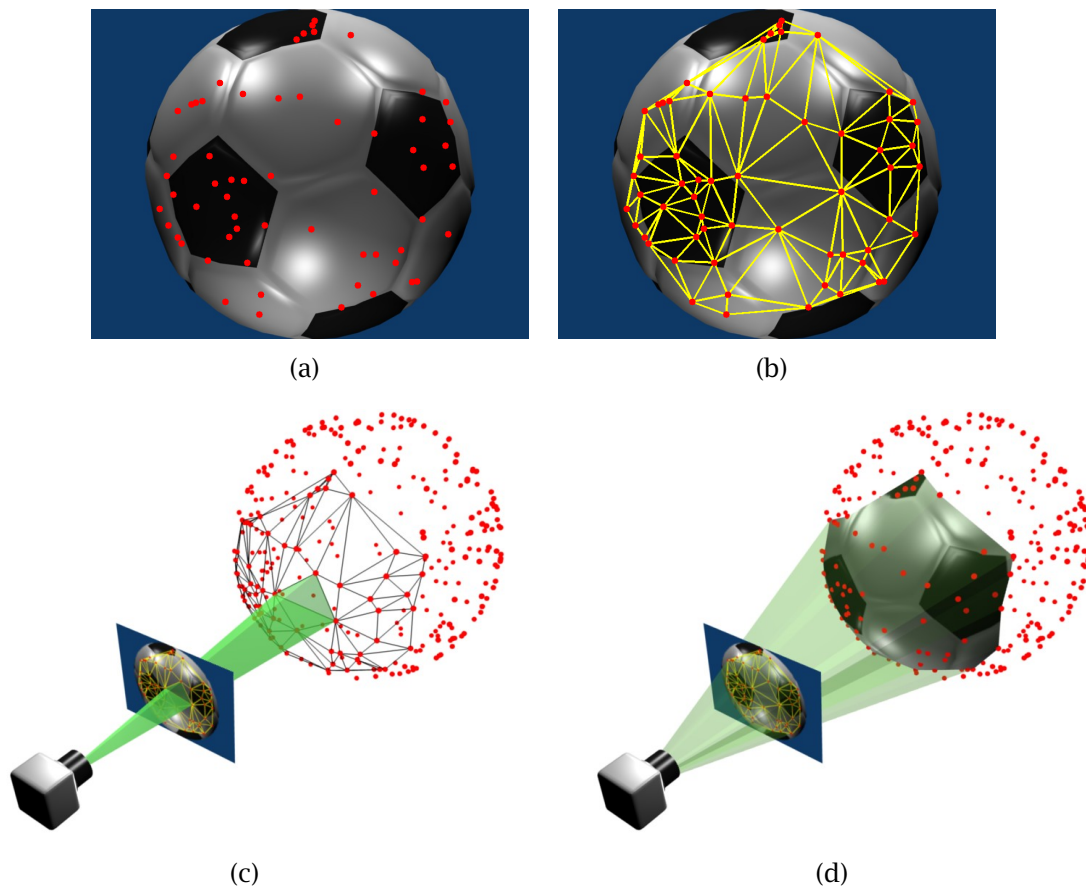
É importante ressaltar que o processo de reconstrução desenvolvido neste trabalho tem por objetivo a geração de uma nuvem de pontos que representa a superfície dos objetos reconstruídos, de acordo com a definição do Problema Principal lançado na [Seção 1.3](#). Entretanto, a *visualização* de uma nuvem de pontos tridimensional em mídias bidimensionais (para efeitos de impressão neste caso) é no mínimo confusa, pois não é possível transmitir a sensação de profundidade dos pontos — o que torna praticamente impossível para o leitor realizar qualquer avaliação visual qualitativa dos resultados obtidos.

Para contornar este problema, as reconstruções serão apresentadas como renderizações de superfícies tridimensionais texturizadas, cuja apreciação é muito mais simples. No caso dos experimentos simulados, os resultados visuais exibidos no restante deste capítulo foram construídos da seguinte maneira:

1. A nuvem de pontos obtida por cada experimento foi processada pelo algoritmo de triangulação de Delaunay [Delaunay, 1934] adaptado para três dimensões, gerando tetraedros cujos vértices coincidem com pontos da nuvem. Este conjunto de tetraedros é uma representação geométrica sólida imprecisa do objeto reconstruído;
2. Em seguida, a textura utilizada original utilizada para a geração das imagens de entrada foi aplicada sobre o sólido gerado;
3. Finalmente, o objeto virtual com as texturas aplicadas foi renderizado pelo aplicativo gráfico *Blender*, versão 2.49b.

No caso dos experimentos reais, o procedimento possui algumas particularidades, ilustradas na [Figura 7.9](#):

1. A partir de cada imagem de entrada obtida pela câmera ([Figura 7.9\(a\)](#)), o conjunto de pontos salientes detectado foi processado pelo algoritmo original de triangulação de Delaunay [Delaunay, 1934], gerando um conjunto de triângulos no plano de imagem ([Figura 7.9\(b\)](#));
2. Cada triângulo obtido no passo anterior — incluindo o correspondente recorte da imagem — foi utilizado para criar uma face triangular texturizada no espaço triangular, com base na correspondência entre pontos salientes e marcos ([Figuras 7.9\(c\)-\(d\)](#));
3. Finalmente, o objeto virtual com as texturas aplicadas foi renderizado pelo aplicativo gráfico *Blender*, versão 2.49b.



**Figura 7.9.** Algoritmo para visualização renderizada dos resultados: **(a)** Possível imagem capturada pela câmera em um determinado instante  $t$ , exibindo os pontos salientes detectados; **(b)** Resultado da triangulação de Delaunay. Cada triângulo limita um recorte da imagem que será usado como textura para a reconstrução; **(c)** Criação dos triângulos tridimensionais e projeção dos recortes da imagem sobre a nuvem de pontos, destacando o registro de um recorte triangular; **(d)** Região da superfície reconstruída a partir da imagem.

Este trabalho não tem por objetivo apresentar inovações no processo de registro de imagens ou mesmo na recriação de sólidos a partir de nuvens de pontos. De fato, ambos os assuntos têm sido ativamente estudados, pois ainda não foram resolvidos em geral.

Mais importante, os algoritmos descritos acima são soluções simples e não representam o que há de mais inovador na área. Por conseguinte, as imagens renderizadas por este processo podem apresentar alguns artefatos indesejados e não relacionados à nuvem de pontos reconstruída, em particular:

- o preenchimento de algumas regiões de concavidade com partes sólidas, incluindo a aplicação de texturas não condizentes com a geometria do objeto;



**Figura 7.10.** Configuração experimental para o caso de testes ZOGIS: **(a)** Vista frontal da caixa que representa o objeto de interesse; **(b)** Vista da câmera superior durante a coleta de dados para o experimento. Note-se o marco fiducial acoplado à câmera, que permite a estimação inicial do deslocamento relativo entre vistas consecutivas,  $\mathbf{u}_t$  (definido na Subseção 5.3.2); **(c)** Algumas imagens da sequência de entrada para o caso de teste em questão.

- algumas variações perceptíveis de intensidade entre regiões de pixels, como consequência da falta de qualquer ajuste radiométrico das imagens utilizadas como textura.

### 7.2.2 Caso de teste: ZOGIS

O conjunto de testes descrito a seguir, que será identificado por ZOGIS, teve como objetivo principal a avaliação quantitativa do processo de reconstrução. Não tem como objetivo, portanto, a avaliação do processo de planejamento que encerra o caráter autônomo deste trabalho.

Os testes foram realizados sobre um conjunto de imagens reais da caixa de um produto comercial (Figura 7.10(a)), medindo aproximadamente  $332 (L) \times 235 (A) \times 83 (P)$  mm. As imagens foram capturadas por uma máquina fotográfica digital *Canon PowerShot SX10 IS* com 10 megapixels, configurada para captura de filme (imagens de  $640 \times 480$  pixels a 30 quadros por segundo em codificação *Apple QuickTime*). O FoV horizontal da câmera é de  $62^\circ$  e vertical de  $48^\circ$ , aproximadamente. Algumas imagens da sequência podem ser vistas na Figura 7.10(c).

A movimentação da câmera foi feita por um operador humano. Para prover as informações necessárias para a fase de predição do processo de estimação — ou seja, o deslocamento relativo entre as poses da câmera,  $\mathbf{u}_t$  (ver definição na Subseção 5.3.2) —, um marco fiducial foi fixado na câmera e rastreado por uma segunda câmera, esta fixa e posicionada sobre a cena (Figura 7.10(b)). O sistema de marcos fiduciais publicado por da Camara Neto et al. [2010] foi adotado para realizar o rastreamento do marco fiducial.

É importante ressaltar duas informações sobre as informações obtidas pela segunda câmera:

1. Apenas o *deslocamento relativo* das poses estimadas do marco fiducial entre imagens consecutivas foi utilizado no processo de estimação. Em outras palavras, a *pose absoluta* dos marcos não foi usada como dado de entrada;
2. Nenhuma informação das imagens desta segunda câmera foi utilizada para a estimação da geometria do objeto reconstruído, mantendo fiel o caráter da reconstrução monocular pela primeira câmera.

Para os experimentos realizados, os parâmetros da SoG (Figura 5.6) foram assim definidos:

- razão de incerteza  $\alpha_{\text{hip}} = 0,3$ ;
- razão geométrica de distribuição  $\beta_{\text{hip}} = 2$ ; e
- distância mínima  $d_{\text{min}} = 30$  mm e máxima  $d_{\text{max}} = 1\,000$  mm,

de onde se deduz que o número inicial de hipóteses criadas para cada um dos pontos salientes é  $H = 5$ .

Para evitar carga excessiva de processamento advinda da criação massiva de hipóteses, este caso de testes a quantidade máxima de marcos não resolvidos foi limitada em 80.





**Figura 7.11.** Visualizações renderizadas da geometria reconstruída no caso de teste ZOGIS.

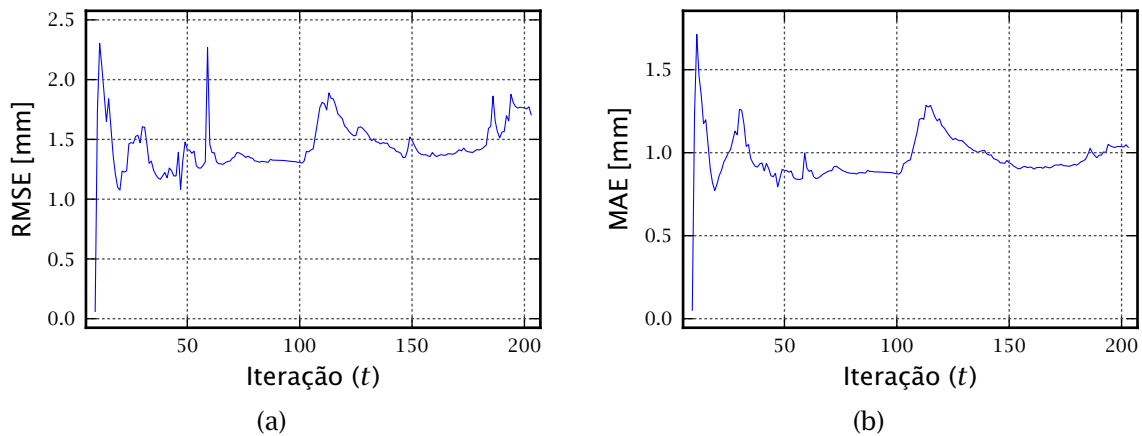
### 7.2.3 Resultados e discussões

Algumas imagens renderizadas da geometria recuperada podem ser vistas na 7.11. Este conjunto de imagens foi gerado de acordo com o algoritmo descrito na Subseção 7.2.1.

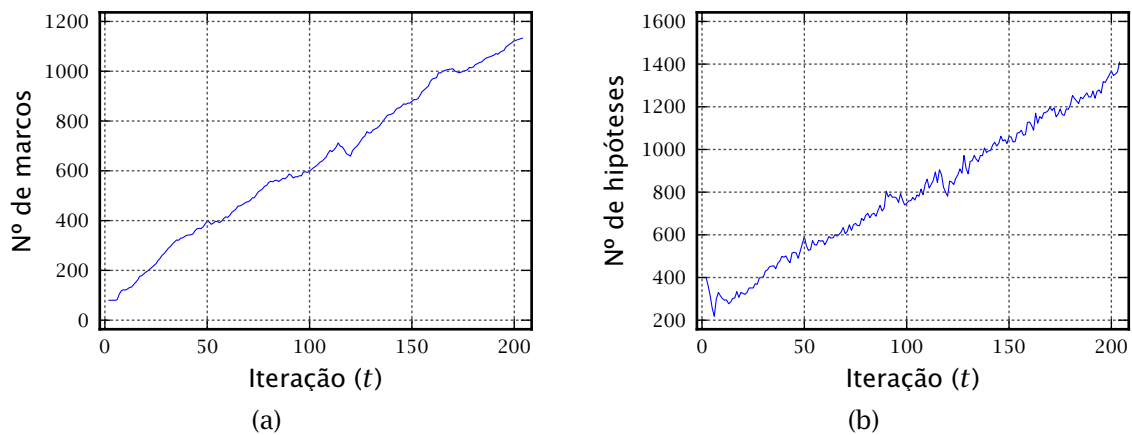
A análise da nuvem de pontos recuperada mostra que os pontos de vista adotados pelo operador forneceram informações suficientes para a recuperação (parcial) de duas faces: a frontal e a superior, ambas visíveis na Figura 7.10(a). Como essas faces são aproximadamente planas, elas podem ser modeladas como um par de planos ortogonais no espaço tridimensional. Assim, uma abordagem adequada para a avaliação quantitativa dos resultados obtidos consiste em buscar o melhor alinhamento entre esses planos e a nuvem de pontos estimada, e quantificar as distâncias entre cada marco e esses planos. Para consolidar esses desvios, foram utilizadas novamente as medidas de RMSE e MAE.

Os resultados são exibidos nas Figuras 7.12(a) e 7.12(b). A característica mais relevante dos dados exibidos é a manutenção das métricas de erro em níveis mais ou menos constantes. Outro aspecto importante está no valor absoluto dessas medidas de erro: observa-se que os valores de MAE são mantidos em geral abaixo de 1 mm, enquanto os valores de RMSE (estes mais sensíveis a *outliers*) não ultrapassam 1,5 mm na maior parte do tempo. Para efeitos de comparação, vale lembrar as dimensões do objeto observado: 332 (L)  $\times$  235 (A)  $\times$  83 (P) mm.

Outro aspecto importante a ser analisado é a evolução na quantidade de



**Figura 7.12.** Análise quantitativa dos resultados para o caso de teste ZOGIS: **(a)** *Root Mean Square Error* (Erro Quadrático Médio), **(b)** *Mean Absolute Error* (Média dos Erros Absolutos).

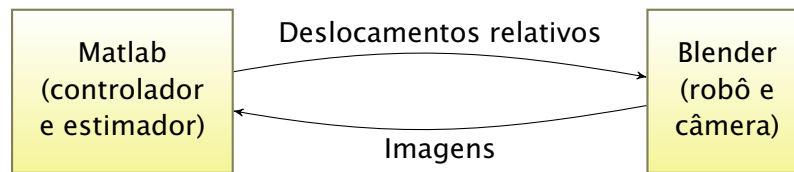


**Figura 7.13.** Contagem de marcros resolvidos e hipóteses para o caso de teste ZOGIS: **(a)** Quantidade total de marcros resolvidos, **(b)** quantidade total de hipóteses.

marcros e hipóteses ao longo do tempo. Essas duas informações são apresentadas nas Figuras 7.13(a) e 7.13(b).

Alguns fatos interessantes podem ser observados:

- A quantidade de marcros (Figura 7.13(a)) cresce de maneira aproximadamente linear, evidenciando um fluxo constante de novos marcros a cada novo ponto de vista. Eventualmente há uma redução na quantidade de marcros (evidenciado por trechos de derivada negativa no gráfico). Este fenômeno é esperado e causado por dois processos: o descarte de marcros pouco observados, antes de sua promoção a marcros resolvidos (Subseção 5.2.6); e a eliminação de marcros espúrios, isto é, aqueles que, uma vez promovidos, são geometricamente



**Figura 7.14.** Ilustração da interação entre as plataformas Matlab e Blender adotado nos experimentos simulados. O Blender simula o robô, recebendo os comandos de deslocamentos relativos do planejador e aplicando-os sobre a câmera virtual que representa o sensor do robô, e retorna as imagens renderizadas como se estas tivessem sido obtidas por uma câmera real.

inconsistentes com suas observações posteriores (Subseção 5.2.7);

- A evolução da quantidade de hipóteses (Figura 7.13(b)) também segue um comportamento geral de crescimento linear, porém sensivelmente mais flutuante quando comparada à quantidade de marcos. De fato, o descarte de hipóteses (Subseção 5.2.5) é contínuo e ocorre a cada iteração do processo.

## 7.3 Avaliação do sistema integrado

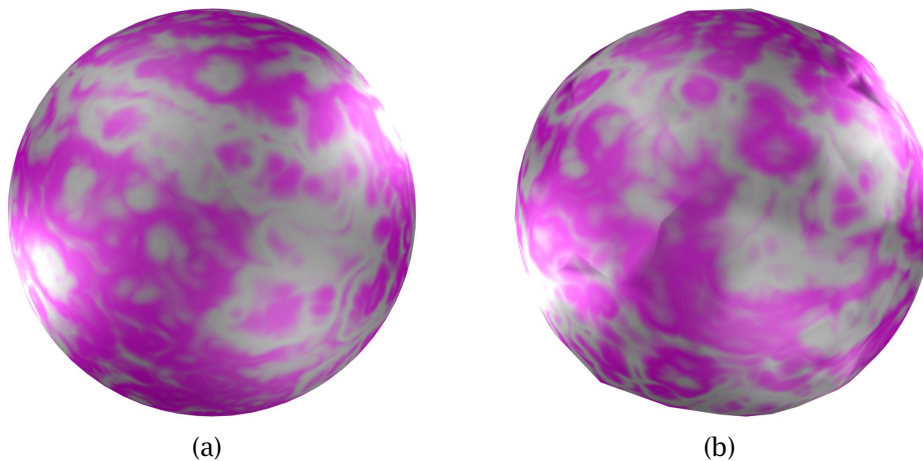
Esta bateria de testes tem por objetivo avaliar os resultados obtidos pelo sistema integrado como um todo. Desta forma, o foco dos experimentos está na avaliação de dois aspectos principais: (i) a avaliação quantitativa da reconstrução obtida, por meio de testes simulados utilizando formas geométricas conhecidas e de modelagem simples; e (ii) a avaliação da cobertura do planejador, com o objetivo de determinar se este foi capaz de determinar sequências de poses capazes de observar o objeto por inteiro.

A avaliação do sistema integrado é dividida em duas partes: testes simulados (Subseção 7.3.1) e experimentos reais (Subseção 7.3.2). No que diz respeito à exibição das reconstruções, também se aplicam as observações descritas na Subseção 7.2.1.

### 7.3.1 Testes simulados

Os testes simulados foram realizados por meio de imagens renderizadas no programa gráfico *Blender*, versão 2.49b, desenvolvido pela *Blender Foundation*.

Para a execução dos testes, foram desenvolvidas rotinas de comunicação por TCP/IP entre o Matlab e o Blender (Figura 7.14). Desta forma, o Blender foi utilizado para simular o aparato robótico com a câmera embutida. O planejador alimenta o renderizador com a sequência de deslocamentos relativos entre poses



**Figura 7.15.** Caso de teste SPHERE: **(a)** Imagem da esfera utilizada nos testes, **(b)** reconstrução obtida após 150 iterações.

sucessivas (da mesma forma que ocorre com um robô real). Esses deslocamentos são aplicados sobre a pose atual da câmera virtual do renderizador, que gera uma imagem e a retorna ao sistema principal. Esta imagem é utilizada como entrada para o processo que extrai pontos salientes.

Em todos os experimentos simulados, a câmera virtual foi configurada para capturar imagens com **FoV** aproximado de  $90^\circ$  na horizontal e  $74^\circ$  na vertical. As imagens geradas possuem resolução de  $640 \times 480$  pixels.

Para os experimentos realizados, os parâmetros da **SoG** (Figura 5.6) foram assim definidos:

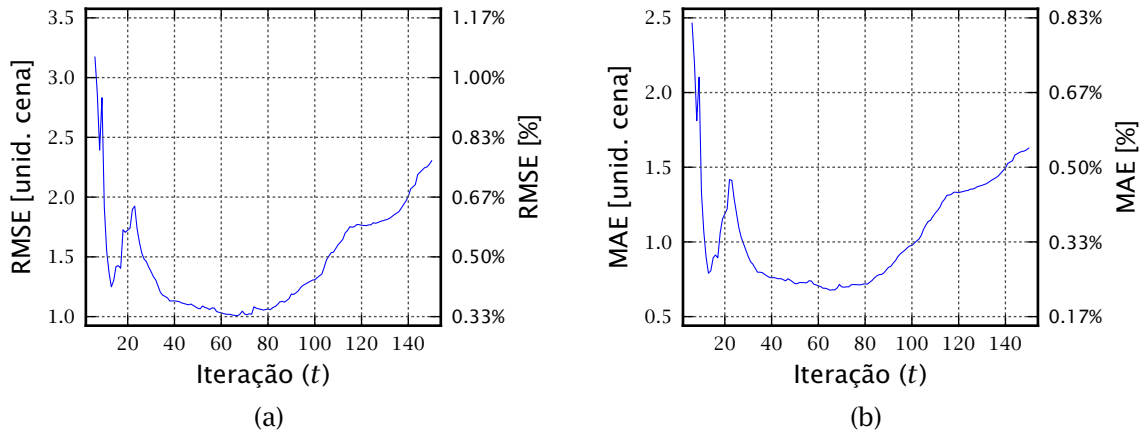
- razão de incerteza  $\alpha_{\text{hip}} = 0,3$ ;
- razão geométrica de distribuição  $\beta_{\text{hip}} = 2$ ; e
- distância mínima  $d_{\text{min}} = 100$  unidades da cena e máxima  $d_{\text{max}} = 1\,000$  mm unidades da cena,

de onde se deduz que o número inicial de hipóteses criadas para cada um dos pontos salientes é  $H = 4$ .

Para evitar carga excessiva de processamento advinda da criação massiva de hipóteses, este caso de testes a quantidade máxima de marcos não resolvidos foi limitada em 80.

### 7.3.1.1 Caso de teste: SPHERE

O primeiro caso de teste simulado será identificado por SPHERE. O objeto de interesse é uma esfera virtual com 300 unidades da cena, cuja imagem de textura



**Figura 7.16.** Análise quantitativa dos resultados para o caso de testes SPHERE: **(a)** *Root Mean Square Error* (Erro Quadrático Médio), **(b)** *Mean Absolute Error* (Média dos Erros Absolutos). A escala percentual (eixo à direita) é relativa ao diâmetro da esfera (300 unidades da cena), facilitando a análise das informações em comparação ao tamanho do objeto.

é um fractal gerado pela rotina “Marble” (“Mármore”) do Blender (Figura 7.15(a)). Esta textura foi escolhida por ser capaz de gerar em torno de centenas de pontos salientes distintos pelo algoritmo SIFT quando observada de qualquer ângulo.

Este caso de teste foi concebido para avaliar a precisão da reconstrução obtida. Dado que os pontos da superfície do objeto de interesse encontram-se todos na superfície da esfera, foi adotada a avaliação da reconstrução descrita a seguir:

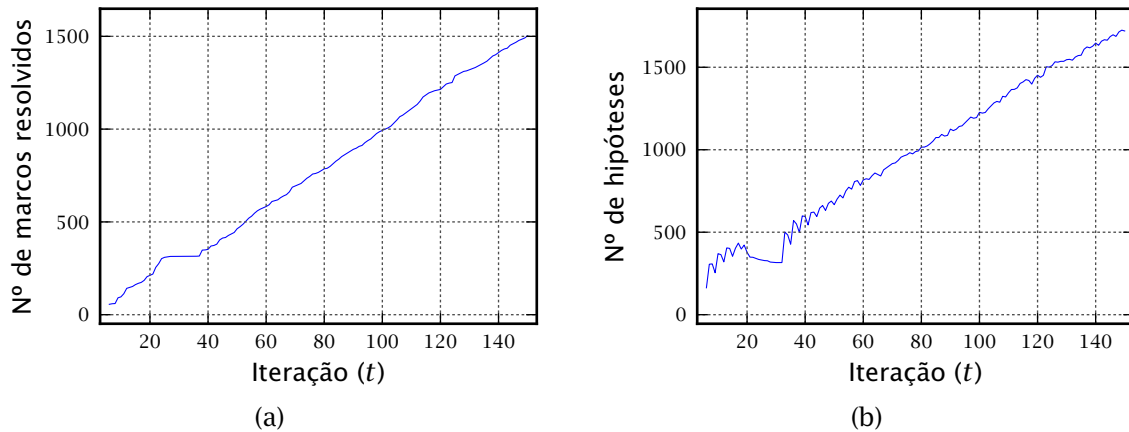
1. Encontrar o centro da esfera que minimiza o somatório das distâncias euclidianas quadráticas entre a superfície da esfera e cada marco resolvido, a partir do raio da esfera  $r = 150$ :

$$(x_c, y_c) = \arg \min \sum_m [\|\mathbf{m}_t^m - (x_c, y_c)\| - r]^2; \quad (7.5)$$

2. Utilizar, como valor de medida de erro para cada marco  $m$ , a distância entre suas coordenadas e a esfera centrada no ponto estimado pela Eq. (7.5).

Para o processo de minimização visto na Eq. (7.5) foi utilizado o algoritmo de Levenberg-Marquardt [Levenberg, 1944; Marquardt, 1963].

**Resultados e discussões.** A progressão temporal das métricas de erro é vista nas Figuras 7.16(a)–(b). Para facilitar a análise dos valores plotados, cada gráfico está também marcado com uma escala de medidas relativas ao diâmetro da esfera



**Figura 7.17.** Contagem de marcros resolvidos e hipóteses para o caso de teste SPHERE: **(a)** Quantidade total de marcros resolvidos, **(b)** quantidade total de hipóteses.

(300 unidades da cena). Observa-se que os valores de MAE são mantidos em geral abaixo de 1,5 unidades da cena (0,50% do diâmetro da esfera), enquanto os valores de RMSE não ultrapassam 2,0 unidades da cena (0,67% do diâmetro da esfera) na maior parte do tempo.

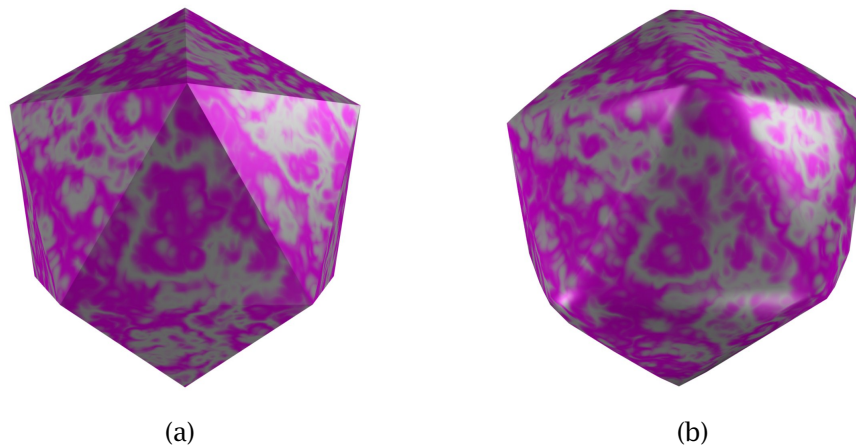
As progressões das quantidades de marcros resolvidos e hipóteses podem ser vistas respectivamente nas Figuras 7.17(a) e 7.17(b). Um aspecto interessante desses gráficos, especialmente quando comparados com os da Figura 7.16, está no crescimento perceptível das medidas de erro no intervalo aproximado  $10 \leq t \leq 25$ . Neste intervalo houve a incorporação de alguns marcros espúrios no processo de estimação (o que se reflete nas métricas de erro) que foram posteriormente eliminados pela etapa de detecção e eliminação de marcros espúrios (Subseção 5.2.7) — evidenciado, no período de tempo seguinte até aproximadamente  $t = 35$ , pela queda no número de hipóteses na Figura 7.17(b) e pela falta de novos marcros resolvidos na Figura 7.17(a).

### 7.3.1.2 Caso de testes: ICOSAHEDRON

O segundo caso de testes simulado foi concebido para avaliar a completude do algoritmo de planejamento. Espera-se que o planejador seja capaz de levar a câmera a um conjunto de pontos de vista que forneçam informações suficientes para a reconstrução completa do objeto.

Este caso de teste, identificado por ICOSAHEDRON, tem como objeto de interesse um icosaedro<sup>a</sup> com tamanho tal que ele se inscreve em uma esfera de

<sup>a</sup>Um *icosaedro* é um poliedro regular formado por 20 faces triangulares equilaterais, com 12



**Figura 7.18.** Caso de teste ICOSAEDRON: **(a)** Imagem do icosaedro utilizado nos testes, **(b)** reconstrução obtida após 200 iterações.

300 unidades da cena. O objeto pode ser visto na Figura 7.18(a) e sua textura é gerada pela mesma rotina fractal “Marble” utilizada para o caso de teste SPHERE.

Para cada iteração de reconstrução parcial do objeto, a reconstrução foi avaliada de acordo com o algoritmo seguinte:

1. Parametrizar o icosaedro como um conjunto de 20 planos no espaço tridimensional, cada um correspondente a uma face;
2. A partir dessa parametrização, encontrar uma translação e uma rotação tridimensionais para a nuvem de pontos, de forma a minimizar o somatório das distâncias euclidianas quadráticas entre cada marco resolvido e o plano mais próximo;
3. Utilizar, como valor de medida de erro para cada marco  $m$ , a distância entre suas coordenadas (após a aplicação da translação e da rotação estimadas sobre a nuvem de pontos) e o plano mais próximo.

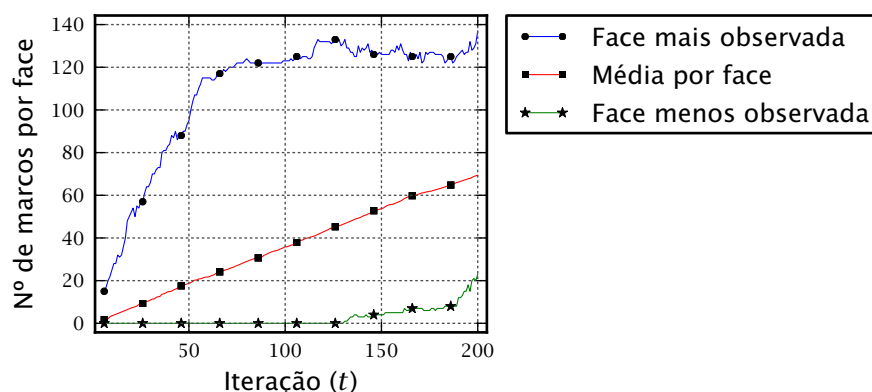
Para a estimação da transformação rígida (translação + rotação) sobre a nuvem de pontos, foi utilizado o algoritmo de Levenberg-Marquardt [Levenberg, 1944; Marquardt, 1963].

Para este caso de testes, a métrica de completude da reconstrução baseia-se na contagem do número de faces observadas. Desta forma, a observação de todas as 20 faces indica que a câmera foi levada a uma sequência de pontos de vista capazes de observar integralmente o objeto.

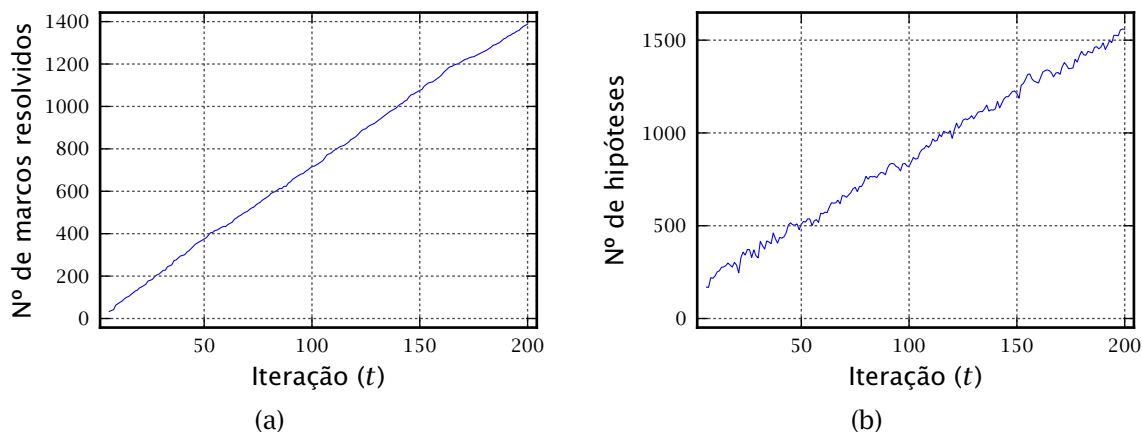
Para realizar a contagem de faces observadas, nota-se que o algoritmo descrito acima associa cada marco a uma das faces do icosaedro: um marco está 

---

vértices e 30 lados.



**Figura 7.19.** Evolução do número de marcos por face para o caso de teste ICOSAHEDRON. O gráfico registra a quantidade de marcos associados à face mais observada e à menos observada, além da média da quantidade de marcos associados a todas as faces.



**Figura 7.20.** Contagem de marcos resolvidos e hipóteses para o caso de teste ICOSAHEDRON: **(a)** Quantidade total de marcos resolvidos, **(b)** quantidade total de hipóteses.

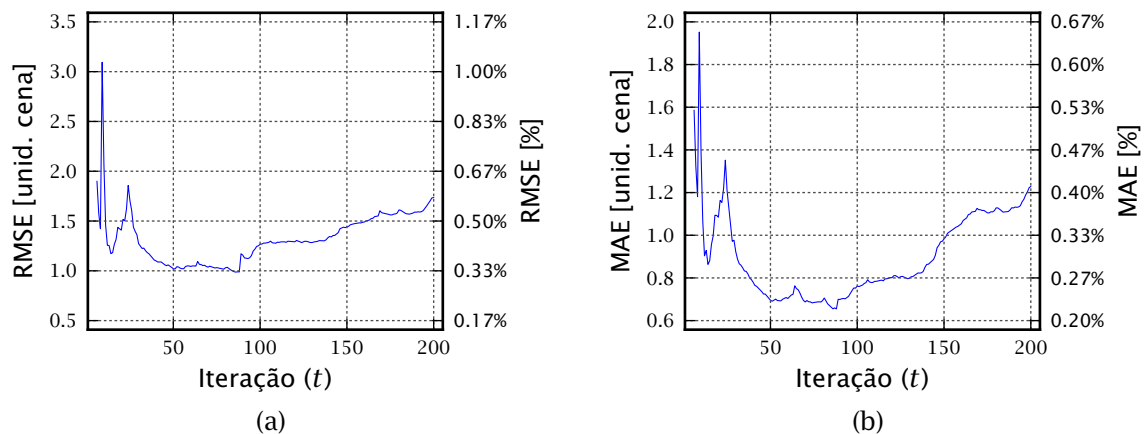
“associado” à face mais próxima. Esta associação será usada como base para contar a quantidade de marcos que representam cada face do icosaedro.

**Resultados e discussões.** Os resultados mais relevantes desta avaliação são exibidos na Figura 7.19. Por questões de clareza na plotagem dos dados, foram plotados somente a quantidade de marcos associados à face mais observada e à menos observada, além da média de marcos por face, para referência comparativa.

Os aspectos de interesse extraídos dos resultados obtidos por este caso de teste podem ser assim resumidos:

1. A quantidade de marcos associados à face mais observada cresce rapidamente





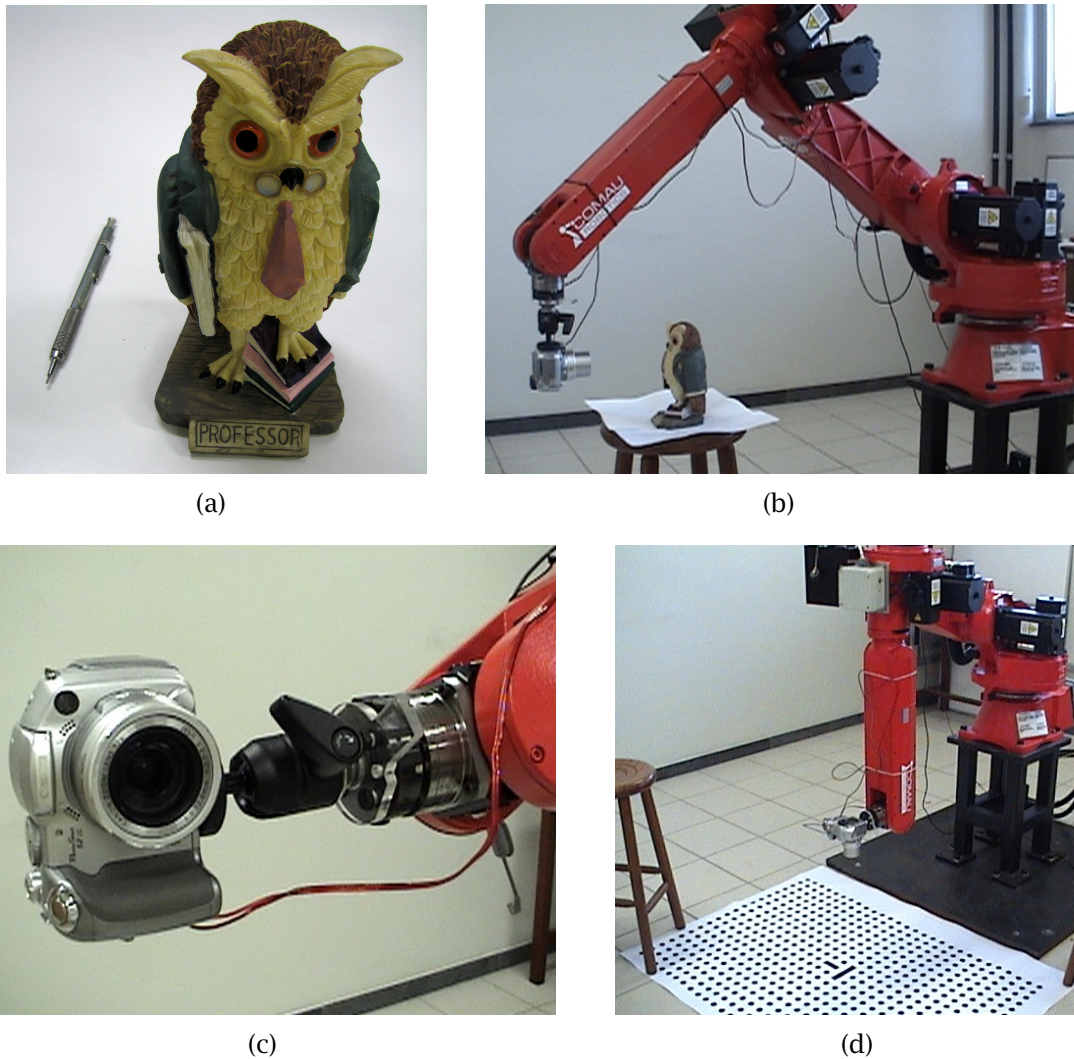
**Figura 7.21.** Medidas de erro para o caso de teste ICOSAHEDRON: **(a)** *Root Mean Square Error* (Erro Quadrático Médio), **(b)** *Mean Absolute Error* (Média dos Erros Absolutos). A escala percentual (eixo à direita) é relativa ao diâmetro da esfera que circunscribe o icosaedro (300 unidades da cena), facilitando a análise das informações em comparação ao tamanho do objeto.

e estabiliza na ordem de 150 marcos. A flutuação dos valores máximos se deve a dois fatores: (i) a eliminação de marcos espúrios e (ii) a presença de marcos próximos às arestas, que podem ser associados a uma ou outra face a cada iteração;

2. A última face passa a ser observada aproximadamente a partir da iteração  $t = 130$ , e a quantidade de marcos associada cresce até o encerramento do experimento;
3. A média de marcos por face cresce continuamente e de forma suave. Este comportamento é condizente com a contagem geral de marcos resolvidos e de hipóteses, exibidas respectivamente nas Figuras 7.20(a) e 7.20(b).

O aspecto mais importante que pode ser extraído como conclusão está na observação do funcionamento correto do planejador, no que diz respeito a garantir a cobertura completa do objeto. Além disso, o fato de que a face mais observada tem uma quantidade de marcos que se torna estável após algumas iterações indica que o planejador evita levar a câmera a observar regiões do objeto que tenham sido suficientemente exploradas, priorizando a exploração de regiões desconhecidas ou pouco conhecidas. Isto é uma consequência direta das métricas utilizadas para a avaliação da utilidade do ponto-alvo, apresentadas na Subseção 6.2.4.

Para encerrar a análise deste caso de teste, as estatísticas dos erros de estimação são apresentadas nas Figuras 7.21(a) e 7.21(b). Ambas as medidas de erro são mantidas em níveis relativamente baixos ao longo de todo o experimento,



**Figura 7.22.** Configuração do caso de teste OWL: **(a)** Objeto de interesse a ser reconstruído pelo experimento; **(b)** Cenário da montagem experimental, mostrando a câmera acoplada ao braço robótico e o objeto de interesse; **(c)** Detalhe da montagem da câmera no braço robótico; **(d)** Montagem para calibração intrínseca e *hand-eye*. Nota-se o alvo de calibração utilizado.

raramente extrapolando a marca de 0,50% do tamanho do objeto.

### 7.3.2 Experimentos reais

Para encerrar o conjunto de testes realizado sobre este trabalho, foi montado um cenário real para a recuperação da geometria de um objeto não estruturado. A experimentação real apresentada a seguir tem por objetivo prover uma base para a avaliação qualitativa (visual) dos resultados.

O caso de teste real será identificado por OWL. O objeto de interesse é uma

estátua artesanal de uma coruja com 198 mm de altura, visto na Figura 7.22(a). Este objeto foi escolhido por não apresentar regiões estruturadas e pelo fato de que a pintura apresenta características fortemente lambertianas, com poucas specularidades, conforme as proposições inicialmente lançadas na Seção 1.3.

A montagem consiste em uma máquina fotográfica digital *Canon PowerShot S2 IS* com 5 megapixels acoplado a um braço robótico *Comau C4G* (Figuras 7.22(b)–(c)). O braço robótico foi adotado por permitir a movimentação da câmera nos 6 graus de liberdade desejados (translação e rotação). As imagens da câmera foram reduzidas para 1/16 de sua resolução original, resultando em imagens de  $648 \times 486$  pixels.

A plataforma utilizada nos testes consiste em um processador Intel® Core™ i7 de 1,73 GHz com 4 GB de RAM, executando um sistema operacional GNU/Linux Ubuntu de 64 bits (com *kernel* de versão 2.6.38).

A calibração *hand-eye*, responsável pela estimação da pose relativa da câmera em relação ao sistema de coordenadas da garra do braço robótico, foi realizada com base no algoritmo de Tsai & Lenz [1988], utilizando a implementação automática desenvolvida pelo Laboratório de Visão Computacional da *Swiss Federal Institute of Technology (ETH)* [Heckscher, 2005; Wengert et al., 2006; Wengert, 2012]. Este algoritmo foi implementado em Matlab como uma extensão do tradicional *Camera Calibration Toolbox for Matlab* [Bouguet, 2008]. Essa implementação também foi utilizada para a estimação dos parâmetros intrínsecos da câmera. A montagem para a calibração pode ser vista na Figura 7.22(d).

Por questões de segurança relativas à integridade do equipamento utilizado para este caso de testes, o planejador foi limitado para evitar certas regiões do espaço, descritas a seguir:

- As posições da câmera foram limitadas de modo a respeitar uma altura mínima do solo, para evitar colisões entre a câmera e o banco sobre o qual o objeto estava repousado (ver Figura 7.22(b)) e para evitar que o próprio banco gerasse oclusão contra a observação do objeto;
- As orientações da câmera foram limitadas de modo a evitar a observação do objeto a partir de um ângulo muito inclinado em relação ao plano horizontal (foi estabelecido um ângulo máximo de  $60^\circ$ ). Durante a montagem do cenário e testes do braço robótico, foi observado que ângulos elevados poderiam levar o braço a extrapolar o espaço de trabalho, impedindo a execução da trajetória planejada.

A consequência da imposição dessas limitações nos resultados obtidos será



**Figura 7.23.** Algumas imagens da sequência de entrada para o caso de teste OWL.

discutida adiante.

### 7.3.2.1 Resultados e discussões

Algumas imagens de entrada podem ser vistas na Figura 7.23. Como se pode perceber, o planejador foi capaz de levar a câmera a observar o objeto de diversos ângulos, permitindo a reconstrução do objeto no campo visual de 360° em relação à horizontal. O planejador também foi capaz de respeitar a distância desejada ao objeto,  $d_{\text{fut}}$  (automaticamente determinada de acordo com o algoritmo descrito na Subseção 6.2.1). Com isto, o planejador foi capaz de evitar colisões com o objeto durante toda a experimentação, ao mesmo tempo mantendo uma distância de observação relativamente constante, facilitando a correspondência entre pontos salientes entre as imagens e o banco de dados.

Parte das imagens foram coletadas sem a preocupação em ocultar o fundo do ambiente de testes, como se pode perceber nas primeiras imagens apresentadas na Figura 7.23. Dado que a metodologia para a correspondência de pontos salientes foi desenvolvida de modo a manter a consistência geométrica dos pareamentos, é esperado que pontos salientes no fundo da imagem sejam descartados, já que o movimento aparente dos pontos salientes ocorre no plano de fundo (portanto incompatível com o movimento aparente dos pontos na superfície do objeto). De fato, o algoritmo de correspondência foi capaz de selecionar pontos salientes do objeto de interesse na maior parte dos casos. As falhas ocorreram apenas em pontos de vista onde o fundo da cena apresentava diversos artefatos visuais, fornecendo uma quantidade de pontos de interesse maior do que os da superfície da estatueta. Para evitar a ocorrência desses problemas, um painel de fundo foi posto entre o objeto e o fundo da cena.

Alguns pontos de vista do objeto reconstruído podem ser vistas na Figura 7.24. O modelo tridimensional foi gerado a partir do algoritmo previamente detalhado na Subseção 7.2.1.

Embora a apreciação dos resultados deste experimento real sejam puramente qualitativos, é importante notar dois fatos fundamentais:

- Pode-se perceber a similaridade da geometria entre o objeto original e a geometria estimada. Esta similaridade evidencia o correto funcionamento da metodologia de estimação de estados apresentada no 5, que por sua vez depende da alimentação de dados consistentes do módulo de correspondência de pontos salientes, proposta no 4;
- O fato de que o objeto reconstruído permite a visualização a partir de qualquer ponto de vista horizontal (reconstrução em 360°) é evidência de que o planejador proposto no 6 foi capaz de conduzir autonomamente a câmera em torno do objeto, garantindo a completude da missão neste aspecto.

Outro ponto digno de nota: As regiões de reconstrução incompleta parecem indicar uma aparente preferência do planejador para observar principalmente a parte superior da estatueta. De fato, a observação das imagens coletadas, apresentadas na Figura 7.23, apresenta poucas observações da parte inferior do objeto. No entanto, esta tendência é consequência das limitações de movimentação impostas à câmera discutidas anteriormente, que levaram o planejador a descartar por diversas vezes a adoção de poses potencialmente perigosas para o aparato experimental. Pelo mesmo motivo, não foram coletadas imagens que observariam o objeto de cima para baixo, de modo que a sua superfície superior não foi totalmente reconstruída.



**Figura 7.24.** Visualizações renderizadas da geometria reconstruída para o caso de teste OWL.

## Capítulo 8

### Conclusões e trabalhos futuros

**A**S CONSIDERAÇÕES APRESENTADAS NESTE CAPÍTULO têm por objetivo a consolidação das contribuições científicas alcançadas neste trabalho, comparando-as com as técnicas presentes no estado-da-arte para a solução de problemas similares ao abordado neste trabalho. Em seguida, serão discutidas as limitações inerentes à solução proposta e as possíveis formas de contorná-las, formando um conjunto de sugestões para trabalhos futuros.

#### 8.1 Contribuições

O presente trabalho teve por objetivo apresentar uma metodologia para reconstrução autônoma de objetos de interesse, por meio de câmeras embarcadas em robôs móveis. O caráter autônomo determina que o robô deve analisar o conhecimento parcial da geometria do objeto até o momento e planejar suas ações futuras de modo a buscar três objetivos: a observação de regiões inexploradas do objeto, promovendo a expansão do conhecimento corrente; o refinamento do conhecimento atual das regiões já exploradas; e a correção da pose estimada do próprio robô, que não pode ser determinada de maneira absoluta na falta de dispositivos globais de posicionamento, como GPS ou marcos estacionários no ambiente.

A literatura científica apresenta diversas abordagens para a realização da reconstrução de objetos e/ou mapeamento de ambientes a partir de imagens obtidas por uma única câmera móvel. No entanto, pouca ênfase é dada em tornar o processo totalmente autônomo, de modo que este possa ser executado sem intervenção humana. Uma consequência direta da busca pela autonomia no processo é a necessidade de que decisões de atuação sejam tomadas a partir de dados incompletos, ou seja, da reconstrução parcial do objeto de interesse. Este fato

impede a utilização de diversas técnicas apresentadas na literatura científica que lidam com o processamento em lote dos dados — aquelas que dependem do volume completo de dados disponível *a priori*, como as diversas abordagens de *multiple-view geometry*.

Por outro lado, o problema de **SLAM** — o mapeamento aliado à estimação da pose do observador —, embora estudado vastamente pela ciência, em geral não é abordado com vistas à automação do processo de reconstrução. Em outras palavras, o estado-da-arte assume que a câmera é controlada por um ser humano, que representa a entidade implicitamente responsável por analisar os dados parciais e tomar decisões acerca dos novos pontos de vista, de modo a garantir a completude da cobertura. É comum também assumir que a movimentação da câmera seja feita sem preocupação com a cobertura da entidade a ser mapeada, como um veículo programado para navegar aleatoriamente pelo ambiente ou uma sequência de imagens geradas por um turista. Nesses casos, a extração de informações é realizada sem qualquer análise com vistas a determinar ou guiar a atuação futura.

As pesquisas em planejamento de movimentação em tarefas exploratórias em geral se restringem à movimentação horizontal, assumindo modelos de representação planares do espaço de navegação. Mesmo as pesquisas capazes de reconstruir a estrutura tridimensional (por exemplo, as paredes de um ambiente predial) tendem a realizar a tomada de decisões a partir do princípio de que os ambientes podem ser modelados como plantas baixas ou de maneira similar — uma simplificação justificável nesses casos. No entanto, essas considerações não se aplicam à pesquisa apresentada neste trabalho, que assume que o robô possui liberdade completa de movimentação tridimensional e que a cena a ser explorada não é necessariamente estruturada. Além disso, as técnicas tradicionais para representação de mapas em duas dimensões não podem ser trivialmente estendidas para ambientes tridimensionais, dada a conseqüente explosão do custo computacional para armazenamento e processamento dos dados.

Finalmente, merece a atenção um aspecto importante do planejamento com vistas à reconstrução de um objeto de interesse: Neste caso, a determinação da atuação deve ser realizada de modo a manter o foco de atenção no objeto de interesse. Esta restrição torna este trabalho distinto da maioria das metodologias existentes para exploração ambiental, já que estas em geral tendem a manter o foco sensorial voltado para as regiões inexploradas, possivelmente confiando em técnicas de pós-processamento com vistas a corrigir eventuais distorções geométricas da representação obtida.

Diante desses desafios, este trabalho se propôs a apresentar inovações em



três frentes interdependentes:

1. O desenvolvimento de um algoritmo para a correspondência de pontos salientes entre pares de imagens focada em manter a consistência geométrica dos pareamentos retornados, de modo a reduzir sensivelmente a ocorrência de correspondências espúrias capazes de corromper o processo de reconstrução. Uma característica importante desta contribuição está em não assumir que a relação geométrica entre regiões das imagens pode ser representada por uma homografia — uma abordagem comum na literatura. Esta simplificação, que assume que a cena imageada é composta por objetos com faces (aproximadamente) planas, vai de encontro à proposição deste trabalho, que assume a presença de objetos não estruturados;
2. A construção de uma técnica de **SLAM** capaz de fornecer resultados parciais eficientes e confiáveis, evitando o custo computacional excessivo de técnicas de filtros de partículas e as imprecisões inerentes às técnicas que atribuem uma única hipótese para o posicionamento dos pontos da cena. Aliado a isso, a abordagem proposta é capaz de fornecer dados sobre os ângulos de observação de cada ponto, permitindo solucionar eficientemente o problema de orientação dos ângulos normais — uma informação crucial para o processo de planejamento. Assim como no caso do planejamento, parte do mérito desta contribuição recai na falta da necessidade de um processo externo para a localização do robô; e
3. O desenvolvimento de uma metodologia para o planejamento da exploração de objetos tridimensionais, a partir de dados que consolidam não só a geometria parcial do objeto em reconstrução, mas também uma forma eficiente para a identificar pontos de vista inéditos e propícios para avançar na tarefa exploratória. Uma dificuldade particular da abordagem proposta está em assumir que o processo de planejamento não dispõe de qualquer informação absoluta acerca da pose do robô ou do objeto a ser reconstruído, o que a diferencia das diversas técnicas conhecidas de exploração que se baseiam na disponibilidade de localização precisa.

Essas três frentes de desenvolvimento — correspondentes, respectivamente, ao conteúdo dos Capítulos 4, 5 e 6 — estão alinhadas com as proposições apresentadas no capítulo introdutório, lançadas por ocasião da formalização do problema tratado neste documento (Seção 1.3).

De maneira mais detalhada, as contribuições científicas são apresentadas a seguir:

### 8.1.1 Correspondência de pontos salientes entre imagens

A primeira contribuição científica deste trabalho trata da correspondência entre pontos salientes de pares de imagens. A inovação está em parametrizar a transformação entre segmentos das imagens como um intervalo de transformações afins. Esta é uma diferença sutil, porém essencial em relação à literatura, já que procura determinar *um intervalo* de transformações, ao invés de uma transformação fixa. Com isso, é possível realizar a correspondência de regiões não planares — uma característica fundamental para as etapas posteriores deste trabalho.

Para chegar a esses limites, realiza-se a correspondência ingênua entre os pontos salientes — isto é, levando-se em consideração tão somente a função de distância entre os vetores descritores. Em seguida, realizam-se análises estatísticas sobre o conjunto de rotações e escalas de cada ponto correspondido, de modo a identificar intervalos para cada uma dessas transformações. O passo final é a identificação dos limites translacionais, o que somente pode ser levado a termo quando se considera a correspondência de uma imagem rotacionada e escalada em relação à outra.

Os resultados obtidos demonstram que a metodologia apresentada é melhor do que a correspondência tradicional em diversos aspectos: (i) reduz sensivelmente a ocorrência de correspondências espúrias, (ii) aumenta a quantidade de correspondências corretas e (iii) é capaz de realizar a correspondência em tempo menor.

### 8.1.2 Estimação conjunta da geometria do objeto de interesse e da pose da câmera

Há duas grandes abordagens para a estimação das coordenadas de pontos da cena em **BO-SLAM**: (1) a triangulação atrasada, onde espera-se a obtenção de pontos de vista suficientes de cada ponto para avaliar uma estimação consensual; e (2) o lançamento imediato de uma ou mais hipóteses para a posição de cada ponto. Por sua vez, tradicionalmente a segunda abordagem recai em uma de duas alternativas: (i) o lançamento de uma única hipótese com grande incerteza, que será posteriormente refinada quando o ponto é reobservado; e (ii) o lançamento de um grande conjunto de hipóteses, de maneira similar aos filtros de partícula.

Este trabalho adota uma solução intermediária: A adoção de um conjunto pequeno e predeterminado de hipóteses, aproximando a distribuição real de probabilidades da posição do ponto por meio de uma **SoG**. Posteriormente, o

conjunto de hipóteses é tratada por um banco de filtros de Kalman, enquanto um processo separado determina e elimina as hipóteses de menor verossimilhança. Esta solução representa um meio-termo entre as duas abordagens tradicionais: (i) evita os problemas causados pela criação de pontos muito distantes de sua posição real, o que demanda diversas observações para que a estimação convirja para o valor esperado e que pode corromper a estimação de outros pontos da cena; e (ii) evita o custo computacional elevado de manter filtros de partículas — algo que alguns autores consideram de custo computacionalmente proibitivo.

Embora soluções semelhantes com SoG já tenham sido previamente utilizadas para a reconstrução de ambientes, este trabalho propõe algumas alterações na forma de criar o conjunto de hipóteses. O aspecto mais relevante está na inserção de um fator aleatório que elimina as tendências de agrupamento geométrico observadas na literatura científica.

### 8.1.3 Planejamento

A principal contribuição deste trabalho trata da formalização de uma metodologia inédita para a tomada de decisões no processo de reconstrução de objetos de interesse — tarefa essencial para que o processo seja verdadeiramente autônomo. Neste trabalho, o planejamento visa determinar uma pose futura para a câmera a partir da pose corrente e da nuvem de pontos que representa a reconstrução parcial do objeto.

O planejamento pode ser compreendido como a determinação de três variáveis: (i) a distância a que a câmera deve se posicionar da superfície do objeto; (ii) o ponto da superfície do objeto que deve ser observado; e (iii) o ponto de vista a ser adotado pela câmera. Em conjunto, essas variáveis definem a pose que a câmera deve adotar.

A distância desejada entre a câmera e o objeto é fixada no momento em que se dispõe dos primeiros marcos resolvidos. Ela é mantida constante ao longo de todo o processo de estimação, de modo a facilitar o processo de correspondência dos pontos salientes, já que espera-se uma variação pequena da escala desses pontos. As demais incógnitas — o ponto a ser observado e o ponto de vista — são determinados concomitantemente, de modo a buscar um equilíbrio entre dois fatores: (i) a observação de pontos próximos à fronteira corrente de exploração; (ii) o posicionamento da câmera de modo a observar regiões inexploradas do ambiente; e (iii) um limite máximo para a mudança angular do ponto de vista. As informações necessárias para a avaliação desses fatores são extraídas a partir

de um mapa tridimensional de ocupação de baixa resolução, capaz de classificar regiões do espaço em ocupadas, livres ou inexploradas; e da manutenção de vetores associados a cada ponto da nuvem, que identificam a direção da qual o ponto foi observado e facilitam a estimação da direção e sentido dos vetores normais na superfície do objeto.

Em conjunto, esses fatores permitem a observação simultânea de pontos conhecidos — o que promove ao mesmo tempo o refinamento da estimação de suas posições e da pose da câmera — e de regiões inexploradas — cumprindo assim o objetivo exploratório.

## 8.2 Limitações e trabalhos futuros

Por ser baseado em correspondências de características visuais salientes das imagens, as metodologias de correspondência de pontos entre imagens e de reconstrução propostas neste trabalho não são adequadas para cenas compostas por objetos desprovidos de textura, já que não é possível extrair pontos de interesse dessas regiões. Este é um problema comum a qualquer metodologia baseada em características visuais — mesmo a aquelas que utilizam múltiplas câmeras —, e somente pode ser contornado pela utilização de outras classes de sensores, como LRFs, câmeras de profundidade e outros sensores ativos.

Da mesma forma, objetos cujas superfícies apresentam especularidades ou efeitos de iluminação local, como transparências e translucências, também não condizem com as suposições lançadas na Seção 3.2 e possivelmente podem levar o algoritmo a gerar estimações que não representam a cena imageada. Novamente, esta é uma consequência direta da adoção de descritores visuais como o SIFT e SURF.

Toda a implementação deste trabalho foi realizada sem vistas à otimização de tempo de execução, ignorando o ganho possivelmente obtido pela divisão em múltiplas *threads* de processamento em uma plataforma com vários núcleos. A investigação desse ganho foi mitigada pela adoção do *Mathworks Matlab* para o desenvolvimento da maior parte do código-fonte — uma linguagem interpretada, limitada nas estruturas de dados disponíveis e que oferece apenas recursos primitivos de paralelização. Entretanto, diversas etapas deste trabalho são facilmente paralelizáveis, e algumas representam focos potenciais para a implementação massivamente paralela em GPUs. Nesse escopo, destacam-se as seguintes fases:

- Todas as etapas do processo de correspondência de características visuais

em imagens (Capítulo 4), que requerem a aplicação repetitiva de sequências de cálculo relativamente simples sobre milhares de elementos;

- No módulo de manutenção de marcos, a avaliação de hipóteses (Subseção 5.2.5) e marcos (Subseção 5.2.6) a serem descartadas, assim como a detecção de marcos espúrios (Subseção 5.2.7);
- Na etapa de estimação de estados, a execução do banco de filtros de Kalman, já que cada filtro opera sobre massas de dados independentes entre si (Subseção 5.3.2);
- Diversas fases do processo de planejamento, incluindo a atualização das células do volume de ocupação (Subseção 6.1.3), a estimação dos vetores normais (Subseção 6.2.3) e a avaliação das funções para a determinação do ponto-alvo (Subseção 6.2.4) e da orientação da câmera (Subseção 6.2.5).

Para o processo de estimação de estados deste trabalho, foi adotado o UKF original, sem modificações ou otimizações, implementado em Matlab e sem qualquer mecanismo de utilização de múltiplos *cores* do processador. Essas decisões de implementação foram adequadas para facilitar a depuração e a análise de resultados intermediários, porém refletiram no tempo necessário para a execução dos cálculos: Nas primeiras iterações, tanto a predição quanto a correção consomem poucos milissegundos cada, mas esse tempo cresce para algo na ordem de 20 s quando a nuvem de pontos atinge cerca de 1 000 marcos resolvidos.

Para reduzir o custo computacional, sugere-se como primeiro passo a otimização da implementação corrente e a sua adaptação para *multi-core*, explorando em particular as técnicas recentes de *General-Purpose computation on Graphics Processing Units* (GPGPU). O passo seguinte é a exploração de variantes mais eficientes de UKF, como o *Square-Root Unscented Kalman Filter* (SR-UKF) [van der Merwe & Wan, 2001]. A adoção dessas variantes requer um estudo cuidadoso diante das particularidades deste trabalho, principalmente no que diz respeito ao aspecto dinâmico da dimensão do estado do sistema (isto é, a quantidade de variáveis de estado muda constantemente, em razão da criação e eliminação contínua de marcos e hipóteses).

Como consequência direta das restrições de custo computacional, a geração de uma nuvem densa de pontos — na ordem de dezenas ou centenas de milhares de pontos — não é o objetivo do trabalho apresentado. Além disso, a correspondência de características visuais tipicamente fornece conjuntos de centenas ou poucos milhares de pares — uma ordem de grandeza baixa para a geração de nuvens densas. Entretanto, a adoção de técnicas mistas de sensoriamento pode

permitir a geração de nuvens densas: Neste caso, os algoritmos apresentados neste trabalho poderiam ser utilizados em conjunto com pares estéreo ou sensores de profundidade rigidamente acoplados com a câmera, de forma que o processo de correção da pose da câmera (parte do SLAM descrito no Capítulo 5) serviria de base para o alinhamento dos mapas de profundidade recuperados de tempos em tempos.<sup>a</sup> O sensor híbrido *Microsoft Kinect™*, lançado no mercado durante o desenvolvimento deste trabalho e recebido com entusiasmo pelas comunidades de Visão Computacional e de Robótica, é aqui apontado como um candidato potencial para o desenvolvimento de futuros trabalhos nessa direção.

Uma desvantagem do processo de estimação de estados apresentado no Capítulo 5 está na suposição intrínseca de que o processamento será realizado por uma única unidade de processamento (no próprio robô ou em um servidor). Esta suposição é evidenciada pela adoção do UKF — um algoritmo que, em sua formulação original, apresenta dificuldades para implementações descentralizadas. Atualmente, a comunidade científica tem dedicado atenção à cooperação robótica na execução de diversas tarefas, e o problema abordado neste trabalho é um bom candidato para a aplicação de times de robôs. Outras variações de filtros de Kalman — em particular o EKF — já foram adaptados para o processamento descentralizado, porém soluções semelhantes ainda não foram desenvolvidas para o UKF.

Uma possível forma de contornar essas limitações, estendendo este trabalho para cenários de robótica cooperativa, está na utilização de técnicas de fusão de dados. Nesta abordagem, cada robô seria responsável pela reconstrução parcial das regiões que observa. Eventualmente, os robôs se comunicariam entre si (ou com uma unidade centralizadora) e técnicas de fusão de dados seriam aplicadas para se obter um consenso global sobre a geometria do objeto de interesse.

Um outro foco para a extensão deste trabalho, ainda considerando o uso de times de robôs, está na coordenação das atividades, em especial na utilização de robôs com características distintas. Neste caso, a metodologia de planejamento deve ser estendida para abarcar métricas de distribuição geométrica dos robôs, promovendo a cobertura da cena em um tempo menor e evitando colisões entre eles.

A metodologia de correspondência de pontos entre imagens apresentada no Capítulo 4 toma por base que a consistência geométrica pode ser representada

---

<sup>a</sup>Nota-se que as técnicas de visão estéreo não poderiam ser trivialmente adotadas no cerne deste trabalho, já que, diante das incertezas de deslocamento da câmera ao longo do tempo, o *baseline* entre imagens consecutivas não é conhecido com precisão.

por um único intervalo de transformações afins. Esta restrição não permite a correspondência de pontos entre regiões cujas transformações são distintas, como quando se capturam sequências de imagens de objetos em profundidades diferentes. Uma metodologia de correspondência mais flexível, que não recai nesta limitação, foi posteriormente apresentada por [da Camara Neto & Campos \[2010\]](#). Da forma apresentada no artigo, esta variação não pode ser utilizada sem supervisão, pois requer a predeterminação da quantidade de transformações distintas observadas entre as imagens. No entanto, os autores acreditam que é possível adaptar o algoritmo de modo que as imagens sejam automaticamente segmentadas em regiões com base na consistência geométrica, permitindo assim o seu uso em aplicações não supervisionadas.

Uma consequência possível desta melhoria está em imprimir robustez a cenas dinâmicas. Uma vez que as imagens podem ser automaticamente segmentadas de maneira a identificar regiões com transformações distintas, é possível analisá-las para determinar se essas são consistentes com a reconstrução tridimensional parcialmente realizada. Se essa consistência for determinada, as informações seriam incorporadas à estimação; caso contrário, a região em questão pode ser ignorada, ou então tratada em um processo de reconstrução à parte.





# Apêndice A

## Filtros de Kalman discretos

Os *Filtros de Kalman discretos* são uma classe de filtros bayesianos para a estimação recursiva dos estados e respectivas incertezas de um processo dinâmico, ruidoso, temporalmente discreto e parcialmente observável, cujos ruídos de processo e de observação são brancos, gaussianos, aditivos e de média zero.

Em sua forma geral, os processos tratados pelos filtros de Kalman são modelados por uma função de transição que depende unicamente do estado imediatamente anterior e das variáveis de controle (entradas) aplicadas no passo corrente:

$$\mathbf{x}_t = \mathcal{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \varepsilon_P, \quad (\text{A.1})$$

onde  $\mathbf{x}_t$  e  $\mathbf{u}_t$  são respectivamente os vetores de estado e de controle em um instante  $t$ . A observação do processo é modelada por uma nova função:

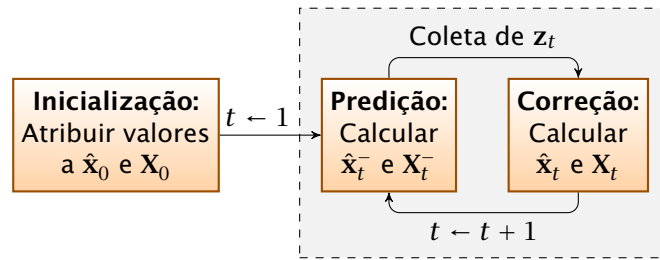
$$\mathbf{z}_t = h(\mathbf{x}_t) + \varepsilon_O, \quad (\text{A.2})$$

onde  $\mathbf{z}_t$  é a observação obtida no instante  $t$ . As variáveis aleatórias  $\varepsilon_P$  e  $\varepsilon_O$  representam respectivamente o ruído (aditivo) de processo e de observação e, conforme estabelecido anteriormente, obedecem a distribuições gaussianas de média zero:

$$\varepsilon_P \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \quad \text{e} \quad (\text{A.3})$$

$$\varepsilon_O \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t), \quad (\text{A.4})$$

onde  $\mathbf{R}_t$  é a *matriz de covariância do ruído de processo* e  $\mathbf{Q}_t$  é a *matriz de covariância*



**Figura A.1.** Ciclo de estimações dos filtros discretos de Kalman. A recursão restringe-se ao retângulo tracejado. A inicialização ocorre somente para inicializar o processo recursivo.

do ruído de observação.

A essência probabilística dos filtros de Kalman está em manter continuamente uma estimação da PDF das variáveis de estado. Nesse contexto, as Eqs. (A.1) e (A.2) são as bases para a avaliação de  $\mathcal{P}(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{z}_t)$ . Essa avaliação é composta por duas etapas:

- *Predição:* Incorpora a incerteza associada à dinâmica do processo no intervalo temporal entre  $t - 1$  e  $t$ , calculando a PDF *a priori*:  $\mathcal{P}(\mathbf{x}_t^- | \mathbf{x}_{t-1}, \mathbf{u}_t)$ , onde  $\mathbf{x}_t^-$  é o estado do sistema após o intervalo de tempo considerado;
- *Correção:* Incorpora a informação advinda da observação em  $t$ ,  $\mathbf{z}_t$ , calculando a PDF *a posteriori*:  $\mathcal{P}(\mathbf{x}_t | \mathbf{x}_t^-, \mathbf{z}_t)$ .

Por definição, nos filtros de Kalman essa PDF é sempre representada por uma distribuição normal multivariada; portanto, as PDFs são totalmente descritas pelos seus dois primeiros momentos:

$$\mathcal{P}(\mathbf{x}_t^- | \mathbf{x}_{t-1}, \mathbf{u}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_t^-, \mathbf{X}_t^-) \quad \text{e} \quad (\text{A.5})$$

$$\mathcal{P}(\mathbf{x}_t | \mathbf{x}_t^-, \mathbf{z}_t) \sim \mathcal{N}(\hat{\mathbf{x}}_t, \mathbf{X}_t), \quad (\text{A.6})$$

onde  $\hat{\mathbf{x}}_t^-$  e  $\hat{\mathbf{x}}_t$  são as médias (primeiro momento) estimadas em cada passo (predição e correção), e  $\mathbf{X}_t^-$  e  $\mathbf{X}_t$  são as correspondentes matrizes de covariância (segundo momento) das incertezas das distribuições de estados. O aspecto recursivo dos filtros discretos de Kalman é ilustrado na Figura A.1.

## A.1 O Filtro de Kalman linear

O LKF<sup>a</sup> [Swerling, 1958, 1959; Kalman, 1960; Kalman & Bucy, 1961] é uma solução estatisticamente ótima<sup>b</sup> para a estimação dos estados e incertezas de um sistema linear, observadas as demais restrições anteriormente definidas para a classe de filtros de Kalman. O caráter de otimalidade estatística se refere ao fato de que as representações das PDFs apresentadas nas Eqs. (A.5) e (A.6) não incorrem em nenhuma aproximação matemática na modelagem do processo.

No LKF, tanto o processo como a observação são modelados como transformações lineares. As Eqs. (A.1) e (A.2) são aqui transcritas de acordo com essa modelagem:

$$\mathbf{x}_t = \mathcal{g}(\mathbf{x}_{t-1}, \mathbf{u}_t) + \varepsilon_P = \mathbf{A}_t \mathbf{x}_{t-1} + \mathbf{B}_t \mathbf{u}_t + \varepsilon_P \quad \text{e} \quad (\text{A.7})$$

$$\mathbf{z}_t = \mathcal{h}(\mathbf{x}_t) + \varepsilon_O = \mathbf{C}_t \mathbf{x}_t + \varepsilon_O, \quad (\text{A.8})$$

onde  $\mathbf{A}_t$  é a *matriz de transição*,  $\mathbf{B}_t$  é a *matriz de controle* e  $\mathbf{C}_t$  é a *matriz de observação*. As equações que regem o LKF são as seguintes:

- Predição:

$$\hat{\mathbf{x}}_t^- = \mathbf{A}_t \hat{\mathbf{x}}_{t-1} + \mathbf{B}_t \mathbf{u}_t \quad \text{e} \quad (\text{A.9a})$$

$$\mathbf{X}_t^- = \mathbf{A}_t \mathbf{X}_{t-1} \mathbf{A}_t^\top + \mathbf{R}_t. \quad (\text{A.9b})$$

- Correção:

$$\mathbf{K}_t = \mathbf{X}_t^- \mathbf{C}_t^\top (\mathbf{C}_t \mathbf{X}_t^- \mathbf{C}_t^\top + \mathbf{Q}_t)^{-1}, \quad (\text{A.10a})$$

onde  $\mathbf{K}_t$  é chamado de *ganho de Kalman*, usado para ponderar a influência das observações, conforme as equações seguintes:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_t - \mathbf{C}_t \hat{\mathbf{x}}_t^-) \quad \text{e} \quad (\text{A.10b})$$

$$\mathbf{X}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{C}_t) \mathbf{X}_t^-. \quad (\text{A.10c})$$

---

<sup>a</sup>Em geral, a literatura se refere a “Filtro de Kalman-Schmidt” ou simplesmente “Filtro de Kalman”. Neste trabalho, adotamos a designação “Filtro de Kalman Linear” para estabelecer uma distinção com a classe de filtros de Kalman.

<sup>b</sup>A expressão “estatisticamente ótima” se refere ao fato de que a solução em questão é matematicamente a melhor possível, ou seja, não há algoritmos capazes de fornecer resultados mais precisos.

## A.2 O Filtro de Kalman Estendido

Algumas soluções foram desenvolvidas para estender o conceito original do LKF a sistemas não lineares. Entretanto, nenhuma das soluções é ótima, já que uma transformação não linear não garante que as PDFs representadas nas Eqs. (A.5) e (A.6) sejam gaussianas; portanto qualquer filtro de Kalman não linear é necessariamente uma aproximação.

O EKF [Jazwinski, 1970; Sorenson, 1985] é uma dessas aproximações, e atualmente é a abordagem utilizada mais frequentemente para sistemas de navegação em geral e SLAM em robótica móvel. No EKF, as PDFs são aproximadas por linearizações em torno da média, e os valores resultantes são aplicados diretamente no processo original do LKF. A linearização é obtida pela avaliação das matrizes jacobianas das funções  $g(\cdot)$  e  $h(\cdot)$ , compostas pelas derivadas parciais dessas funções em relação às variáveis de estado. Matematicamente:

$$\nabla g_t \triangleq \frac{\partial g(\mathbf{x}_{t-1}, \mathbf{u}_t)}{\partial \mathbf{x}_{t-1}} \quad \text{e} \quad (\text{A.11a})$$

$$\nabla h_t \triangleq \frac{\partial h(\mathbf{x}_t^-)}{\partial \mathbf{x}_t^-}. \quad (\text{A.11b})$$

As matrizes jacobianas são usadas diretamente nas equações que regem as duas etapas do EKF, apresentadas a seguir:

- Predição:

$$\hat{\mathbf{x}}_t^- = g(\hat{\mathbf{x}}_{t-1}, \mathbf{u}_t) \quad \text{e} \quad (\text{A.12a})$$

$$\mathbf{X}_t^- = \nabla g_t \mathbf{X}_{t-1} \nabla g_t^\top + \mathbf{Q}_t. \quad (\text{A.12b})$$

- Correção:

$$\mathbf{K}_t = \mathbf{X}_t^- \nabla h_t^\top (\nabla h_t \mathbf{X}_t^- \nabla h_t^\top + \mathbf{Q}_t)^{-1}, \quad (\text{A.13a})$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t [\mathbf{z}_t - h(\hat{\mathbf{x}}_t^-)] \quad \text{e} \quad (\text{A.13b})$$

$$\mathbf{X}_t = (\mathbf{I} - \mathbf{K}_t \nabla h_t) \mathbf{X}_t^-. \quad (\text{A.13c})$$

## A.3 O Filtro de Kalman *Unscented*

O EKF é uma abordagem interessante para a estimação de processos não lineares, mas apresenta duas características inconvenientes: (i) apenas a média é propagada através das funções não lineares (as covariâncias não o são), e a linearização pode

se revelar uma aproximação grosseira; e (ii) a avaliação das jacobianas pode ser computacionalmente cara, especialmente se as derivadas parciais não puderem ser analiticamente calculadas.

O UKF [Julier & Uhlmann, 1997; Wan & van der Merwe, 2000] é um filtro de Kalman não linear que não apresenta esses inconvenientes. No UKF, um conjunto de vetores de estados representativo de uma PDF — chamados de *pontos Sigma* — é escolhido e transformado pela função de transição; esses valores transformados são usados para ajustar os parâmetros da nova PDF. O mesmo processo é aplicado sobre a função de observação durante a etapa de correção.

Basicamente, o UKF é um filtro de partículas e assemelha-se aos processos de Monte Carlo. No entanto, em vez de selecionar as partículas por meio de sorteio aleatório sobre a distribuição de probabilidades, o UKF, por meio dos pontos Sigma, seleciona partículas estatisticamente representativas, de modo que o processo de transformação pode ser realizado com uma quantidade muito menor de partículas (e portanto com um custo computacional significativamente reduzido) do que a requerida para os algoritmos de Monte Carlo.

É importante observar que, mesmo sendo mais simples de implementar (por não requerer o cálculo das matrizes jacobianas), o UKF tem a mesma complexidade assintótica do EKF e produz resultados pelo menos tão bons quanto o EKF [Thrun et al., 2005].

### A.3.1 A Transformação *Unscented*

O processo descrito — seleção dos pontos sigma de uma PDF, avaliação da função não linear e reconstrução da nova PDF — é chamado de UT. No contexto de UKF, as PDFs são frequentemente modeladas por distribuições normais multivariadas, de modo que a UT é usada para reconstruir as médias e covariâncias de uma PDF submetida a uma função de transformação (transição ou de observação, aqui genericamente representada por  $f(\cdot)$ ) e aos ruídos dessa transformação (descritos por uma matriz de covariância, aqui genericamente representada por  $\mathbf{N}$ ):

$$\mathcal{N}(\hat{\mathbf{v}}_a, \mathbf{R}_a) \xrightarrow{\text{UT } [f(\cdot), \mathbf{N}]} \mathcal{N}(\hat{\mathbf{v}}_p, \mathbf{R}_p). \quad (\text{A.14})$$

Os pontos Sigma são assim escolhidos:

$$\chi^{[0]} = \hat{\mathbf{v}}_a, \quad (\text{A.15a})$$

$$\left[ \chi^{[1]} \mid \chi^{[2]} \mid \dots \mid \chi^{[n]} \right] = \left[ \hat{\mathbf{v}}_a \mid \hat{\mathbf{v}}_a \mid \dots \mid \hat{\mathbf{v}}_a \right] + \sqrt{(n + \lambda) \mathbf{R}_a} \quad \text{e} \quad (\text{A.15b})$$

$$\left[ \chi^{[n+1]} \mid \chi^{[n+2]} \mid \dots \mid \chi^{[2n]} \right] = \left[ \hat{\mathbf{v}}_a \mid \hat{\mathbf{v}}_a \mid \dots \mid \hat{\mathbf{v}}_a \right] - \sqrt{(n + \lambda) \mathbf{R}_a}, \quad (\text{A.15c})$$

onde  $\chi^{[i]}$  são os pontos Sigma e  $n$  é a dimensão do vetor de estados. No contexto deste trabalho, a raiz quadrada de uma matriz positiva semi-definida  $M$ ,  $\sqrt{M}$ , é uma matriz  $S$ , também positiva semi-definida, tal que  $M = S S^\top$ . O termo  $\lambda$  é definido como:

$$\lambda = \alpha^2 (n + \kappa) - n, \quad (\text{A.16})$$

onde  $\alpha$  e  $\kappa$  são parâmetros que regulam a distância entre a média  $\hat{\mathbf{v}}_a$  e os demais pontos Sigma. O termo  $\alpha$  geralmente é fixado em um valor positivo pequeno, em geral  $10^{-3} \leq \alpha \leq 1$ . O segundo parâmetro,  $\kappa$ , é um parâmetro secundário de escala que quase sempre é fixado em  $\kappa = 0$ .

A etapa seguinte da **UT** consiste em avaliar os pontos Sigma pela função de transformação (aqui representada por  $f(\cdot)$ ) e ponderados por alguns valores que representam o peso de cada ponto Sigma no resultado final:

$$\xi^{[i]} = f(\chi^{[i]}) \quad (\text{A.17})$$

$$\hat{\mathbf{v}}_p = \sum_{i=0}^{2n} w_m^{[i]} \xi^{[i]} \quad (\text{A.18})$$

$$\mathbf{R}_p = \sum_{i=0}^{2n} w_c^{[i]} (\xi^{[i]} - \hat{\mathbf{v}}_p) (\xi^{[i]} - \hat{\mathbf{v}}_p)^\top + \mathbf{N}, \quad (\text{A.19})$$

onde  $\hat{\mathbf{v}}_p$  e  $\mathbf{R}_p$  são o novo vetor de estados e sua covariância estimados pelos pontos Sigma;  $w_m^{[i]}$  e  $w_c^{[i]}$  são os pesos das ponderações, definidos da seguinte maneira:

$$w_m^{[i]} = \begin{cases} \frac{\lambda}{n + \lambda} & \text{para } i = 0 \\ \frac{1}{2(n + \lambda)} & \text{para } i > 0 \end{cases} \quad (\text{A.20})$$

$$w_c^{[i]} = \begin{cases} \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta) & \text{para } i = 0 \\ \frac{1}{2(n + \lambda)} & \text{para } i > 0 \end{cases} \quad (\text{A.21})$$

e  $\beta$  tem relação com a distribuição dos ruídos sobre o vetor de estados, onde  $\beta = 2$  é o valor adequado para modelos baseados em distribuições normais.

### A.3.2 Os cálculos do Filtro de Kalman *Unscented*

A sequência de cálculos do **UKF** pode ser consolidada nos seguintes passos:

1. *Predição*: É concluída pela aplicação direta da **UT**:

$$\mathcal{N}(\hat{\mathbf{x}}_{t-1}, \mathbf{X}_{t-1}) \xrightarrow{\text{UT}[g(\cdot, \mathbf{u}_t), \mathbf{R}_t]} \mathcal{N}(\hat{\mathbf{x}}_t^-, \mathbf{X}_t^-). \quad (\text{A.22})$$

2. *Correção*: A **UT** é usada para avaliar uma **PDF** no espaço de observações:

$$\mathcal{N}(\hat{\mathbf{x}}_t^-, \mathbf{X}_t^-) \xrightarrow{\text{UT}[h(\cdot), \mathbf{Q}_t]} \mathcal{N}(\hat{\mathbf{z}}_t, \mathbf{S}_t), \quad (\text{A.23})$$

cujos parâmetros servem de entrada para o cálculo do ganho de Kalman:

$$\mathbf{K}_t = \left[ \sum_{i=0}^{2n} w_c^{[i]} (\chi_t^{[i]} - \hat{\mathbf{x}}_t^-) (\xi_t^{[i]} - \hat{\mathbf{z}}_t)^\top \right] \mathbf{S}_t^{-1}, \quad (\text{A.24})$$

onde  $\chi_t^{[i]}$  são os pontos Sigma (calculados conforme as Eqs. (A.15a)-(A.15c)) e  $\xi_t^{[i]}$  são as correspondentes avaliações pela função de observação (conforme Eq. (A.17)). Os parâmetros da **PDF** dos estados *a posteriori* são calculados como segue:

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_t - \hat{\mathbf{z}}_t) \quad \text{e} \quad (\text{A.25a})$$

$$\mathbf{X}_t = \mathbf{X}_t^- - \mathbf{K}_t \mathbf{S}_t \mathbf{K}_t^\top. \quad (\text{A.25b})$$





## Apêndice B

# Probabilidades e o logaritmo da razão de chances

N<sup>O</sup> ESCOPO DESTA TRABALHO, todos os eventos são tratados sob o ponto de vista estocástico. Em outras palavras, qualquer evento  $x$  é tratado por meio de sua probabilidade de ocorrência,  $\mathcal{P}(x)$ , onde  $0 \leq \mathcal{P}(x) \leq 1$  e  $\mathcal{P}(x) + \mathcal{P}(\neg x) = 1$ .

Nos casos em que a probabilidade de um evento é avaliada unicamente pela fusão de sucessivas evidências coletadas por observação, a probabilidade é comumente expressa pelo *logaritmo da razão de chances* (*log-odds ratio*). Essa representação é vantajosa porque evita as instabilidades numéricas das probabilidades próximas de 0 ou 1 [Thrun et al., 2005] e permite uma implementação eficiente e elegante, pois a incorporação de evidências é realizada simplesmente por meio de operações de soma e subtração.

A *razão de chance*, ou simplesmente *chance*, de um evento qualquer  $x$  é definida como a razão entre a probabilidade de ocorrência do evento e a probabilidade de sua não ocorrência:

$$\text{odds}(x) \triangleq \frac{\mathcal{P}(x)}{\mathcal{P}(\neg x)} = \frac{\mathcal{P}(x)}{1 - \mathcal{P}(x)}, \quad (\text{B.1})$$

de onde se conclui que  $\text{odds}(x) \in \mathbb{R}^+$ . O *logaritmo da razão de chances* é simplesmente o logaritmo desta razão e corresponde à função logit aplicada à probabilidade do evento:

$$\text{logit}[\mathcal{P}(x)] \triangleq \log[\text{odds}(x)] = \log \frac{\mathcal{P}(x)}{1 - \mathcal{P}(x)}, \quad (\text{B.2})$$

de onde  $\text{logit}[\mathcal{P}(x)] \in \mathbb{R}$ .

---

```

1 procedure INCORPORA_EVIDÊNCIA_EM_PROBABILIDADE ( $P_{\text{ant}}, P_x, P_{\text{obs}}$ )
2   ▷  $P_{\text{ant}}$ : probabilidade do evento  $x$  em  $t - 1$ ,  $\mathcal{P}(x | z_{1\dots t-1})$ 
3   ▷  $P_{\text{obs}}$ : probabilidade do evento  $x$  dada uma observação,  $\mathcal{P}(x | z_t)$ 
4   ▷  $P_x$ : probabilidade incondicional do evento  $x$ ,  $\mathcal{P}(x)$ 
5   ▷  $P_{\text{post}}$ : probabilidade do evento  $x$  em  $t$ ,  $\mathcal{P}(x | z_{1\dots t})$ 
6    $P_{\text{post}} \leftarrow \text{sigm}[\text{logit}(P_{\text{obs}}) - \text{logit}(P_x) + \text{logit}(P_{\text{ant}})]$ 
7   return  $P_{\text{post}}$ 
8 end procedure

```

---

**Algoritmo B.1.** Procedimento `Incorpora_Evidência_Em_Probabilidade`: Incorpora uma evidência em uma estimação recursiva de probabilidade.

Em uma sequência temporal discreta cuja observação em um instante  $t$  é representada por  $z_t$ , a probabilidade de um evento no instante  $t - 1$  (condicionada a todas as observações anteriores,  $z_{1\dots t-1}$ ) pode ser combinada à observação  $z_t$ , obtida no instante  $t$ , para a avaliação da nova probabilidade do evento:

$$\begin{aligned} \text{logit}[\mathcal{P}(x | z_{1\dots t})] &= \text{logit}[\mathcal{P}(x | z_t)] - \text{logit}[\mathcal{P}(x)] + \\ &+ \text{logit}[\mathcal{P}(x | z_{1\dots t-1})], \end{aligned} \quad (\text{B.3})$$

onde  $\mathcal{P}(x)$  é a probabilidade *a priori* (incondicional) do evento  $x$ , utilizada também para inicializar a sequência temporal:

$$\mathcal{P}(x_0) = \mathcal{P}(x). \quad (\text{B.4})$$

A função inversa do logit é a função sigmoide,  $\text{sigm}(x)$ :

$$\text{sigm}(x) = \text{logit}^{-1}(x) \triangleq \frac{1}{1 + e^{-x}}, \quad (\text{B.5})$$

que permite reescrever a Eq. (B.3) para avaliar diretamente a probabilidade combinada do evento:

$$\begin{aligned} \mathcal{P}(x | z_{1\dots t}) &= \text{sigm}\{\text{logit}[\mathcal{P}(x | z_t)] - \text{logit}[\mathcal{P}(x)] + \\ &+ \text{logit}[\mathcal{P}(x | z_{1\dots t-1})]\}. \end{aligned} \quad (\text{B.6})$$

A Eq. (B.6) é apresentada em forma algorítmica no **Algoritmo B.1**.

Pela regra de Bayes, é possível reescrever a probabilidade do evento em função

da sequência de observações,  $\mathcal{P}(x | z_{1\dots t})$ , da seguinte maneira:

$$\mathcal{P}(x | z_{1\dots t}) = \frac{\mathcal{P}(z_t | x, z_{1\dots t-1}) \mathcal{P}(x | z_{1\dots t-1})}{\mathcal{P}(z_t | z_{1\dots t-1})} \quad (\text{B.7})$$

e, como as observações dependem unicamente do vetor de estados, tem-se  $\mathcal{P}(z_t | x, z_{1\dots t-1}) = \mathcal{P}(z_t | x)$ , de onde:

$$\mathcal{P}(x | z_{1\dots t}) = \frac{\mathcal{P}(z_t | x) \mathcal{P}(x | z_{1\dots t-1})}{\mathcal{P}(z_t | z_{1\dots t-1})}. \quad (\text{B.8})$$

A regra de Bayes aplicada sobre o modelo de observação,  $\mathcal{P}(z_t | x)$ , fornece:

$$\mathcal{P}(z_t | x) = \frac{\mathcal{P}(x | z_t) \mathcal{P}(z_t)}{\mathcal{P}(x)}, \quad (\text{B.9})$$

que, pela substituição na Eq. (B.8), fornece:

$$\mathcal{P}(x | z_{1\dots t}) = \frac{\mathcal{P}(x | z_t) \mathcal{P}(z_t) \mathcal{P}(x | z_{1\dots t-1})}{\mathcal{P}(x) \mathcal{P}(z_t | z_{1\dots t-1})}. \quad (\text{B.10})$$

De maneira análoga, a probabilidade do evento complementar de  $x$ ,  $\neg x$ , pode ser expressada por:

$$\mathcal{P}(\neg x | z_{1\dots t}) = \frac{\mathcal{P}(\neg x | z_t) \mathcal{P}(z_t) \mathcal{P}(\neg x | z_{1\dots t-1})}{\mathcal{P}(\neg x) \mathcal{P}(z_t | z_{1\dots t-1})}. \quad (\text{B.11})$$

A razão entre as Eqs. (B.10) e (B.11) permite simplificar a relação entre as probabilidades e eliminar alguns termos cuja avaliação é difícil de ser feita. Com isso, tem-se:

$$\begin{aligned} \frac{\mathcal{P}(x | z_{1\dots t})}{\mathcal{P}(\neg x | z_{1\dots t})} &= \frac{\mathcal{P}(x | z_t)}{\mathcal{P}(\neg x | z_t)} \frac{\mathcal{P}(x | z_{1\dots t-1})}{\mathcal{P}(\neg x | z_{1\dots t-1})} \frac{\mathcal{P}(\neg x)}{\mathcal{P}(x)} \\ &= \frac{\mathcal{P}(x | z_t)}{1 - \mathcal{P}(x | z_t)} \frac{\mathcal{P}(x | z_{1\dots t-1})}{1 - \mathcal{P}(x | z_{1\dots t-1})} \frac{1 - \mathcal{P}(x)}{\mathcal{P}(x)}. \end{aligned} \quad (\text{B.12})$$

A avaliação do logaritmo dos termos da Eq. (B.12) permite expressar a igualdade por meio de uma soma:

$$\log \frac{\mathcal{P}(x | z_{1\dots t})}{\mathcal{P}(\neg x | z_{1\dots t})} = \log \frac{\mathcal{P}(x | z_t)}{1 - \mathcal{P}(x | z_t)} + \log \frac{\mathcal{P}(x | z_{1\dots t-1})}{1 - \mathcal{P}(x | z_{1\dots t-1})} - \log \frac{\mathcal{P}(x)}{1 - \mathcal{P}(x)}, \quad (\text{B.13})$$

de onde, pela definição de logit da Eq. (B.2):

$$\text{logit}[\mathcal{P}(x | z_{1\dots t})] = \text{logit}[\mathcal{P}(x | z_{1\dots t-1})] + \text{logit}[\mathcal{P}(x | z_t)] - \text{logit}[\mathcal{P}(x)]. \quad (\text{B.14})$$

## Referências bibliográficas

- Alspach, D. & Sorenson, H. (1972). Nonlinear bayesian estimation using Gaussian sum approximations. *IEEE Transactions on Automatic Control*, 17(4):439-448. ISSN 0018-9286.
- Ayache, N. & Faugeras, O. D. (1988). Building, registrating, and fusing noisy visual maps. *International Journal of Robotics Research (IJRR)*, 7(6):45-65. ISSN 0278-3649.
- Bailey, T. (2003). Constrained initialisation for bearing-only SLAM. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp. 1966-1971, Taipei, Taiwan. ISSN 1050-4729.
- Bailey, T. & Durrant-Whyte, H. F. (2006). Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108-117. ISSN 1070-9932.
- Bailey, T.; Nieto, J.; Guivant, J.; Stevens, M. & Nebot, E. (2006). Consistency of the EKF-SLAM algorithm. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3562-3568, Beijing, China. IEEE. ISSN 2153-0858.
- Bay, H.; Tuytelaars, T. & Gool, L. J. V. (2006). SURF: Speeded Up Robust Features. Em Leonardis, A.; Bischof, H. & Pinz, A., editores, *Proceedings of the 9<sup>th</sup> European Conference on Computer Vision (ECCV)*, pp. 404-417, Graz, Austria. Springer-Verlag.
- Botev, Z. I. (2006). A novel nonparametric density estimator. Relatório técnico, The University of Queensland.
- Bouguet, J.-Y. (2008). Camera calibration toolbox for Matlab. Disponível em [http://www.vision.caltech.edu/bouguetj/calib\\_doc](http://www.vision.caltech.edu/bouguetj/calib_doc). Visitado em 2008-06-23.

- Bourgault, F.; Makarenko, A.; Williams, S.; Grocholsky, B. & Durrant-Whyte, H. F. (2002). Information based adaptive robotic exploration. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pp. 540-545, EPFL, Switzerland. IEEE. ISSN 2153-0858.
- Castellanos, J. A.; Martinez-Cantin, R.; Tardós, J. D. & Neira, J. (2007). Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 55(1):21-29. ISSN 0921-8890.
- Castellanos, J. A.; Neira, J. & Tardós, J. D. (2004). Limits to the consistency of EKF-based SLAM. Em *Proceedings of the 5<sup>th</sup> IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*, pp. 1244-1249, Lisbon, Portugal.
- Ceccarelli, N.; Marco, M. D.; Garulli, A.; Giannitrapani, A. & Vicino, A. (2006). Set membership localization and map building for mobile robots. Em Menini, L.; Zaccarian, L. & Abdallah, C. T., editores, *Current trends in nonlinear systems and control: In Honor of Petar Kokotovic And Turi Nicosia*, pp. 289-308. Birkäuser.
- Chatila, R. & Laumond, J. (1985). Position referencing and consistent world modeling for mobile robots. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp. 138-145, St. Louis, Missouri. ISSN 1050-4729.
- Chekhlov, D.; Pupilli, M.; Mayol-Cuevas, W. W. & Calway, A. (2007). Robust real-time visual SLAM using scale prediction and exemplar based feature description. Em *Proceedings of the 2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-7, Minneapolis, MN, USA. IEEE Computer Society. ISSN 1063-6919.
- Costa, A.; Kantor, G. & Choset, H. (2004). Bearing-only landmark initialization with unknown data association. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1764-1769, New Orleans, LA, USA. ISSN 1050-4729.
- Cox, I. (1991). Blanche: an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(3):193-204. ISSN 1042-296X.

- Crowley, J. L. (1989). World modeling and position estimation for a mobile robot using ultrasonic ranging. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pp. 674-680, Scottsdale, AZ, USA. ISSN 1050-4729.
- da Camara Neto, V. F. & Campos, M. F. M. (2010). On the improvement of image feature matching under perspective transformations. Em *Proceedings of the 23<sup>rd</sup> Conference on Graphics, Patterns and Images (SIBGRAPI)*, Gramado, RS, Brazil. ISSN 1530-1834.
- da Camara Neto, V. F.; de Mesquita, D. B.; Garcia, R. F. & Campos, M. F. M. (2010). On the design and evaluation of a precise scalable fiducial marker framework. Em *Proceedings of the 23<sup>rd</sup> Conference on Graphics, Patterns and Images (SIBGRAPI)*, Gramado, RS, Brazil. ISSN 1530-1834.
- Davison, A. & Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(7):865-880. ISSN 0162-8828.
- Davison, A. J. (2002). SLAM with a single camera. Em *SLAM/CML Workshop at IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11-15, Washington, DC, USA.
- Davison, A. J. (2003). Real-time simultaneous localisation and mapping with a single camera. Em *Proceedings of the 9<sup>th</sup> International Conference on Computer Vision (ICCV)*, pp. 1403-1410, Nice, France. ISSN 1550-5499.
- Davison, A. J.; Reid, I. D.; Molton, N. D. & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(6):1052-1067. ISSN 0162-8828.
- Delaunay, B. (1934). Sur la sphère vide. *Bulletin de l'Academie Des Sciences de l'Union des Républiques Soviétiques Socialistes, Classe des Sciences Mathématiques et Naturelles (Izvestija Akademiia Mauk SSSR, Otdelenie Matematischeskikh i Estestuennykh Nauk)*, 6:793-800.
- Durrant-Whyte, H. F. & Bailey, T. (2006). Simultaneous localization and mapping (SLAM): part I. *IEEE Robotics & Automation Magazine*, 13(2):99-110. ISSN 1070-9932.

- Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46-57. ISSN 0018-9162.
- Feder, H. J. S.; Leonard, J. J. & Smith, C. M. (1999). Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research (IJRR)*, 18(7):650-668. ISSN 0278-3649.
- Ferranti, E.; Trigoni, N. & Levene, M. (2007). Brick & mortar: an on-line multi-agent exploration algorithm. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 761-767, Roma, Italy. ISSN 1050-4729.
- Fischler, M. A. & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381-395. ISSN 0001-0782.
- Freeman, W. & Adelson, E. (1991). The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9):891-906. ISSN 0162-8828.
- Frese, U. (2006). A discussion of simultaneous localization and mapping. *Autonomous Robots*, 20(1):25-42. ISSN 0929-5593.
- Gool, L. J. V.; Moons, T. & Ungureanu, D. (1996). Affine/photometric invariants for planar intensity patterns. Em Buxton, B. F. & Cipolla, R., editores, *Proceedings of the 4<sup>th</sup> European Conference on Computer Vision (ECCV)*, volume 1, pp. 642-651, Cambridge, UK. Springer-Verlag.
- Harris, C. & Stephens, M. (1988). A combined corner and edge detector. Em *Proceedings of the 4<sup>th</sup> Alvey Vision Conference*, pp. 147-151, Manchester, UK.
- Heckscher, M. (2005). Endoscope calibration setup. Relatório técnico, Swiss Federal Institute of Technology (ETH), Zurich.
- Horn, B. K. (1986). *Robot Vision*. McGraw-Hill Higher Education. ISBN 0070303495.
- Howard, A.; Wolf, D. F. & Sukhatme, G. S. (2004). Towards 3D mapping in large urban environments. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 419-424, Sendai, Japan. IEEE. ISSN 2153-0858.



- Huang, S. & Dissanayake, G. (2006). Convergence analysis for extended Kalman filter based SLAM. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 412-417, Orlando, Florida, USA. ISSN 1050-4729.
- Huang, S. & Dissanayake, G. (2007). Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036-1049.
- Huang, S.; Kwok, N.; Dissanayake, G.; Ha, Q. & Fang, G. (2005). Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1091-1096, Barcelona, Spain. ISSN 1050-4729.
- Hygounenc, E.; Jung, I.-K.; Souères, P. & Lacroix, S. (2004). The autonomous blimp project of LAAS-CNRS: Achievements in flight control and terrain mapping. *International Journal of Robotics Research (IJRR)*, 23:473-511. ISSN 0278-3649.
- Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. Academic Press. ISBN 0123815509.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, 2ª edição. ISBN 978-0-387-95442-4.
- Julier, S. & Uhlmann, J. K. (2001). A counter example to the theory of simultaneous localization and map building. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4238-4243, Seoul, Korea. ISSN 1050-4729.
- Julier, S. J. & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. Em *Proceedings of the AeroSense: The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*, pp. 182-193.
- Jung, I.-K. & Lacroix, S. (2003). Simultaneous localization and mapping with stereovision. Em *Proceedings of the 13<sup>th</sup> International Symposium of Robotics Research*, Siena, Italy.
- Kadir, T.; Zisserman, A. & Brady, M. (2004). An affine invariant salient region detector. Em *Proceedings of the 8<sup>th</sup> European Conference on Computer Vision (ECCV)*, pp. 228-241, Prague, Czech Republic. Springer-Verlag.

- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(D):35-45. ISSN 0021-9223.
- Kalman, R. E. & Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Transactions of the ASME — Journal of Basic Engineering*, 83:95-107. ISSN 0021-9223.
- Ke, Y. & Sukthankar, R. (2004). PCA-SIFT: A more distinctive representation for local image descriptors. Em *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 506-513, Washington, DC, USA. IEEE Computer Society. ISSN 1063-6919.
- Kim, J.-H. & Sukkarieh, S. (2003). Airborne simultaneous localisation and map building. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pp. 406-411, Taipei, Taiwan. ISSN 1050-4729.
- Koenderink, J. & van Doorn, A. (1987). Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367-375. ISSN 0340-1200.
- Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79-86. ISSN 0003-4851.
- Kwok, N. M. & Dissanayake, G. (2004). An efficient multiple hypothesis filter for bearing-only SLAM. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pp. 736-741, Sendai, Japan. IEEE. ISSN 2153-0858.
- Kwok, N. M.; Dissanayake, G. & Ha, Q. P. (2005). Bearing-only SLAM using a SPRT based Gaussian sum filter. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1109-1114, Barcelona, Spain. ISSN 1050-4729.
- Lemaire, T.; Berger, C.; Jung, I.-K. & Lacroix, S. (2007). Vision-based SLAM: Stereo and monocular approaches. *International Journal of Computer Vision (IJCV)*, 74(3):343-364. ISSN 0920-5691.
- Lemaire, T.; Lacroix, S. & Sola, J. (2005). A practical 3D bearing-only SLAM algorithm. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2449-2454, Edmonton, AB, Canada. IEEE. ISSN 2153-0858.

- Leonard, J. J. & Durrant-Whyte, H. F. (1991). Simultaneous map building and localization for an autonomous mobile robot. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Osaka, Japan. IEEE. ISSN 2153-0858.
- Leonard, J. J. & Rikoski, R. J. (2000). Incorporation of delayed decision making into stochastic mapping. Em *Proceedings of the 7<sup>th</sup> International Symposium on Experimental Robotics (ISER)*, pp. 533-542, Honolulu, Hawaii, USA.
- Leonard, J. J.; Rikoski, R. J.; Newman, P. M. & Bosse, M. (2002). Mapping partially observable features from multiple uncertain vantage points. *International Journal of Robotics Research (IJRR)*, 21(10-11):943-976. ISSN 0278-3649.
- Leung, C.; Huang, S. & Dissanayake, G. (2006a). Active SLAM using model predictive control and attractor based exploration. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026-5031, Beijing, China. IEEE. ISSN 2153-0858.
- Leung, C.; Huang, S. & Dissanayake, G. (2008). Active SLAM for structured environments. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1898-1903, Pasadena, California. ISSN 1050-4729.
- Leung, C.; Huang, S.; Kwok, N. M. & Dissanayake, G. (2006b). Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898-910. ISSN 0921-8890.
- Levenberg, K. (1944). A method for the solution of certain problems in least squares. *Quarterly Applied Mathematics*, 2:164-168. ISSN 0033-569X.
- Lingemann, K.; Surmann, H.; Nuchter, A. & Hertzberg, J. (2004). Indoor and outdoor localization for fast mobile robots. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pp. 2185-2190, Sendai, Japan. IEEE. ISSN 2153-0858.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. Em *Proceedings of the 7<sup>th</sup> International Conference on Computer Vision (ICCV)*, volume 2, pp. 1150-1157, Corfu, Greece. ISSN 1550-5499.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91-110. ISSN 0920-5691.

- Makarenko, A. A.; Williams, S. B.; Bourgault, F. & Durrant-Whyte, H. F. (2002). An experiment in integrated exploration. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pp. 534-539, EPFL, Switzerland. IEEE. ISSN 2153-0858.
- Marquardt, D. (1963). An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal of on Applied Mathematics*, 11(2):431-441. ISSN 0036-1399.
- Martinell, A.; Tomatis, N. & Siegwart, R. (2005). Some results on SLAM and the closing the loop problem. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 334-339, Edmonton, AB, Canada. IEEE. ISSN 2153-0858.
- Matas, J.; Chum, O.; Urban, M. & Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. Em *Proceedings of the 13<sup>th</sup> British Machine Vision Conference (BMVC)*, pp. 384-393, Cardiff, UK.
- Maybeck, P. S. (1979). *Stochastic Models, Estimation, and Control*, volume 1. Academic Press. ISBN 0-12-480701-1.
- Mikolajczyk, K. & Schmid, C. (2002). An affine invariant interest point detector. Em Heyden, A.; Sparr, G.; Nielsen, M. & Johansen, P., editores, *Proceedings of the 7<sup>th</sup> European Conference on Computer Vision (ECCV)*, pp. 128-142, Copenhagen, Denmark. Springer-Verlag.
- Mikolajczyk, K. & Schmid, C. (2003). A performance evaluation of local descriptors. Em *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pp. 257-263, Madison, WI, USA. IEEE Computer Society. ISSN 1063-6919.
- Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615-1630. ISSN 0162-8828.
- Montgomery, D. C. (2000). *Design and Analysis of Experiments*. Wiley, 5<sup>a</sup> edição. ISBN 0471316490.
- Moravec, H. & Elfes, A. (1985). High-resolution maps from wide-angle sonar. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Louis, Missouri. ISSN 1050-4729.

- Morel, J.-M. & Yu, G. (2008). On the consistency of the SIFT Method. Relatório técnico 2008-26, Centre de Mathématiques et de Leurs Applications (CMLA), Centre National de la Recherche Scientifique (CNRS) et École Normale Supérieure de Cachan (ENS), France.
- Morel, J.-M. & Yu, G. (2009). ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal of of Imaging Sciences (SIIMS)*, 2(2):438-469. ISSN 1936-4954.
- Nüchter, A.; Lingemann, K.; Hertzberg, J. & Surmann, H. (2007). 6D SLAM — 3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699-722. ISSN 1556-4967.
- Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33(3):1065-1076. ISSN 0003-4851.
- Russell, S. J. & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2ª edição. ISBN 0-13-790395-2.
- Ryde, J. & Hu, H. (2006). Mutual localization and 3D mapping by cooperative mobile robots. Em *Proceedings of the 9<sup>th</sup> International Conference on Intelligent Autonomous Systems*, pp. 217-224, University of Tokyo, Tokyo, Japan.
- Ryde, J. & Hu, H. (2007). Mobile robot 3D perception and mapping without odometry using multi-resolution occupancy lists. Em *Proceedings of the International Conference on Mechatronics and Automation (ICMA)*, pp. 331-336, Harbin, Heilongjiang, China.
- Schaffalitzky, F. & Zisserman, A. (2003). Automated location matching in movies. *Computer Vision and Image Understanding (CVIU)*, 92(2-3):236-264. ISSN 1077-3142.
- Se, S.; Lowe, D. & Little, J. (2002). Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research (IJRR)*, 21:735-758. ISSN 0278-3649.
- Se, S.; Ng, H.-K.; Jasiobedzki, P. & Moyung, T.-J. (2004). Vision based modeling and localization for planetary exploration rovers. Em *Proceedings of the 55<sup>th</sup> International Astronautical Congress*, Vancouver, Canada.

- Shi, J. & Tomasi, C. (1994). Good features to track. Em *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 593–600, Seattle, WA, USA. IEEE Computer Society. ISSN 1063-6919.
- Sim, R. (2005a). Stabilizing information-driven exploration for bearings-only SLAM using range gating. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3396–3401, Edmonton, AB, Canada. IEEE. ISSN 2153-0858.
- Sim, R. (2005b). Stable exploration for bearings-only SLAM. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2411–2416, Barcelona, Spain. ISSN 1050-4729.
- Sim, R. & Roy, N. (2005). Global A-optimal robot exploration in SLAM. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 661–666, Barcelona, Spain. ISSN 1050-4729.
- Sinha, S. N.; Frahm, J.-M.; Pollefeys, M. & Genc, Y. (2006). GPU-based video feature tracking and matching. Em *Workshop on Edge Computing Using New Commodity Architectures (EDGE 2006)*, Chapel Hill, NC, USA.
- Sinha, S. N.; Frahm, J.-M.; Pollefeys, M. & Genc, Y. (2007). Feature tracking and matching in video using programmable graphics hardware. *Machine Vision and Applications*, pp. 1–11. Springer-Verlag.
- Smith, R.; Self, M. & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. Em *Autonomous Robot Vehicles*, pp. 167–193. Springer-Verlag, New York, Inc., New York, NY, USA.
- Smith, R. C. & Cheeseman, P. (1986). On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research (IJRR)*, 5(4):56–68. ISSN 0278-3649.
- Smith, S. M. & Brady, J. M. (1997). SUSAN — a new approach to low level image processing. *International Journal of Computer Vision (IJCV)*, 23:45–78. ISSN 0920-5691.
- Solà, J.; Monin, A. & Devy, M. (2008). Undelayed landmarks initialization for monocular SLAM. Relatório técnico, Laboratoire d'Analyse et d'Architecture des Systèmes, Centre National de la Recherche Scientifique (LAAS-CNRS), Toulouse, Occitania, France.

- Solà, J.; Monin, A.; Devy, M. & Lemaire, T. (2005). Undelayed initialization in bearing only SLAM. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2499–2504, Edmonton, AB, Canada. IEEE. ISSN 2153-0858.
- Sorenson, H. W., editor (1985). *Kalman Filtering: Theory and Application*. IEEE Press. ISBN 978-0879421915.
- Sorenson, H. W. & Alspach, D. L. (1971). Recursive bayesian estimation using Gaussian sums. *Automatica*, 7:465–479. ISSN 0005-1098.
- Stachniss, C.; Hahnel, D. & Burgard, W. (2004). Exploration with active loop-closing for FastSLAM. Em *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pp. 1505–1510, Sendai, Japan. IEEE. ISSN 2153-0858.
- Swerling, P. (1958). A proposed stagewise differential correction procedure for satellite tracking and predictions. Relatório técnico P-1292, RAND Corporation, Santa Monica, CA.
- Swerling, P. (1959). A proposed stagewise differential correction procedure for satellite tracking and predictions. *Journal of Astronautic Science*, 6:46–59.
- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*. MIT Press. ISBN 0-262-20162-3.
- Thrun, S.; Hähnel, D.; Ferguson, D.; Montemerlo, M.; Triebel, R.; Burgard, W.; Baker, C.; Omohundro, Z.; Thayer, S. & Whittaker, W. (2003). A system for volumetric robotic mapping of abandoned mines. Em *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4270–4275, Taipei, Taiwan. ISSN 1050-4729.
- Thrun, S.; Koller, D.; Ghahmarani, Z. & Durrant-Whyte, H. F. (2002). SLAM updates require constant time. Relatório técnico, School of Computer Science, Carnegie Mellon University.
- Trajkovic, M. & Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16(2):75–87. ISSN 0262-8856.

- Tsai, R. Y. & Lenz, R. K. (1988). A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. Em *Proceedings of the 4<sup>th</sup> International Symposium on Robotics Research (ISRR)*, pp. 287-297, Cambridge, MA, USA. The MIT Press.
- Tuytelaars, T. & Gool, L. V. (1999). Content-based image retrieval based on local affinely invariant regions. Em *International Conference on Visual Information Systems*, pp. 493-500, Amsterdam, Netherlands.
- Tuytelaars, T. & Gool, L. V. (2000). Wide baseline stereo matching based on local, affinely invariant regions. Em *Proceedings of the 11<sup>th</sup> British Machine Vision Conference (BMVC)*, pp. 412-425, Bristol, UK.
- van der Merwe, R. & Wan, E. A. (2001). The Square-Root Unscented Kalman Filter for state and parameter-estimation. Em *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Salt Lake City, UT, USA. ISSN 1520-6149.
- Velodyne Lidar, Inc. (2009). HDL-64E High-Definition LIDAR. Disponível em <http://www.velodyne.com/lidar/>. Visitado em 2009-07-01.
- Wan, E. A. & van der Merwe, R. (2000). The Unscented Kalman Filter for nonlinear estimation. Em *Proceedings of the IEEE Symposium on Adaptive Systems for Signal Processing, Communication and Control*, pp. 153-158, Lake Louise, Alberta, Canada.
- Wang, H. & Brady, M. (1995). Real-time corner detection algorithm for motion estimation. *Image and Vision Computing*, 13(9):695-703. ISSN 0262-8856.
- Wengert, C. (2012). Fully automatic camera and hand to eye calibration. Disponível em [http://www.vision.ee.ethz.ch/software/calibration\\_toolbox/calibration\\_toolbox.php](http://www.vision.ee.ethz.ch/software/calibration_toolbox/calibration_toolbox.php). Visitado em 2012-01-18.
- Wengert, C.; Reeff, M.; Cattin, P. C. & Székely, G. (2006). Fully automatic endoscope calibration for intraoperative use. Em *Bildverarbeitung für die Medizin*. Springer-Verlag.