

MINISTÉRIO DA DEFESA  
EXÉRCITO BRASILEIRO  
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA  
INSTITUTO MILITAR DE ENGENHARIA  
(Real Academia de Artilharia, Fortificação e Desenho, 1792)

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE DEFESA  
TESE DE DOUTORADO

ADÃO DE MELO NETO

**Mapeamento de Ambiente Interno  
Semi-estruturado com Múltiplos Veículos  
utilizando Sensor Visual Embarcado**

Rio de Janeiro  
2012

INSTITUTO MILITAR DE ENGENHARIA

ADÃO DE MELO NETO

**Mapeamento de Ambiente Interno  
Semi-estruturado com Múltiplos Veículos  
utilizando Sensor Visual Embarcado**

Tese apresentada ao Curso de Engenharia de Defesa do Instituto Militar de Engenharia, como requisito para a obtenção do título de Doutor em Engenharia de Defesa.

**Banca Examinadora**

Paulo Fernando Ferreira Rosa (IME) - Ph.D.  
Paulo César Pellanda (IME) - Docteur ENSAE  
Ricardo Choren Noya (IME) - D.Sc  
Antônio Eduardo Carrilho da Cunha (IME) D.Sc  
Vitor Ferreira Romano (UFRJ) - Dott.Ric.  
Marley Maria Bernardes Rebuzzi Vellasco (PUC/RJ) - Ph.D  
Luis Marcos Garcia Gonçalves (UFRN) - D.Sc

Rio de Janeiro  
2012

# Dedicatória

Ao Instituto Militar de Engenharia, alicerce da minha formação e aperfeiçoamento.

# Agradecimentos

Agradeço a todas as pessoas que me incentivaram, apoiaram e possibilitaram esta oportunidade de ampliar meus horizontes. Ao meu pai Jair e mãe Ivone, minha esposa Marieta e filha Patrícia Lívia. A todos os meus professores. Em especial aos meus orientadores Paulo Fernando Ferreira Rosa e Paulo César Pellanda, por suas disponibilidades e atenções e ao meu amigo Thiago Eustáquio Alves de Oliveira, por sua colaboração.

# Resumo

A exploração integrada de um ambiente é uma atividade de alto nível em que métodos de exploração, localização, mapeamento e navegação devem ser combinados tal que veículos autônomos sejam capazes de mapear um ambiente maximizando ganhos (acurácia na posição e orientação dos veículos) e minimizando custos (tempo gasto na exploração).

A exploração autônoma de um ambiente é uma atividade militar relevante por permitir o cumprimento de missões específicas, tal como a localização e a destruição de artefatos militares ou o mapeamento de um ambiente, tudo com a salvaguarda da integridade física do pessoal envolvido.

Um marco em um ambiente é algo que pode ser identificado através de sensores (como laser, sonar, infravermelho e câmera) que são utilizados na sua extração. A descrição de um marco, através de suas características, dependerá do sensor utilizado. No entanto, independentemente do sensor utilizado, uma das características de um marco são suas coordenadas 2D ou 3D.

O SLAM (*Simultaneous Localization and Mapping*) – em português, Localização e Mapeamento Simultâneos – é uma área de pesquisa que está relacionada com a capacidade de auto-localização do veículo e com a obtenção de um mapa de características de marcos do ambiente em que o veículo se encontra. As duas atividades são correlatas, pois, para obter um mapa do ambiente, é necessário que o veículo conheça sua localização, ao passo que para a localização do veículo é necessário que o mesmo tenha um mapa do ambiente. Em muitos casos, para localização do veículo, é utilizada a combinação de um GPS (*global positioning system*) com uma IMU (*inertial measurement unit*). No entanto, a qualidade das informações obtidas através desse sistema pode ser afetada pelo número de satélites à vista e por interferências eletromagnéticas. Em consequência, os veículos autônomos obtêm uma baixa precisão para a navegação em um ambiente interno.

O problema do SLAM pode ser assim definido: dados um veículo autônomo dentro de um ambiente desconhecido e apenas observações relativas

a marcos detectados no ambiente em relação ao veículo, construir um mapa de características desses marcos e calcular, simultaneamente, uma estimativa da localização do veículo baseado no mapa construído.

A exploração de um ambiente pode ser conduzida com o auxílio de uma grade de ocupação que resolve o problema da geração de um mapa consistente do ambiente a partir de medições ruidosas e incertas de um sensor (laser, no caso deste trabalho), sob a suposição de que a pose do veículo é conhecida. Como não existe veículo com odometria perfeita, uma grade é utilizada após o problema do SLAM ter sido resolvido por meio de alguma técnica, considerando que o espaço de trabalho tenha tamanho limitado.

Neste trabalho, é investigada a exploração integrada de um ambiente com um par de veículos usando uma estratégia de exploração baseada em grade de ocupação e uma estratégia SLAM para localização dos veículos. A exploração usa conceitos de custo e utilidade de células-fronteira. Além disso, o método de SLAM utilizado é baseado no algoritmo FastSLAM com marcos extraídos do ambiente a partir de sensores visuais com a técnica SIFT (*Scale-Invariant Feature Transform*) e em um mapa de características comum aos veículos. Ambas as atividades - localização dos veículos e exploração do ambiente - são coordenadas por um agente central.

Os resultados mostram que quando dois veículos comunicam-se com um agente central construindo um mapa de características dos marcos comum aos veículos, a tarefa de exploração torna-se mais eficiente do que a realizada com mapas dedicados, pois a precisão nas posição e orientação dos veículos são incrementadas.

## LISTA DE FIGURAS

1.1	Veículo observando um marco no ambiente. . . . .	19
1.2	Marcos SIFT (setas brancas). . . . .	20
1.3	Sensor visual estereoscópico [53]. . . . .	20
1.4	Exemplificação da construção de um mapa de característica de marcos	21
1.5	Exemplificação do SLAM [18]. . . . .	22
1.6	Exemplo de um conjunto de partículas do FastSLAM. . . . .	23
1.7	Abordagem FastSLAM com um mapa comum aos veículos. . . . .	24
1.8	Exemplo de um conjunto de partículas do FastSLAM com um mapa comum a dois veículos. . . . .	24
1.9	Grade de ocupação. . . . .	25
1.10	Exploração coordenada baseada em grade de ocupação. . . . .	25
1.11	Exploração Integrada: exploração (baseada em grade) e localização (baseada na abordagem FastSLAM com um mapa comum). . . . .	26
2.1	Filtro de Partículas: com poucas observações a pose do veículo é incerta (a). Após observações o filtro de partículas convergiu para uma distribuição unimodal(b) [18]. . . . .	31
2.2	Veículo obtendo a distância ( $\tau$ ) e a orientação ( $\phi$ ) em relação a um marco (a). Estes valores podem ser obtidos através de um sensor a laser (b). . . . .	32
2.3	Rede de Bayes para SLAM com veículo único. Se o caminho do veí- culo $\chi^t$ é conhecido (região sombreada), então a posição dos marcos $\theta_1$ e $\theta_2$ são condicionalmente independentes [18]. . . . .	35
2.4	Amostras (ou partículas ) projetadas de um modelo de movimento probabilístico [18]. . . . .	39
2.5	O amostrador por importância projeta amostras da distribuição pro- posta (Gaussiana - linha pontilhada) com quantidades proporcionais aos seus pesos, que são maiores na região onde a distribuição meta (linha sólida) é maior que a proposta [18]. . . . .	43
2.6	Ambiguidade na medição. Grande erros de medição conduzem a uma ambiguidade entre marcos próximos [18]. . . . .	46

2.7	Incerteza por ambiguidade do movimento devido ao movimento do veículo poder resultar em diferentes hipóteses de associação de dados para diferentes poses do veículo [18]. . . . .	47
2.8	Modelo de movimento usado . . . . .	51
3.1	(a) Imagem original e (b) Imagem com características SIFT identificadas (setas brancas). . . . .	56
3.2	Imagem da figura 3.1(a) com borramento Gaussiano aplicado (Eq.3.1). . . . .	58
3.3	Representação do procedimento de obtenção das DoG para diversas oitavas de uma imagem [21]. . . . .	59
3.4	Duas imagens suavizadas em escalas diferindo por um fator $k$ e a respectiva imagem DoG. . . . .	60
3.5	Deteção de extremos no espaço-escala [21]. . . . .	61
3.6	Histograma de orientações de um ponto-chave. . . . .	64
3.7	Mapa de gradientes para $n = 2$ regiões com $k = 4$ pixels [21]. . . . .	66
3.8	Construção do descritor para um ponto-chave de $2 \times 2$ com 48 elementos [21]. . . . .	66
3.9	Este gráfico mostra a estabilidade da detecção para localização e orientação de pontos-chave e a correspondência final para um banco de dados como uma função da distorção afim [21]. . . . .	68
3.10	A linha pontilhada exhibe o percentual de pontos-chave corretamente correspondido em função do tamanho do banco (usando escala logarítmica). A linha sólida mostra o percentual de pontos-chave que recebe uma atribuição correta em orientação, escala e orientação [21]. . . . .	69
3.11	Processo de correspondência entre duas imagens com o SIFT. . . . .	70
4.1	Na observação de um marco visual 3D (característica ( <i>keypoint</i> ) SIFT) $z_t = \{d_x, d_y, d_z\}$ e $d_t$ (descritor da característica ( <i>keypoint</i> )) devem ser obtidos. . . . .	72
4.2	Modelo de movimento usado . . . . .	76
5.1	Formação de imagem em uma câmera e no olho humano [33]. . . . .	82
5.2	Modelo de <i>pinhole</i> : um <i>pinhole</i> (a abertura de <i>pinhole</i> ) deixa somente atravessar aqueles raios de luz que interceptam um ponto particular no espaço; estes raios então formam uma imagem projetada no plano da imagem. . . . .	83
5.3	Um ponto $Q = (X, Y, Z)$ é projetado sobre o <i>plano imagem</i> através de um raio passando pelo <i>centro de projeção</i> , e o ponto resultante sobre a imagem é $q = (x, y, f)$ . . . . .	84
5.4	Distorção radial: raios mais distantes do centro de uma lente são mais inclinados se comparado com raios que passam perto do centro; portanto, os lados de um quadrado parecem curvar no <i>plano imagem</i> (distorção do tipo barril) [34]. . . . .	85
5.5	Imagem da câmera distorcida (esquerda) e corrigida (direita) [53]. . . . .	85
5.6	Em um equipamento estéreo perfeitamente alinhado, a profundidade $Z$ é obtida por semelhança de triângulos. . . . .	87
5.7	Profundidade ( $Z$ ) e disparidade ( $d$ ) são inversamente relacionados, tal que uma medição precisa da profundidade é restrita somente a objetos próximos. . . . .	88
5.8	Câmera estereoscópica “Bumblebee 2” [53] . . . . .	89

5.9	Alcance <i>versus</i> precisão da câmera estereoscópica ”Bumblebee 2” [53]	89
5.10	Alinhamento matemático (e não físico) de duas câmeras em um plano de visualização tal que linhas de <i>pixels</i> entre as câmeras estejam perfeitamente alinhadas [34]. . . . .	90
5.11	Espaço de pesquisa após a retificação estéreo. . . . .	90
5.12	Sensor Visual <i>kinect</i> [52]. . . . .	91
5.13	Imagem infravermelha do padrão de pontos projetados sobre uma cena (esquerda) e imagem resultante - mapa de profundidade - (direita) [52]. . . . .	91
5.14	Relação entre a profundidade relativa ( $Z_k$ ) e a disparidade medida ( $d$ ) [52]. . . . .	92
6.1	Esquema da exploração com dois veículos em que um agente central é responsável pela construção do mapa de características (localização) e da grade de ocupação (exploração). . . . .	95
6.2	Rede de Bayes para SLAM com dois veículos. . . . .	96
7.1	Mapa do tipo grade de ocupação (direita) de um ambiente (esquerda): as regiões claras e escuras correspondem, respectivamente, a espaços livres e ocupados. . . . .	101
7.2	Cone de um sensoriamento a laser obtido pelo veículo: $\chi_t = (x, y, \alpha)^T$ é a pose do veículo; $\beta$ é o ângulo de abertura do feixe laser ( $\beta = 180^\circ$ neste caso); $z_{max}$ é o alcance máximo do sensor laser; $z_{j,sens}$ e $\phi_{j,sens}$ são, respectivamente a distância e a orientação obtidas pelo sensor laser ( $z_{j,sens} \leq z_{max}; -\beta/2 \leq \phi_{j,sens} \leq \beta/2; 1 \leq j \leq 360$ ); $r_i$ é a distância entre $\chi_t$ e o centro de massa de uma célula $G_i$ de coordenadas $(x_i, y_i)$ . . . . .	103
7.3	Exploração baseada em fronteiras ( $\oplus$ ) de uma grade com 01 veículo ( $\circ$ ) [6]. . . . .	105
7.4	(a) grade de ocupação com células livres (branco), ocupadas (grandes pontos) e desconhecidas (pequenos pontos); (b) conjunto de células-fronteira na grade (borda de fronteira); e (c) fronteiras na grade ( $\oplus$ ) [6]. . . . .	106
7.5	Exploração baseada em fronteiras de uma grade com 02 veículos ( $\circ$ ) [8]. . . . .	107
7.6	Troca de grades locais na exploração com múltiplos veículos ( $\circ$ ) [8].	108
7.7	Exploração coordenada [01/04]. . . . .	111
7.8	Exploração coordenada [02/04]. . . . .	112
7.9	Exploração coordenada [03/04]. . . . .	113
7.10	Exploração coordenada [04/04]. . . . .	114
8.1	Esquema geral da abordagem. A área sombreada corresponde ao algoritmo FastSLAM com mapa comum (caso “b”). . . . .	116
8.2	Esquema da exploração que utiliza a abordagem FastSLAM com mapa comum. . . . .	116
8.3	Experimento 01 - Ambiente explorado com veículos em suas poses iniciais . . . . .	117
8.4	Modelo de movimento usado . . . . .	117

8.5	Experimento 01 - Trajetória percorrida pelos veículos em uma tarefa de exploração do ambiente descrito pela Fig.8.3: veículo 01 (linha contínua) e veículo 02 (linha tracejada). Número total de passos por veículo: 25. . . . .	118
8.6	Equivalência em metros do erro de $1^\circ = 0,01745$ rad na orientação do veículo: $E = \int_{0,5}^{3,5} \left( \frac{A \sin(1) + (A - A \cos(1))}{3} \right) dA = 0,0352$ m, onde $A$ é o alcance do sensor visual (0,5 a 3,5 metros), $Erro_Y = A \sin(1)$ e $Erro_X = A - A \cos(1)$ . . . . .	119
8.7	Experimento 02 - Ambiente explorado com veículos em suas poses iniciais . . . . .	121
8.8	Experimento 02 - Trajetória percorrida pelos veículos em uma tarefa de exploração do ambiente descrito pela Fig.8.7: veículo 01 (linha contínua) e veículo 02 (linha tracejada). Número total de passos por veículo: 28. . . . .	121
8.9	Experimento 03 - Ambiente explorado com veículos em suas poses iniciais . . . . .	121
8.10	Experimento 03 - Trajetória percorrida pelos veículos em uma tarefa de exploração do ambiente descrito pela Fig.8.9: veículo 01 (linha contínua) e veículo 02 (linha tracejada). Número total de passos por veículo: 27. . . . .	124
8.11	Experimento 04 - Caminho (10 metros) em um ambiente (Fig.8.12) contendo mesas, cadeiras, computadores, etc, percorrido - de forma simulada por um veículo. A coleta de dados com o sensor <i>kinect</i> ocorreu a cada 1 metro. . . . .	124
8.12	Experimento 04 - Imagem do laboratório mencionado na Figura 8.11 referente, respectivamente, a pose simulada (1,0,90) do veículo. . . . .	125
8.13	Experimento 04 - Número médio de atualizações no mapa em função de $P_0$ ( $E_0 = 1, 2$ e 1000 partículas). . . . .	126
8.14	Experimento 04 - Número médio de atualizações no mapa em função de $E_0$ ( $P_0 = 0, 9$ e 1000 partículas). . . . .	127
8.15	Experimento 04 - Correspondência das características SIFT das imagens (setas brancas) colhidas nos passos 2 e 3 (Fig.8.11) considerando $E_0 = 0, 4$ . . . . .	127
8.16	Experimentos 05 e 06 - Imagens referentes as poses (4;3,5;180) e (0;0;0) do veículo (Fig.8.18). . . . .	128
8.17	Experimento 05 - Pioneer 3DX com sensores laser (LMS-200) e <i>kinect</i> . . . . .	128
8.18	Experimento 05 - Caminho (16 passos) percorrido pelo veículo Pioneer 3DX no ambiente da figura 8.16. Cada passo tem 0,5 ou 1,0 metros. . . . .	129
8.19	Experimento 05 - Tempo de processamento em função do número de marcos no mapa a cada passo ( $P_0 = 0, 9$ e $E_0 = 0, 5$ ). . . . .	129
8.20	Experimento 05 - Atualizações em cada passo ( $P_0 = 0, 9$ e $E_0 = 0, 5$ ). . . . .	129
8.21	Experimento 05 - Número de marcos processados por passo. . . . .	130
8.22	Experimento 05 - Grade de maior resolução - <i>mapa a laser</i> - gerado pela melhor partícula ( $P_0 = 0, 9$ e $E_0 = 0, 5$ ). . . . .	130
8.23	Experimento 06 - Caminho (13 passos) percorrido - de forma simulada - pelos veículos 01 e 02 : cada passo tem 0,5 ou 1,0 metros. . . . .	131

8.24	Experimento 06 - Número médio de atualizações de marcos por passo ( $E_0 = 0,5$ e $P_0 = 0,9$ ) . . . . .	132
8.25	Experimento 06: Percepção de marcos visuais. . . . .	133
8.26	Experimento 06 - Mapa 2D dos marcos gerados pela melhor partícula ( $E_0 = 0,5$ e $P_0 = 0,9$ ) . . . . .	134
8.27	Experimento 07 - Caminho (8 passos) percorrido - de forma simulada - pelos veículos 01 e 02 : cada passo tem 0,5 metro. . . . .	135
8.28	Experimento 07 - Número médio de atualizações de marcos por passo ( $E_0 = 0,5$ e $P_0 = 0,9$ ) . . . . .	136
8.29	Experimento 07: Percepção de marcos visuais. . . . .	137
8.30	Avaliação da abordagem: trajetórias dos veículos. . . . .	137
8.31	Avaliação da abordagem: Percepção de marcos visuais. . . . .	138
10.1	Ciclo do filtro de Kalman discreto. A <i>atualização devido ao tempo</i> projeta a estimativa do estado corrente para frente no tempo. A <i>atu-</i> <i>alização devido a medição</i> ajusta a estimativa projetada pela medição real naquele instante de tempo. . . . .	145

## LISTA DE TABELAS

2.1	Amostragem dos modelo de movimento (Fig.2.8) e observação (Fig.2.2). $N(0, \sigma)$ é uma distribuição normal com média zero e desvio padrão $\sigma$ .	52
2.2	Reamostragem considerando 1 veículo ( $M = 40$ )	55
3.1	SIFT <i>versus</i> Outras abordagens.	69
4.1	Amostragem do modelo de movimento (Fig.4.2) e observação (Fig.4.1). $N(0, \sigma)$ é uma distribuição normal com média zero e desvio padrão $\sigma$ .	76
6.1	Reamostragem considerando 2 veículos ( $M = 40$ )	98
7.1	Probabilidade nas notações <i>log-odds</i> (Eq.7.5) e padrão (Eq.7.6).	102
7.2	Função $atan_2$ utilizada no algoritmo 7.2	104
8.1	Amostragem dos modelo de movimento (Fig.8.4) e observação (Fig.4.1). $N(0, \sigma)$ é uma distribuição normal com média zero e desvio padrão $\sigma = 0.5$ . Valores em metros e radianos.	117
8.2	Resultados do experimento 01 (Fig.8.5- 2000 (caso “a”) e 4000 (caso “b”) partículas)	118
8.3	Resultados do experimento 01 (Fig.8.5 - 4000 (caso “a” e “b”) partí- culas)	119
8.4	Resultados do experimento 02 (Fig.8.8- 2000 (caso “a”) e 4000 (caso “b”) partículas)	122
8.5	Resultados do experimento 02 (Fig.8.8-4000 partículas)	123
8.6	Resultados do experimento 03 (Fig.8.10- 2000 (caso “a”) e 4000 (caso “b”) partículas)	125
8.7	Resultados do experimento 03 (Fig.8.10-4000 partículas)	126
8.8	Experimento 04 - análise dos descritores 1	127
8.9	Experimento 04 - análise dos descritores 2.	128
8.10	Experimento 05: quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de $E_0$	130

8.11	Experimento 06 - Quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de $E_0 = 0,5$ . . . . .	131
8.12	Experimento 07 - Quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de $E_0 = 0,5$ . . . . .	133
8.13	Comparação das abordagens de Exploração Integrada. . . . .	139
10.1	Experimentos da tabela 8.2 . . . . .	148
10.2	Experimentos da tabela 8.3 . . . . .	149
10.3	Experimentos da tabela 8.4 . . . . .	150
10.4	Experimentos da tabela 8.5 . . . . .	151
10.5	Experimentos da tabela 8.6 . . . . .	152
10.6	Experimentos da tabela 8.7 . . . . .	153

# Lista de símbolos

$t$	Instante de tempo discreto
$M$	Número de partículas
$m$	Índice da partícula ( $m \in \{1 \dots M\}$ )
$\Theta^{[m]}$	Mapa de características de marcos detectados relativos a partícula $m$
$N^{[m]}$	Número de marcos no mapa $\Theta^{[m]}$
$h$	Modelo de Movimento
$\chi_t^{[m]}$	<i>pose</i> $(x, y, \alpha)$ estimada do veículo composto pelas coordenadas $x, y$ e orientação $\alpha$
$R$	Matriz de covariância do erro de observação
$g$	Modelo de observação
$z_t$ e $v_t$	Observação
$\hat{z}_{n,t}$ e $\hat{v}_{n,t}$	Observação predita
$Q$	Matriz de covariância do erro de movimento
$p_{n,t}^{[m]}$	Probabilidade de uma observação $z_t$ estar associada ao marco $\theta_n$
$a_t$	Associação (correspondência) entre um marco observado e um marco $\theta_{a_t} \in \Theta^{[m]}$ (O marco $\theta_{a_t}$ no mapa $\Theta^{[m]}$ foi observado)
$\psi_{n,t}^{[m]}$	Posição estimada do marco $\theta_n$
$\Sigma_{n,t}^{[m]}$	Covariância da posição estimada do marco $\theta_n$
$w_t^{[m]}$	Peso da partícula $m$
$d_n$	Descritor SIFT do marco observado $\theta_n$
$S$	Conjunto de Partículas
$\chi_{(i)}^t$	Caminho seguido pelo veículo ( $i$ ).
$a^t$	Histórico de associações de dados
$\chi_{(1:k)}^t$	Conjunto de caminhos dos $k$ veículos
$u_t$	Ação de controle sobre o veículo
$u_{(1:k)}^t$	Conjunto de ações de controle sobre os $k$ veículos
$z_{(1:k)}^t$	Conjunto de medições dos $k$ veículos
$G$	Mapa do tipo grade de ocupação
$G_i$	Célula da grade

# Lista de siglas

SLAM	<i>Simultaneous localization and mapping</i>
GPS	<i>Global positioning system</i>
IMU	<i>Inertial measurement unit</i>
SIFT	<i>Scale-invariant feature transform</i>
MCL	<i>Monte Carlo Localization</i>
DoG	<i>Função diferença de gaussiana</i>
RGB	<i>Red-Green-Blue</i>
pose	<i>atitude, posição e orientação</i>
SURF	<i>Speeded Up Robust Feature</i>
PCA-SIFT	<i>Principal Components Analysis - Scale invariant feature transform</i>
GLOH	<i>Gradient Location and Orientation Histogram</i>
EKF	<i>Extended Kalman Filter</i>

<b>1</b>	<b>Introdução</b>	<b>19</b>
1.1	Objetivo . . . . .	25
1.2	Trabalhos Relacionados . . . . .	26
1.3	Organização . . . . .	28
<b>2</b>	<b>FastSLAM</b>	<b>29</b>
2.1	Introdução . . . . .	29
2.2	Representação fatorada da distribuição de probabilidade posterior . . . . .	34
2.3	Prova da fatorização do FastSLAM . . . . .	35
2.4	FastSLAM - associação de dados conhecida . . . . .	37
2.4.1	Amostragem de uma nova pose . . . . .	38
2.4.2	Atualizando as estimativas de um marco . . . . .	39
2.4.3	Calculando o peso das partículas . . . . .	42
2.4.4	Reamostragem por importância . . . . .	44
2.4.5	Distribuição de probabilidade posterior sobre o caminho do veículo (revisitada) . . . . .	44
2.5	FastSLAM - Associação de dados desconhecida . . . . .	45
2.5.1	Incerteza na associação de dados . . . . .	45
2.5.2	Associação de dados por partícula . . . . .	46
2.5.3	Associação de dados por máxima verossimilhança por partícula . . . . .	48
2.6	Adicionando novos marcos . . . . .	48
2.7	Algoritmo FastSLAM (marcos $2D$ ) . . . . .	49
2.7.1	Explicação do Algoritmo 2.1 . . . . .	51
2.7.2	Detalhes do Algoritmo 2.1 . . . . .	52
2.8	Reamostragem Estratificada . . . . .	52

2.9	Conclusão . . . . .	53
<b>3</b>	<b>Extração de características em Imagens</b>	<b>56</b>
3.1	Introdução . . . . .	56
3.2	Detecção de extremos no espaço-escala . . . . .	57
3.3	Localização precisa dos pontos-chave . . . . .	61
3.3.1	Eliminação de respostas nas bordas . . . . .	62
3.4	Atribuição da Orientação dos Descritores . . . . .	63
3.5	Construção do Descritor Local . . . . .	65
3.5.1	Sensibilidade a trocas afins . . . . .	67
3.5.2	Correspondência para grandes bases de dados . . . . .	67
3.5.3	Encontrando pontos em comum ( <i>Matching</i> ) . . . . .	68
3.6	O SIFT em relação a outras abordagens . . . . .	69
3.7	Conclusão . . . . .	70
<b>4</b>	<b>FastSLAM Visual</b>	<b>71</b>
4.1	Introdução . . . . .	71
4.2	FastSLAM para marcos <i>3D</i> . . . . .	73
4.3	Conclusão . . . . .	80
<b>5</b>	<b>Obtenção de Coordenadas <i>3D</i></b>	<b>81</b>
5.1	Conceitos Fundamentais . . . . .	81
5.1.1	Introdução . . . . .	81
5.1.2	Modelo <i>Pinhole</i> . . . . .	82
5.1.3	Distorções da lente . . . . .	84
5.2	Imageamento estéreo . . . . .	86
5.2.1	Triangulação . . . . .	87
5.2.2	Retificação estéreo . . . . .	88
5.2.3	Correspondência estéreo . . . . .	89
5.3	Coordenadas <i>3D</i> obtidas de um sensor <i>Kinect</i> . . . . .	91
5.4	Conclusão . . . . .	94
<b>6</b>	<b>FastSLAM Visual com um Mapa Comum aos Veículos</b>	<b>95</b>
6.1	Introdução . . . . .	95
6.2	Reamostragem Estratificada . . . . .	97
6.3	Conclusão . . . . .	97
<b>7</b>	<b>Exploração com Grades de Ocupação</b>	<b>100</b>
7.1	Mapeamento com grade de ocupação . . . . .	100
7.2	Exploração com um veículo . . . . .	105
7.3	Exploração com múltiplos veículos . . . . .	106
7.4	Exploração coordenada . . . . .	108

7.4.1	Custo da navegação . . . . .	108
7.4.2	Utilidade das células-fronteira . . . . .	109
7.4.3	Seleção dos destinos . . . . .	110
7.5	Conclusão . . . . .	110
<b>8</b>	<b>Experimentos e avaliação da abordagem</b>	<b>115</b>
8.1	Experimento 01 . . . . .	115
8.2	Experimento 02 . . . . .	120
8.3	Experimento 03 . . . . .	121
8.4	Experimento 04 . . . . .	122
8.5	Experimento 05 . . . . .	124
8.6	Experimento 06 . . . . .	126
8.7	Experimento 07 . . . . .	131
8.8	Avaliação da Abordagem . . . . .	134
8.9	Conclusão . . . . .	138
<b>9</b>	<b>Conclusão</b>	<b>140</b>
9.1	Contribuições . . . . .	141
9.2	Perspectivas Futuras . . . . .	142
<b>10</b>	<b>Anexos</b>	<b>143</b>
10.1	Filtro de Kalman . . . . .	143
10.1.1	Filtro de Kalman Discreto . . . . .	143
10.1.2	Filtro de Kalman extendido . . . . .	146
10.2	Tabelas . . . . .	148

A exploração integrada de um ambiente é uma atividade de alto nível em que métodos de exploração, localização, mapeamento e navegação devem ser combinados tal que veículos autônomos sejam capazes de mapear um ambiente maximizando ganhos (acurácia na pose, i.e., posição e orientação) e minimizando custos (tempo gasto na exploração).

Um marco em um ambiente (Fig.1.1) é algo que pode ser identificado através de sensores (como laser, sonar, infravermelho e câmera) que são utilizados na sua extração. A descrição de um marco, através de suas características, dependerá do sensor utilizado. No entanto, independentemente do sensor utilizado, uma das características de um marco são as estimativas de suas coordenadas 2D ou 3D.

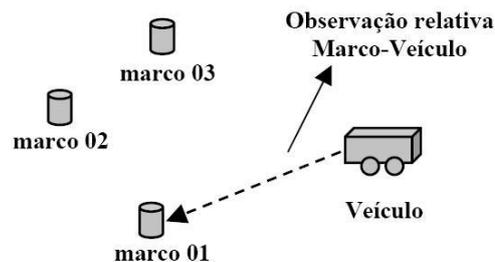


Figura 1.1: Veículo observando um marco no ambiente.

Em um ambiente interno, a obtenção de características adicionais de marcos naturais com os sensores tradicionais (laser e sonar e infravermelho) não é direta, ou seja, os dados brutos destes sensores têm que ser trabalhados para serem utilizados.

O algoritmo de visão computacional SIFT permite que se obtenha marcos visuais de uma imagem (Fig.1.2). Estes marcos são invariantes a escala e

rotação da imagem e parcialmente invariante ao ponto de vista 3D da câmera e a luminosidade. Um marco SIFT é também descrito através de um vetor de 128 posições, o descritor SIFT.

Embora a utilização de marcos visuais SIFT em um ambiente interno tenha como vantagem o fato de serem descritos pelos respectivos descritores SIFT, estes descritores são parcialmente invariantes ao ponto de vista 3D da câmera. O mesmo marco observado por duas câmeras de uma diferença de ponto de vista 3D de  $50^\circ$  tem uma correspondência entre descritores reduzida em 50%.

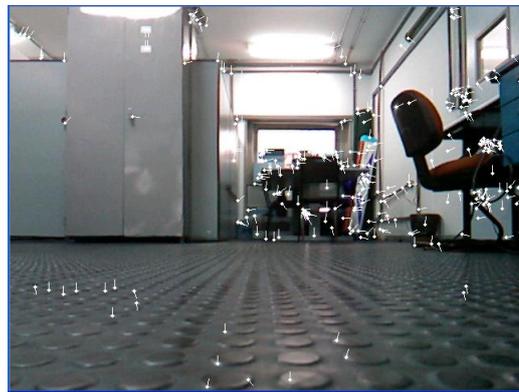


Figura 1.2: Marcos SIFT (setas brancas).

Um sensor visual estereoscópico (Fig.1.3) permite que se obtenha as distâncias de um marco visual em relação ao sensor visual.



Figura 1.3: Sensor visual estereoscópico [53].

O SLAM (*Simultaneous Localization and Mapping*) é uma área de pesquisa que está relacionada com a capacidade de auto-localização do veículo e com a obtenção de um mapa de características de marcos do ambiente (Fig.1.4) em que o veículo se encontra. As duas atividades são correlatas, pois, para obter um mapa do ambiente, é necessário que o veículo conheça sua localização, ao passo que para a localização do veículo é necessário que o mesmo tenha um mapa do ambiente. Em muitos casos, para localização do veículo, é utilizada a combinação de um GPS (*global positioning system*) com uma IMU (*inertial measurement unit*). No entanto, a qualidade das informações obtidas através desse sistema pode ser afetada pelo número de satélites à vista e por interferências eletromagnéticas. Em consequência, os

veículos autônomos obtêm uma baixa precisão para a navegação em um ambiente interno.

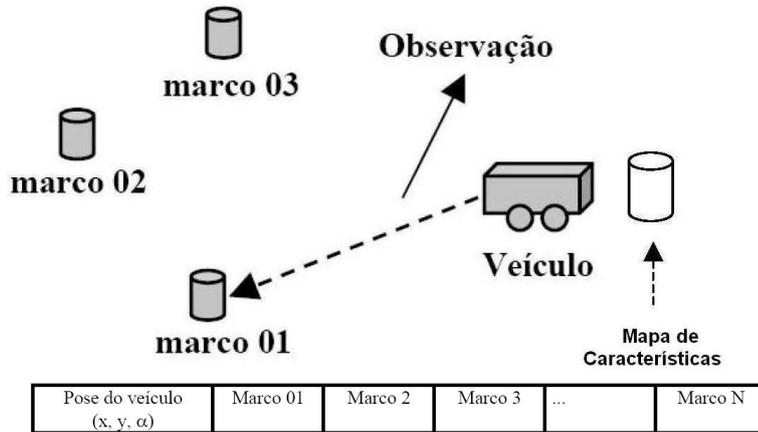


Figura 1.4: Exemplificação da construção de um mapa de característica de marcos

O problema do SLAM pode ser assim definido: dados um veículo autônomo dentro de um ambiente desconhecido e apenas observações relativas a marcos detectados no ambiente em relação ao veículo (Fig.1.1), construir um mapa de características desses marcos e calcular, simultaneamente, uma estimativa da localização do veículo baseado no mapa construído.

Na figura 1.5, temos a exemplificação do SLAM. A medida que o veículo desloca-se no ambiente, observando e mapeando novos marcos ((a),(b) e (c)), a elipse de incerteza de sua pose e da posição dos referidos marcos crescem. Quando o veículo reobserva um marco previamente mapeado, cuja posição é relativamente conhecida, esta reobservação fornece ao veículo informações sobre sua pose momentânea, cuja elipse de incerteza é reduzida (*fechamento do loop*) (d)<sup>1</sup>.

No SLAM as reobservações devem ocorrer frequentemente, a fim de que as incertezas não cresçam.

Quando um veículo observa um marco no ambiente, ele tem que verificar, baseado nas características do marco, se já existe no mapa de características, ou se é um novo marco. Este processo é conhecido como associação de dados. Se é um novo marco, ele é inserido no mapa; caso contrário, o marco associado é atualizado e uma estimativa de localização do veículo é obtida.

Normalmente a associação de dados é baseada somente na probabilidade de associação entre a pose estimada do veículo e as coordenadas estimadas dos marcos nos mapa. O marco no mapa que possua a maior probabilidade de associação, é escolhido como a associação correta.

<sup>1</sup>As coordenadas do marco reobservado são também atualizadas de acordo com a nova pose estimada do veículo.

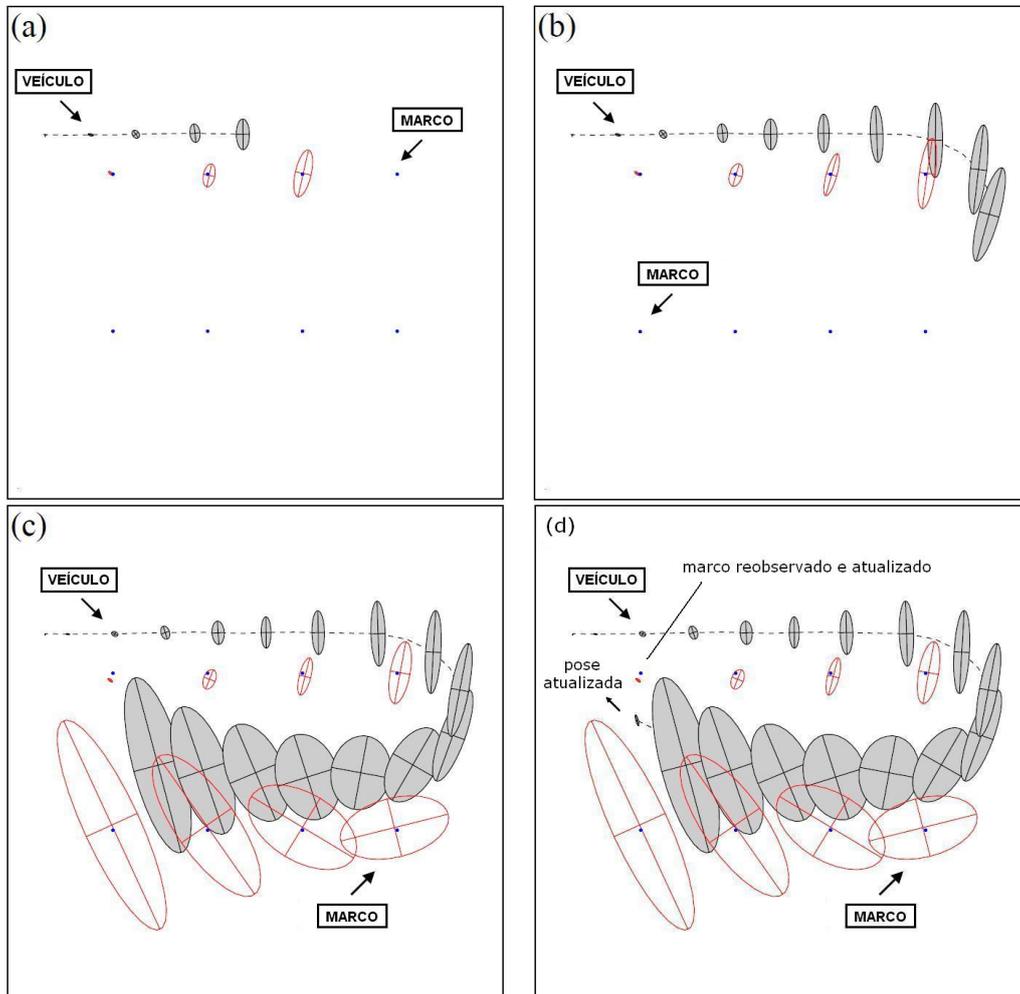


Figura 1.5: Exemplificação do SLAM [18].

O descritor SIFT permite que se melhore o processo de associação de dados dos respectivos marcos visuais SIFT, através de uma medida adicional de associação: a distância Euclidiana entre o descritor do marco observado e dos marcos anteriormente mapeados (capítulo 4).

O EKF-SLAM é uma técnica de SLAM baseada no Filtro de Kalman Extendido (EKF) (seção 10.1) em que um único mapa de características é utilizado para estimar a pose de um veículo e dos marcos (Fig.1.4). Como uma única hipótese de associação de dados é mantida, este estimador facilmente diverge caso uma associação de dados errônea seja feita.

Um filtro de partícula representa uma distribuição de probabilidade posterior usando um conjunto finito de amostras de estado, ou partículas. Regiões com alta probabilidade contêm uma alta densidade de partículas,

por outro lado, regiões com baixa probabilidade contém poucas ou nenhuma partícula.

O FastSLAM (capítulo 3) é um algoritmo SLAM que é baseado em dois estimadores, o filtro de partículas e o Filtro de Kalman Extendido (EKF). O FastSLAM permite a fatoração da distribuição de probabilidade posterior (do caminho do veículo e do mapa) em um produto de problemas de estimação de dimensão mais baixa. A pose do veículo é estimada através do conjunto de partículas do filtro de partículas e as coordenadas de cada marco mapeado (cada partícula possui seu mapa de características de marcos), através de um EKF condicionado a pose estimada do veículo. A figura 1.6 ilustra o conjunto de partículas do FastSLAM.

No FastSLAM, a associação de dados é feita por cada partícula. Quando um veículo observa um marco no ambiente, cada partícula escolhe o marco em seu mapa com a maior probabilidade de associação. Esta probabilidade de associação corresponderá ao peso da partícula<sup>2</sup>. Em cada passo, a pose estimada pela partícula com maior peso é escolhida como a pose estimada do veículo.

	Peso	Pose do veículo (x, y, $\alpha$ )	Marco 01	Marco 2	Marco 3	...	Marco N
1	0,9						
2	0,001						
3	0,001						
...							
...							
...							
1000	0,9						

Figura 1.6: Exemplo de um conjunto de partículas do FastSLAM.

Na abordagem FastSLAM com um mapa de características comum aos veículos (Fig.1.7), cada partícula contém uma estimativa da pose dos veículos e uma estimativa dos marcos mapeados pelos veículos (Fig.1.8).

O número de partículas necessárias para se obter uma boa estimação cresce exponencialmente com a dimensão do estado a ser estimado [25], ou seja, quando se usa um mapa comum a dois veículos, o número de partículas tem que ser dobrado.

A exploração de um ambiente pode ser conduzida com o auxílio de uma

---

<sup>2</sup>Por simplicidade estamos considerando que em uma observação, o veículo obtenha apenas um marco no ambiente.

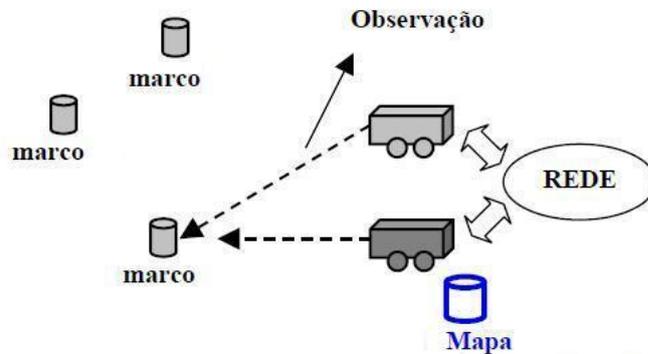


Figura 1.7: Abordagem FastSLAM com um mapa comum aos veículos.

	Peso 1	Peso 2	Peso	Pose do veículo 1 (x, y, $\alpha$ )	Pose do veículo 2 (x, y, $\alpha$ )	Marco 1	Marco 2	Marco 3	...	Marco N
1	0,9	0,001	0,03							
2	0,001	0,001	0,001							
3	0,001	0,001	0,001							
...										
...										
...										
1000	0,9	0,9	0,9							

Figura 1.8: Exemplo de um conjunto de partículas do FastSLAM com um mapa comum a dois veículos.

grade de ocupação (Fig.1.9) que resolve o problema da geração de um mapa consistente do ambiente a partir de medições ruidosas e incertas de um sensor (laser, no caso deste trabalho), sob a suposição de que a pose do veículo é conhecida. Como não existe veículo com odometria perfeita, uma grade é utilizada após o problema do SLAM ter sido resolvido por meio de alguma técnica, considerando que o espaço de trabalho tenha tamanho limitado.

Em uma estratégia de exploração coordenada (Fig.1.10), os veículos exploram o ambiente, sob a coordenação de um agente central.

Estratégias de exploração tentam mapear um terreno evitando revisitar áreas já exploradas. Elas são subótimas no contexto do SLAM, pois neste caso o veículo necessita revisitar locais de forma a se localizar e reduzir a incerteza em sua pose (*fechar o loop*).

A motivação para utilização da abordagem FastSLAM com um mapa comum aos veículos é a de que o veículo não necessita retornar a uma área

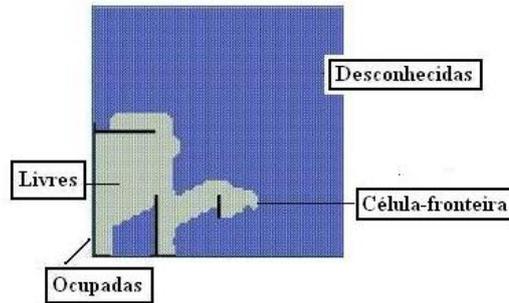


Figura 1.9: Grade de ocupação.

anteriormente explorada a fim de reduzir a incerteza de sua pose (Fig.1.5). O veículo pode reduzir a incerteza de sua pose quando reobserva um marco anteriormente mapeado pelo outro veículo.

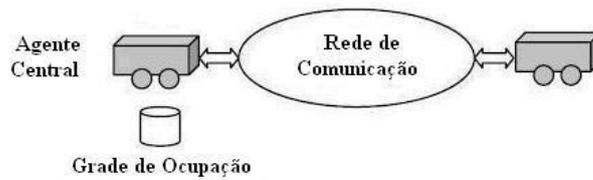


Figura 1.10: Exploração coordenada baseada em grade de ocupação.

Nas seções 1.1 e 1.2, apresentaremos, respectivamente, os nossos objetivos e os trabalhos relacionados e na seção 1.3, a organização deste trabalho.

## 1.1 Objetivo

Os objetivos do presente trabalho são (a) mostrar, através de resultados experimentais obtidos em ambientes simulados, que quando dois veículos comunicam-se com um agente central construindo um mapa de características dos marcos comum aos veículos, a tarefa de exploração torna-se mais eficiente do que a realizada com mapas dedicados, pois a precisão nas posição e orientação dos veículos são incrementadas e (b) apresentar e avaliar a implementação da abordagem em um ambiente real usando um veículo robótico autônomo e sensores (visual estereoscópico e a laser). Estas contribuições foram vislumbradas a partir do trabalho apresentado em [39] no que se refere a abordagem FastSLAM com mapa comum e a estratégia de associação de dados (Fig.1.11).

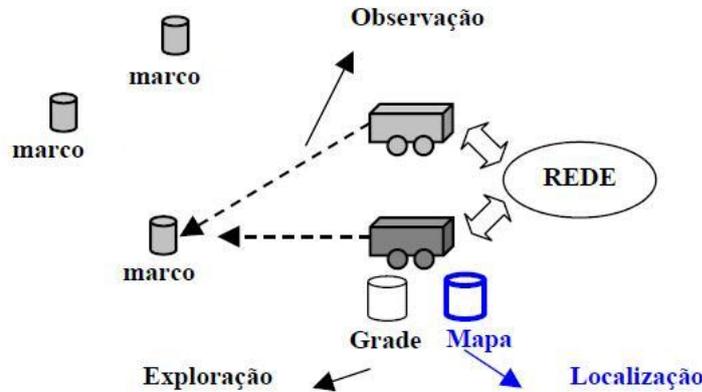


Figura 1.11: Exploração Integrada: exploração (baseada em grade) e localização (baseada na abordagem FastSLAM com um mapa comum).

## 1.2 Trabalhos Relacionados

Diversos estudos sobre o mapeamento têm sido conduzidos. Algumas abordagens estimam a pose usando somente a hodometria, o que frequentemente conduz a resultados imprecisos. Outras usam técnicas de SLAM (EKF-SLAM [11] e FastSLAM [18]) com a extração de características a partir de dados brutos de sensores laser e sonares ([38] e [31]). A abordagem EKF-SLAM tem como desvantagem o fato de ser custosa e do filtro facilmente divergir devido a hipótese única de associação de dados desta abordagem. Por outro lado, o uso dos sensores laser e sonar, com as abordagens EKF-SLAM e FastSLAM, na obtenção de características de marcos naturais detectados em um ambiente interno não é direta, ou seja, os dados brutos destes sensores têm que ser trabalhados para serem utilizados.

Em [29], o algoritmo FastSLAM [18] é utilizado para estimar simultaneamente o mapa e o caminho de um único veículo. Nesse trabalho, características SIFT, extraídas do ambiente com o auxílio de um par de câmeras estéreo, são rastreadas durante *frames* consecutivos e aquelas que são mais estáveis sob diferentes pontos de vista, são utilizadas como marcos no mapa de características das partículas. Além disso, na associação de dados, os autores propõem uma nova medida de distância, entre o descritor SIFT do marco observado e o do marco no mapa de características, ao invés da tradicional distância Euclidiana ao quadrado. Desta forma, segundo os autores, a associação de dados é melhorada, pois melhores e menores mapas são obtidos, desde que correspondências erradas podem ser evitadas. Uma desvantagem desta técnica é gerenciamento de cada característica SIFT nos diversos *frames*

para melhoria na associação de dados, que só pode ser otimizada após o rastreamento dos respectivos *frames*.

Abordagens para SLAM com múltiplos veículos podem ser agrupadas em duas soluções. Na primeira, cada veículo estima seu próprio mapa usando suas observações e, em um estágio posterior, um mapa comum é formado pela fusão dos mapas individuais dos veículos. Na segunda, a estimação de toda trajetória e do mapa é feita conjuntamente. Um único mapa é obtido usando as observações de todos os veículos.

Os trabalhos apresentados em [19] e [28] podem ser classificados no primeiro grupo. Em [19], cada veículo constrói o seu próprio mapa e, ao mesmo tempo tenta continuamente se localizar nos mapas construídos pelos outros veículos usando filtros de partículas. A abordagem pode lidar com a situação em que as poses iniciais dos veículos são desconhecidas. No entanto, a fusão dos mapas individuais é computacionalmente custosa. [28] propõe um algoritmo para múltiplos veículos – que é baseado em [18] – onde eles iniciam a partir de uma pose desconhecida. Nessa abordagem, cada veículo constrói e mantém seu próprio mapa. Quando eles estão na mesma linha de visada, os mapas são fundidos. Embora esta abordagem tenha a vantagem de lidar com a pose inicial desconhecida, o instante em que ocorre a fusão de mapas é incerta, podendo ocorrer quando o erro inerente ao algoritmo FastSLAM tenha crescido o suficiente para que a vantagem inerente a fusão dos dados seja reduzida.

As abordagens apresentadas em [16] e [39] pertencem ao segundo grupo. [16] usa um filtro de Kalman estendido (EKF) para estimar um vetor de estado formado pelas poses de todos os veículos e um conjunto de marcos 2D. Os veículos obtêm observações e constroem um mapa unificado usando as equações de atualização do EKF [11]. As posições iniciais dos veículos devem ser conhecidas com antecedência e a associação de dados é assumida como conhecida. Nesse caso, a principal desvantagem decorre do fato de que uma única hipótese sobre a pose do veículo é mantida. [39] apresenta um algoritmo baseado em [18] onde o mapa é comum aos veículos e é assumido que eles possuam pose inicial previamente conhecida. Os experimentos foram conduzidos com a observação de marcos visuais em um ambiente simulado no *MatLab*. Segundo os autores, a abordagem é adequada para um grupo de dois ou três veículos, pois a precisão do mapa estimado com este grupo, devido a estratégia de associação de dados utilizada, é similar a obtida com apenas um veículo (com o mesmo número de partículas) e, além disso, o custo computacional não aumenta consideravelmente. No entanto, os resultados obtidos pelos autores provêm apenas da observação de marcos visuais em um ambiente simulado; os parâmetros utilizados na estratégia de associação de dados do algoritmo FastSLAM não são definidos; a invariância parcial do

descriptor do marco é conduzida de forma simulada; os veículos não utilizam uma estratégia de exploração e, ao longo do texto, análises são conduzidas, considerando, na maioria das vezes, uma quantidade reduzida de partículas, insuficientes para obtenção de uma correta associação de dados no ambiente simulado e em um real. Portanto, pelas razões elencadas, uma análise definitiva dos resultados obtidos, de certa forma, ficam comprometidos.

Em relação à estratégia de exploração, destacamos a abordagem descrita em [24] que usa o conceito de custo e utilidade de células-fronteira.

Em [43],[45],[46],[47],[48],[49],[50] e [51] apresentamos resultados parciais relacionados ao tema desta tese.

As principais contribuições do presente trabalho são (a) mostrar, através de resultados experimentais obtidos em ambientes simulados, que quando dois veículos comunicam-se com um agente central construindo um mapa de características dos marcos comum aos veículos, a tarefa de exploração torna-se mais eficiente do que a realizada com mapas dedicados, pois a precisão nas posição e orientação dos veículos são incrementadas e (b) apresentar e avaliar a abordagem em um ambiente real usando um veículo robótico autônomo e sensores (visual estereoscópico e a laser). Estas contribuições foram vislumbradas a partir do trabalho apresentado em [39] no que se refere a abordagem FastSLAM com mapa comum e a estratégia de associação de dados.

Todas as outras contribuições são secundárias e servem de suporte para as contribuições principais. Dentre essas, destacamos a apresentada no capítulo 4, que descreve a adaptação do algoritmo FastSLAM para marcos 3D obtidos um sensor visual estereoscópico.

## 1.3 Organização

Os capítulos do presente trabalho tratam dos seguintes assuntos: capítulo 2- FastSLAM; capítulo 3 - extração de características em imagens com o algoritmo SIFT; capítulo 4 - FastSLAM visual; capítulo 5 - obtenção de coordenadas 3D com uso do sensores visuais; capítulo 6 - FastSLAM visual com um mapa de características comum aos veículos; capítulo 7 - estratégias de exploração do ambiente (com um ou mais veículos) utilizando mapas do tipo grade de ocupação; capítulo 8 - experimentos, resultados obtidos e avaliação da abordagem; capítulo 9 - conclusões e perspectivas futuras. O capítulo 10 contém os anexos do presente trabalho: seção 10.1, filtragem de Kalman e seção 10.2, detalhes de experimentos realizados.

## 2.1 Introdução

O problema da localização e mapeamento simultâneos (SLAM - *Simultaneous Localization and Mapping*) tem recebido grande atenção no mundo da robótica. Ele surge no contexto da navegação de veículos em um ambiente desconhecido onde nenhum mapa preciso do ambiente<sup>1</sup> é conhecido. Este problema envolve mover o veículo tentando recuperar um mapa do ambiente, enquanto simultaneamente a própria pose do veículo (localização e orientação) no mapa é estimada. Uma vez que o modelo de movimento de um veículo está sujeito a erros, o problema do mapeamento necessariamente induz a um problema de localização do veículo - e portanto o nome SLAM. Aplicações de SLAM incluem diversos tipos de ambientes: interno, externo, subaquático, subterrâneo e a exploração planetária.

Problemas de mapeamento variam em graus de dificuldade. No caso mais simples, o veículo tem acesso a um sistema combinado GPS/IMU, que fornece a ele uma informação de sua pose. O problema de adquirir um mapa com uma pose conhecida é significativamente mais fácil do que o problema do SLAM. Quando este sistema é indisponível, como no caso de um ambiente interno, subaquático ou subterrâneo, a incerteza na pose do veículo irá inevitavelmente crescer durante o mapeamento. Tais erros tem um efeito indesejável, uma vez que eles induzem a erros sistemáticos no mapa. O SLAM trata deste difícil problema de adquirir um mapa sem uma fonte externa da informação da pose do veículo.

O problema do SLAM é definido como o problema de recuperar o mapa e

---

<sup>1</sup>Mapa de características de marcos naturais detectados no ambiente.

a pose do veículo (localização e orientação) a partir de dados adquiridos pelo veículo. O veículo reúne informações sobre marcos próximos e também mede seu próprio movimento. Ambos os tipos de medidas são sujeitas a erros. Eles são compilados em uma estimativa probabilística do mapa junto com a pose momentânea do veículo (localização e orientação).

Em [18], os autores descrevem o FastSLAM, um algoritmo de SLAM que é baseado na filtragem de partículas, que iremos explicar nesta seção. O FastSLAM explora a propriedade de independência condicional que surge da estrutura do problema do SLAM, ou seja, a fatoração da distribuição de probabilidade posterior (do caminho do veículo e do mapa) em um produto de problemas de estimação de dimensão mais baixa. Em consequência, o algoritmo escala de maneira eficiente para grande mapas e é robusto a ambiguidades significativas na associação de dados.

Um filtro de partícula representa a distribuição de probabilidade posterior usando um conjunto finito de amostras de estado, ou partículas. Regiões com alta probabilidade contêm uma alta densidade de partículas, por outro lado, regiões com baixa probabilidade contêm poucas ou nenhuma partícula. Dado um número suficiente de amostras, esta representação não parametrizada pode aproximar distribuições multi-modais arbitrariamente complexas. No limite de um número finito de amostras, a distribuição verdadeira pode ser reconstruída de maneira quase exata. Dada esta representação, a equação de atualização do filtro de Bayes [29] pode ser representada usando um procedimento simples de amostragem

Filtros de partículas têm sido aplicados com sucesso em uma grade variedade de problemas de estimação do mundo real. Um dos exemplos mais comuns de filtragem de partículas em robótica é a Localização de Monte Carlo (MCL) [13]. No MCL, um conjunto de partículas é utilizado para representar a distribuição de possíveis poses de um veículo em relação a um mapa fixo. Um exemplo é mostrado na figura 2.1. Neste exemplo, não é dada nenhuma informação prévia sobre a pose do veículo. Esta completa incerteza é representada por partículas espalhadas com probabilidade uniforme pelo mapa, como mostrado na figura 2.1(a). A figura 2.1(b) exhibe o filtro de partículas após a incorporação de um certo número de controles e observações. A distribuição de probabilidade posterior convergiu, de forma aproximada, para uma distribuição unimodal.

A capacidade de controlar a crença multi-modal e de incluir modelos de medição e de movimento não-lineares faz a performance do filtro de partículas particularmente robusto. Contudo, o número de partículas necessários para controlar uma dada crença, no pior caso, cresce exponencialmente com a dimensão do estado. Desta forma, algoritmos de filtragem de partículas padrão são restritos a problemas de dimensão relativamente baixa. Filtro

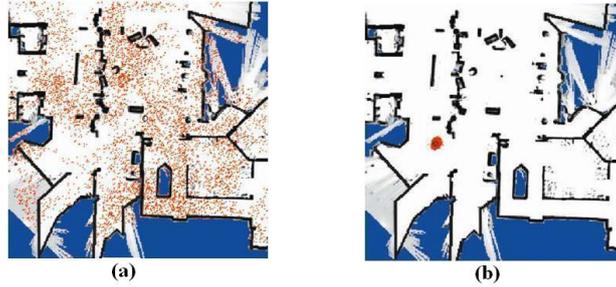


Figura 2.1: Filtro de Partículas: com poucas observações a pose do veículo é incerta (a). Após observações o filtro de partículas convergiu para uma distribuição unimodal(b) [18].

de partículas são especialmente inadequados para problemas SLAM que tem milhares de dimensões. Contudo, será mostrado como o problema do SLAM pode ser fatorado em um conjunto de problemas de estimação de marcos (independentes) condicionados a um caminho estimado do veículo. A distribuição de probabilidade do caminho do veículo é de baixa dimensão e pode ser estimado de forma eficiente usando um filtro de partículas.

Para descrever o FastSLAM mais formalmente, vamos denotar o mapa por  $\Theta$ . O mapa é constituído por um conjunto de marcos (que possuem características, tais como suas coordenadas), cada uma das quais vai ser denotados  $\theta_n$ . O número total de marcos será denotado por  $N$ . A pose do veículo é definida por  $\chi_t$ , onde  $t$  é um índice de tempo discreto. A pose dos veículos tipicamente compreende as coordenadas cartesianas bidimensionais, juntamente com a sua orientação angular  $\chi_t = (x, y, \alpha)$ . A sequência de  $\chi^t = \chi_1, \chi_2, \dots, \chi_t$  indica o caminho do veículo até o instante  $t$ .

Para adquirir um mapa, o veículo deve ter a capacidade de perceber (sensoriar) o ambiente. Sensores de medição transmitem informações sobre distância, aparência, etc. de marcos próximos. Isto é ilustrado na figura 2.2, em que um veículo obtém a distância  $\tau$  e a orientação  $\phi$  em relação a um marco. Sem perda de generalidade, assumimos - por conveniência - que o veículo observa somente um marco a cada vez (avistamentos de múltiplos marcos são processados sequencialmente). A medição no tempo  $t$ , denotado  $z_t$ , pode ser a distância e a orientação de um marco nas proximidades. Para cada medição  $z_t$ ,  $a_t$  especifica a correspondência ou associação do marco observado no mapa  $\Theta$  ( $a_t \in \{1, \dots, N\}$ ).

O núcleo do algoritmo FastSLAM é a lei probabilística que especifica o processo de acordo com o qual as medições do sensor são geradas. Este

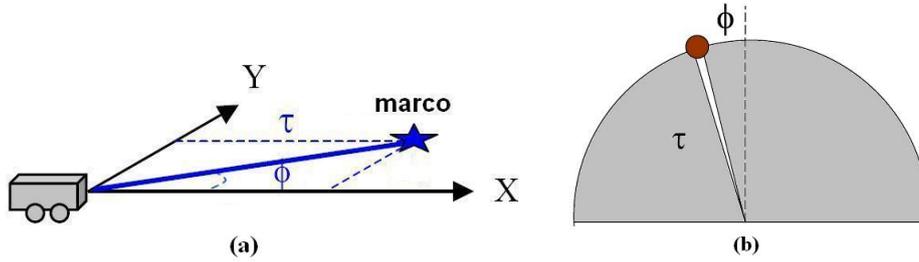


Figura 2.2: Veículo obtendo a distância ( $\tau$ ) e a orientação ( $\phi$ ) em relação a um marco (a). Estes valores podem ser obtidos através de um sensor a laser (b).

modelo será referido como modelo de medição e é da forma seguinte:

$$p(z_t | \chi_t, \theta_{a_t}, a_t) = g(\theta_{a_t}, \chi_t) + \varepsilon_t \quad (2.1)$$

O modelo de medição está condicionado à pose  $\chi_t$  do veículo, à identidade do marco  $a_t$  e ao específico marco  $\theta_{a_t}$  no mapa  $\Theta$  que está sendo observado. Ele é governado por uma função  $g$  (determinística) distorcida por ruídos. O ruído no instante  $t$  é modelado pela variável aleatória  $\varepsilon_t$ , que é considerada como normalmente distribuída com média zero e covariância  $R_t$ . A suposição de ruído Gaussiano normalmente é apenas uma aproximação, mas que tende a funcionar bem em uma variedade de sensores. A função de medição  $g$  é geralmente não-linear em seus argumentos. Um exemplo comum são as medidas de distância e orientação para marcos, como discutido acima (Fig.2.2). A distância e orientação para marcos são facilmente calculados através de simples funções trigonométricas que são não-lineares nas variáveis que representam a pose do veículo e nas coordenadas dos marcos sensorizadas pelo veículo.

Uma segunda fonte de informação para resolução de problemas SLAM são os controles do veículo. Os controles são denotados  $u_t$ , e referem-se aos comandos motores que ocorreram no intervalo de tempo  $[t-1, t]$ . A lei probabilística que regula a evolução das poses é vulgarmente referenciada como modelo de movimento cinemático, e assume a seguinte seguinte forma:

$$p(\chi_t | u_t, \chi_{t-1}) = h(u_t, \chi_{t-1}) + \delta_t \quad (2.2)$$

Como esta expressão sugere, a pose no instante  $t$  é uma função  $h$  da pose do veículo no instante de tempo anterior, distorcida por um ruído Gaussiano. Este ruído é capturado pela variável aleatória  $\delta_t$ , cuja média é zero e cuja covariância será denotada por  $Q_t$ . Como no caso do modelo de medição, a função  $h$  é geralmente não linear em seus argumentos.

O objectivo da SLAM é a recuperação do mapa a partir das medidas do sensor  $z^t = \{z_1, z_2, \dots, z_t\}$  e dos controlos  $u^t = \{u_1, u_2, \dots, u_t\}$ . A maioria dos algoritmos SLAM são exemplos de filtros de Bayes [29], e, como tal, recupera a distribuição de probabilidade sobre o mapa  $\Theta$  e a pose momentânea do veículo  $\chi_t$ :

$$p(\chi_t, \Theta | z^t, u^t, a^t) \quad (2.3)$$

A maioria dos algoritmos SLAM são instanciações do filtro de Bayes<sup>2</sup>, que calcula esta distribuição de uma (distribuição) calculada no passo anterior:

$$p(\chi^t, \Theta | z^t, u^t, a^t) = \eta p(z_t | \chi_t, \theta_{a_t}, a_t) \int p(\chi_t | \chi_{t-1}, u_t) p(\chi_{t-1}, \Theta | z^{t-1}, u^{t-1}, a^{t-1}) d\chi_{t-1} \quad (2.4)$$

$\eta$  é a constante de normalização (que é equivalente a  $p(z_t | z^{t-1}, u^{t-1}, a_t)$  nesta equação). Esta constante não depende de qualquer das variáveis sobre as quais a distribuição de probabilidade posterior é computada<sup>3</sup>.

O filtro de Bayes é o núcleo de muitas aplicações de SLAM e, à primeira vista, alguém poderia considerar que a distribuição de probabilidade anterior (Eq.2.3) captura todas as informações relevantes, e portanto seria o *padrão de ouro* para o SLAM em robótica. Contudo, existem outras distribuições mais elaboradas que podem ser estimadas no SLAM.

O algoritmo FastSLAM, em particular, estima a distribuição de probabilidade posterior sobre caminhos do veículo ( $\chi^t$ ), não apenas poses momentâneas ( $\chi_t$ ), junto com o mapa  $\Theta$ :

$$p(\chi^t, \Theta | z^t, u^t, a^t) \quad (2.5)$$

---

<sup>2</sup>Um filtro de Bayes computa a crença  $bel(\chi_t)$  de uma determinada variável de estado  $\chi$  no instante  $t$  a partir das medições e dados de controle:

1. Filtro de Bayes ( $bel(\chi_t), u_t, z_t$ )
2. Para todo  $\chi_t$  fazer

$$\begin{aligned} \overline{bel}(\chi_t) &= \int p(\chi_t | u_t, \chi_{t-1}) bel(\chi_{t-1}) d\chi \\ bel(\chi_t) &= \eta p(z_t | \chi_t) \overline{bel}(\chi_t) \end{aligned}$$

3. Fim
4. retorne  $bel(\chi_t)$

<sup>3</sup>A notação  $\eta$  será adotada para constantes de normalização genéricas, a despeito dos valores atuais das mesmas

A primeira vista, estimar a distribuição de probabilidade posterior sobre todo o caminho  $\chi^t$  pode parecer uma escolha questionável. A medida que o tamanho do caminho aumenta, o mesmo ocorre com o espaço sobre o qual a distribuição de probabilidade posterior (Eq.2.5) é definida. Esta propriedade parece estar em desacordo com a execução em tempo real do filtro. Contudo, como será visto, tipos específicos de filtros computam a distribuição de probabilidade posterior sobre caminhos tão eficientemente quando sobre poses momentâneas. Entretanto, a motivação principal é que a equação 2.5 pode ser decomposta em um produto de pequenos termos (Eq.2.7).

O filtro de Bayes, para computar a distribuição de probabilidade posterior (Eq.2.5), é definido da seguinte forma:

$$p(\chi^t, \Theta | a^t, u^t, z^t) = \eta p(z_t | \chi_t, \theta_{a_t}, a_t) p(\chi_t | \chi_{t-1}, u_t) p(\chi_{t-1}, \Theta | z^{t-1}, u^{t-1}, a^{t-1}) \quad (2.6)$$

Esta equação difere do filtro de Bayes padrão (Eq.2.4) pela ausência do sinal de integral.

## 2.2 Representação fatorada da distribuição de probabilidade posterior

A distribuição de probabilidade posterior da equação 2.6 possui uma característica importante, ou seja, ela pode ser fatorada no seguinte produto:

$$p(\chi^t, \Theta | a^t, z^t, u^t) = \underbrace{p(\chi^t | a^t, z^t, u^t)}_{\text{caminho}} \prod_{n=1}^N \underbrace{p(\theta_n | \chi^t, a^t, z^t)}_{\text{marco}} \quad (2.7)$$

Esta fatorização afirma que o cálculo da distribuição de probabilidade posterior sobre caminhos  $\chi^t$  e mapas  $\Theta$  pode ser decomposto em  $N + 1$  estimadores recursivos, um estimador sobre os caminhos dos veículos e  $N$  estimadores distintos sobre a localização dos marcos condicionados ao caminho estimado  $\chi^t$ . O produto destas probabilidades representa a distribuição de probabilidade posterior desejada na forma fatorada. Esta representação fatorada é exata, não apenas uma aproximação. É uma propriedade genérica do problema do SLAM.

Para ilustrar a corretude desta fatorização, a figura 2.3 descreve graficamente o processo de aquisição de dados, na forma de uma rede Bayesiana dinâmica. O gráfico sugere que cada medição  $z_1, \dots, z_t$  seja uma função das posições dos marcos correspondentes, juntamente com a pose do veículo no

instante em que a medição foi realizada. O conhecimento da localização exata de um marco não irá dizer nada sobre a localização dos outros marcos.

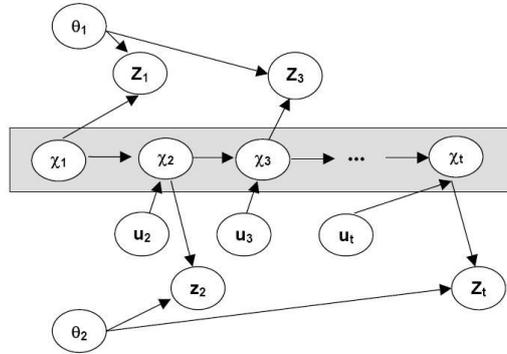


Figura 2.3: Rede de Bayes para SLAM com veículo único. Se o caminho do veículo  $\chi^t$  é conhecido (região sombreada), então a posição dos marcos  $\theta_1$  e  $\theta_2$  são condicionalmente independentes [18].

## 2.3 Prova da fatorização do FastSLAM

A independência condicional da posição dos marcos é dada pela seguinte forma em produto:

$$p(\Theta|\chi^t, a^t, z^t) = \prod_{n=1}^N p(\theta_n|\chi^t, a^t, z^t) \quad (2.8)$$

Observe que todas as probabilidades são condicionadas ao caminho do veículo  $\chi^t$ . A derivação da equação 2.8 exige a distinção de dois possíveis casos: se o marco  $\theta_n$  (no mapa  $\Theta$ ) foi ( $a_t = n$ ) ou não ( $a_t \neq n$ ) observado na medição mais recente  $z_t$  de um marco no ambiente.

Em particular, se  $a_t \neq n$ , a medição mais recente  $z_t$  não tem nenhum efeito sobre a distribuição de probabilidade posterior, na pose do veículo  $\chi_t$  ou na correspondência  $a_t$ . Desta forma, temos:

$$p(\theta_n|\chi^t, a^t, z^t) = p(\theta_n|\chi^{t-1}, a^{t-1}, z^{t-1}) \quad (2.9)$$

Se  $a_t = n$ , isto é, se o marco  $\theta_n$  (no mapa  $\Theta$ ) foi observado na medição mais recente  $z_t$  de um marco no ambiente, a situação convida à aplicação da regra de Bayes<sup>4</sup>, seguido de algumas simplificações:

<sup>4</sup> Teorema de Bayes:  $P(A|B)P(B) = P(A \cap B) = P(B \cap A) = P(B|A)P(A)$

$$\begin{aligned}
p(\theta_{a_t}|\chi^t, a^t, z^t) &= \frac{p(z_t|\theta_{a_t}, \chi^t, a^t, z^{t-1})p(\theta_{a_t}|\chi^t, a^t, z^{t-1})}{p(z_t|\chi^t, a^t, z^{t-1})} \\
&= \frac{p(z_t|\chi_t, \theta_{a_t}, a_t)p(\theta_{a_t}|\chi^{t-1}, a^{t-1}, z^{t-1})}{p(z_t|\chi^t, a^t, z^{t-1})}
\end{aligned} \tag{2.10}$$

Isto fornece a seguinte expressão para a probabilidade  $p(\theta_{a_t}|\chi^{t-1}, a^{t-1}, z^{t-1})$ :

$$p(\theta_{a_t}|\chi^{t-1}, a^{t-1}, z^{t-1}) = \frac{p(\theta_{a_t}|\chi^t, a^t, z^t)p(z_t|\chi^t, a^t, z^{t-1})}{p(z_t|\chi_t, \theta_{a_t}, a_t)} \tag{2.11}$$

A prova da correção da equação 2.8 é feita por indução matemática. Vamos supor que a distribuição de probabilidade posterior no instante  $t - 1$  já esteja fatorada:

$$p(\Theta|\chi^{t-1}, a^{t-1}, z^{t-1}) = \prod_{n=1}^N p(\theta_n|\chi^{t-1}, a^{t-1}, z^{t-1}) \tag{2.12}$$

Esta afirmação é trivialmente verdadeira em  $t = 1$ , pois no início o veículo não tem conhecimento sobre qualquer marco, e, portanto, todas as estimativas são independentes. No instante  $t$ , a distribuição de probabilidade posterior é da seguinte forma:

$$\begin{aligned}
p(\Theta|\chi^t, a^t, z^t) &= \frac{p(z_t|\Theta, \chi^t, a^t, z^{t-1})p(\Theta|\chi^t, a^t, z^{t-1})}{p(z_t|\chi^t, a^t, z^{t-1})} \\
&= \frac{p(z_t|\chi_t, \theta_{a_t}, a_t)p(\Theta|\chi^{t-1}, a^{t-1}, z^{t-1})}{p(z_t|\chi^t, a^t, z^{t-1})}
\end{aligned} \tag{2.13}$$

A ligação da hipótese indutiva da equação 2.12 fornece:

$$\begin{aligned}
p(\Theta|\chi^t, a^t, z^t) &= \frac{p(z_t|\chi_t, \theta_{a_t}, a_t)}{p(z_t|\chi^t, a^t, z^{t-1})} \prod_{n=1}^N p(\theta_n|\chi^{t-1}, a^{t-1}, z^{t-1}) \\
&= \frac{p(z_t|\chi_t, \theta_{a_t}, a_t)}{p(z_t|\chi^t, a^t, z^{t-1})} \underbrace{p(\theta_{a_t}|\chi^{t-1}, a^{t-1}, z^{t-1})}_{Eq.2.11} \prod_{n \neq a_t} \underbrace{p(\theta_n|\chi^{t-1}, a^{t-1}, z^{t-1})}_{Eq.2.9} \\
&= p(\theta_{a_t}|\chi^t, a^t, z^t) \prod_{n \neq a_t} p(\theta_n|\chi^t, a^t, z^t) = \prod_{n=1}^N p(\theta_n|\chi^t, a^t, z^t)
\end{aligned} \tag{2.14}$$

Note-se que as equações 2.11 e 2.9 foram substituídas como indicado. Isto mostra a exatidão da equação 2.8. A correção da equação principal 2.7 segue diretamente deste resultado e da seguinte transformação genérica:

$$\begin{aligned}
p(\chi^t, \Theta|a^t, z^t, u^t) &= p(\chi^t|a^t, z^t, u^t)p(\Theta|\chi^t, a^t, z^t, u^t) = p(\chi^t|a^t, z^t, u^t)p(\Theta|\chi^t, a^t, z^t) \\
&= p(\chi^t|a^t, z^t, u^t) \prod_{n=1}^N p(\theta_n|\chi^t, a^t, z^t)
\end{aligned} \tag{2.15}$$

Note que o condicionamento de todo o caminho  $\chi^t$  é de fato essencial para este resultado.

## 2.4 FastSLAM - associação de dados conhecida

A natureza fatorial da distribuição de probabilidade posterior oferece significativas vantagens. O FastSLAM explora a representação fatorada através da manutenção de  $N + 1$  filtros, um para cada um dos fatores na equação 2.7. Ao fazer isso, todos os  $N + 1$  filtros são de baixa dimensão.

Mais especificamente, o FastSLAM estima a distribuição de probabilidade posterior sobre os caminhos dos veículos usando um filtro de partículas com  $M$  partículas. A distribuição de probabilidade posterior sobre a localização dos  $N$  marcos são estimados por meio de  $N$  filtros de Kalman Extendidos (EKFs). Cada EKF estima a pose um único marco, e portanto, ele tem uma baixa dimensão. Os EKFs individuais são condicionados aos caminhos do veículo. Portanto, cada partícula possui seu próprio conjunto de EKFs. No total existem  $NM$  EKFs, um para cada marco no mapa.

Cada partícula no FastSLAM é da forma:

$$S_t^{[m]} = \left[ \chi^{t,[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}}_{\text{marco.1}}, \overbrace{\psi_{2,t}^{[m]}, \Sigma_{2,t}^{[m]}}^{\text{marco.2}}, \dots, \psi_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \right] \tag{2.16}$$

A notação  $[m]$  indica o índice da partícula;  $\chi^{t,[m]}$  é o seu caminho estimado; e  $\psi_{n,t}^{[m]}$  e  $\Sigma_{n,t}^{[m]}$  são, respectivamente, a média e a variância da representação Gaussiana da característica do marco  $\theta_n$  (suas coordenadas). Juntas, estas quantidades formam a  $m$ -ésima partícula  $S_t^{[m]}$ , de um total de  $M$ , utilizadas na computação da distribuição da probabilidade posterior do FastSLAM.

A Filtragem, isto é, o cálculo da distribuição de probabilidade posterior no instante  $t$  da distribuição no instante  $t - 1$  envolve gerar um novo conjunto de partículas  $S_t$ , a partir de  $S_{t-1}$ , o conjunto de partículas do passo anterior. O novo conjunto de partículas incorpora a última ação de controle  $u_t$  e observação  $z_t$  (com a correspondente associação de dados  $a_t$  entre o marco observado e o marco no mapa  $\Theta$ ). Esta atualização é executada nos passos discriminados nas seções 2.4.1, 2.4.2, 2.4.3, 2.4.4 e 2.4.5:

### 2.4.1 Amostragem de uma nova pose

O conjunto de partículas  $S_t$  é calculada incrementalmente, com base no conjunto  $S_{t-1}$ , observação  $z_t$ , e ação de controle  $u_t$ . Uma vez que não se pode projetar amostras a partir da distribuição de probabilidade posterior do SLAM no instante  $t$ , as amostras serão projetadas a partir de uma distribuição mais simples chamada de distribuição proposta, corrigindo a diferença por meio de uma técnica chamada de amostragem por importância.

Em geral, a amostragem por importância é um algoritmo para se projetar amostras de funções para o qual não existe um procedimento de amostragem direta. Para cada amostra obtida da distribuição proposta, é atribuído um peso igual à razão da distribuição de probabilidade posterior e à distribuição proposta naquele ponto do espaço amostral (Eq.2.37). A partir daí – em um processo conhecido com reamostragem (seção 2.4.4) – um novo conjunto de amostras não ponderadas (sem um peso) é obtida do conjunto de amostras ponderado com probabilidade proporcional aos pesos.

A distribuição proposta do FastSLAM gera – para cada partícula  $S_{t-1}^{[m]}$  – *chutes* (*palpites*) para a estimativa da pose do veículo no instante  $t$  ( $\chi_t^{[m]}$ ). Este *chute* é obtido a partir da amostragem do modelo de movimento probabilístico (Eq.2.2):

$$\chi_t^{[m]} \sim p(\chi_t | \chi_{t-1}, u_t) \quad (2.17)$$

Esta estimativa é adicionada ao conjunto temporário de partículas, junto com o caminho  $\chi^{t-1, [m]}$ . Sobre a hipótese de que o conjunto de partículas  $S_{t-1}$  seja distribuído de acordo com  $p(\chi^{t-1} | z^{t-1}, u^{t-1}, a^{t-1})$ , que é assintoticamente correta, as novas partículas projetadas da distribuição proposta são distribuídas de acordo com:

$$p(\chi^t | z^{t-1}, u^t, a^{t-1}) \quad (2.18)$$

É importante notar que o modelo de movimento pode ser qualquer função não-linear. Isto contrasta com o EKF-SLAM [11], que requer um modelo de

movimento linearizado. Independentemente da distribuição proposta, projetar uma nova pose é uma operação de tempo constante para toda partícula, não depende do tamanho do mapa. O passo de amostragem é descrito graficamente na figura 2.4, em que os pontos são amostras (ou partículas) que foram projetadas a partir de um modelo de movimento probabilístico.

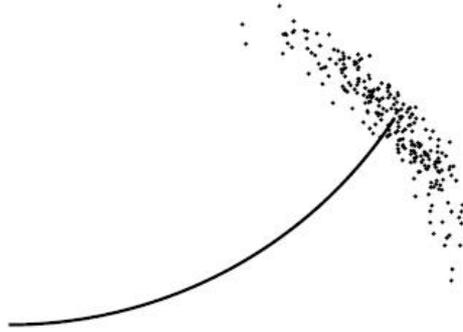


Figura 2.4: Amostras (ou partículas) projetadas de um modelo de movimento probabilístico [18].

## 2.4.2 Atualizando as estimativas de um marco

O FastSLAM representa a estimativa condicional dos marcos  $p(\theta_n | \chi^t, z^t, u^t, a^t)$  usando EKF's de baixa dimensão. Nesta seção, é assumido que as associações de dados  $a^t$  (entre um marco observado no ambiente e um marco  $\theta_{a_t}$  no mapa  $\Theta$ ) são conhecidas. Na seção 2.5, esta restrição será removida.

Uma vez que as estimativas dos marcos são condicionadas ao caminho do veículo, existem  $N$  EKF's em cada cada partícula de  $S_t$ . A distribuição de probabilidade posterior sobre a posição do  $n$ -ésimo marco  $\theta_n$  é facilmente obtida. Seu cálculo depende da condição  $n = a_t$ , isto é, se o marco  $\theta_{a_t=n}$  do mapa  $\Theta$  foi observado ou não no instante  $t$ . Para o marco observado  $\theta_{a_t}$ , segue-se o procedimento usual de expansão da distribuição de probabilidade posterior através da regra de Bayes.

$$p(\theta_{a_t} | \chi^t, z^t, u^t, a^t) = \eta p(z_t | \theta_{a_t}, \chi^t, z^{t-1}, u^t, a^t) p(\theta_{a_t} | \chi^t, z^{t-1}, u^t, a^t) \quad (2.19)$$

A observação  $z_t$  depende somente de  $\theta_{a_t}$ ,  $\chi_t$  e  $a_t$ . Similarmente,  $\theta_{a_t}$  não é afetada por  $\chi_t$ ,  $u_t$ , ou  $a_t$  sem a observação  $z_t$ .

$$p(\theta_{a_t} | \chi^t, z^t, u^t, a^t) = \underbrace{\eta p(z_t | \theta_{a_t}, \chi_t, a_t)}_{1^\circ \text{ termo}} \overbrace{p(\theta_{a_t} | \chi^{t-1}, z^{t-1}, u^{t-1}, a^{t-1})}^{2^\circ \text{ termo}} \quad (2.20)$$

Para  $n \neq a_t$ , a estimativa posterior do marco permanece inalterada:

$$p(\theta_{n \neq a_t} | \chi^t, z^t, u^t, a^t) = p(\theta_{n \neq a_t} | \chi^{t-1}, z^{t-1}, u^{t-1}, a^{t-1}) \quad (2.21)$$

O FastSLAM implementa a atualização da equação 2.20 usando um EKF. Como na solução EKF para o SLAM, este filtro usa uma aproximação Gaussiana linear para o modelo de observação. Note que, com um modelo de observação Gaussiano linear, a distribuição resultante  $p(\theta_{a_t} | \chi^t, z^t, u^t, a^t)$  é uma Gaussiana exata, mesmo que o modelo de movimento seja não linear. Esta é uma consequência da amostragem sobre a pose do veículo.

O modelo de observação não linear  $g(\chi_t, \theta_{a_t})$  é aproximado usando a expansão de primeira ordem de Taylor. O estimador do marco está condicionado a um caminho fixo do veículo, tal que esta expansão é somente sobre  $\theta_{a_t}$ <sup>5</sup>:

$$\hat{z}_t = g(\chi_t^{[m]}, \psi_{a_t, t-1}^{[m]}) \quad (2.22)$$

$$G_{\theta_{a_t}} = \nabla_{\theta_{a_t}} g(\chi_t, \theta_{a_t}) \Big|_{\chi_t = \chi_t^{[m]}, \theta_{a_t} = \psi_{a_t, t-1}^{[m]}} \quad (2.23)$$

$$g(\chi_t, \theta_{a_t}) \approx \hat{z}_t + G_{\theta}(\theta_{a_t} - \psi_{a_t, t-1}^{[m]}) \quad (2.24)$$

Sob esta aproximação, o primeiro termo do produto em 2.20 é distribuído de acordo com a equação 2.25. Assume-se que o ruído de observação é Gaussiano com covariância  $R_t$ :

$$p(z_t | \theta_i, \chi_t, a_t) \sim N(z_t; \hat{z}_t + G_{\theta}(\theta_{a_t} - \psi_{a_t, t-1}^{[m]}), R_t) \quad (2.25)$$

O segundo termo do produto em 2.20 é também uma Gaussiana, igual ao estado do EKF no instante  $t - 1$ :

$$p(\theta_{a_t} | \chi^{t-1}, z^{t-1}, u^{t-1}, a^{t-1}) \sim N(\theta_{a_t}; \psi_{a_t, t-1}^{[m]}, \Sigma_{a_t, t-1}^{[m]}) \quad (2.26)$$

A média e a covariância do produto pode ser obtida usando equações padrões de atualização do EKF:

$$\hat{z}_t = g(\chi_t^{[m]}, \psi_{a_t, t-1}^{[m]}) \quad (2.27)$$

---

<sup>5</sup>Na equação 2.23, uma matriz Jacobiana é uma matriz formada pelas derivadas parciais de primeira ordem de uma função vetorial

$$G_{\theta_{a_t}} = \nabla_{\theta_{a_t}} g(\chi_t, \theta_{a_t}) \Big|_{\chi_t = \chi_t^{[m]}, \theta_{a_t} = \psi_{a_t, t-1}^{[m]}} \quad (2.28)$$

$$Z_{n,t} = G_{\theta_{a_t}} \Sigma_{a_t, t-1}^{[m]} G_{\theta_{n_t}}^T + R_t \quad (2.29)$$

$$K_t = \Sigma_{a_t, t-1}^{[m]} G_{\theta_{a_t}}^T Z_{n,t}^{-1} \quad (2.30)$$

$$\psi_{a_t, t}^{[m]} = \psi_{a_t, t-1}^{[m]} + K_t (z_t - \hat{z}_t) \quad (2.31)$$

$$\Sigma_{a_t, t}^{[m]} = (I - K_t G_{\theta_{a_t}}) \Sigma_{a_t, t-1}^{[m]} \quad (2.32)$$

Atualizar os filtros dos marcos é uma operação de tempo constante por partícula, uma vez que cada filtro de marco possui um tamanho fixo. A quantidade de tempo necessário para incorporar uma observação não depende do número total de marcos.

No caso planar do SLAM, o veículo normalmente observa a distância e a orientação dos marcos como mostrado na figura 2.2. Assumindo que a pose do veículo é descrita como  $\chi_t = (x, y, \varphi)$  e a posição do marco observado  $\theta_j$  no mapa de características  $\Theta$  seja  $(\psi_{a_t, x}, \psi_{a_t, y})$ , a função  $g(\chi_t, \theta_{a_t})$  pode ser escrita como:

$$g(\chi_t, \theta_{a_t}) = \begin{bmatrix} \tau(\chi_t, \theta_{a_t}) \\ \phi(\chi_t, \theta_{a_t}) \end{bmatrix} \quad (2.33)$$

Estes valores podem ser preditos pela seguinte equação:

$$\begin{bmatrix} \tau(\chi_t, \theta_{a_t}) \\ \phi(\chi_t, \theta_{a_t}) \end{bmatrix} = \begin{bmatrix} \sqrt{(\psi_{a_t, x} - x)^2 + (\psi_{a_t, y} - y)^2} \\ \tan^{-1} \frac{(\psi_{a_t, y} - y)}{(\psi_{a_t, x} - x)} - \varphi \end{bmatrix} \quad (2.34)$$

O Jacobiano  $G_{\theta_{a_t}}$  é então igual a:

$$G_{\theta_{a_t}} = \begin{bmatrix} \frac{\partial \tau}{\partial \psi_{a_t, x}} & \frac{\partial \tau}{\partial \psi_{a_t, y}} \\ \frac{\partial \phi}{\partial \psi_{a_t, x}} & \frac{\partial \phi}{\partial \psi_{a_t, y}} \end{bmatrix} = \begin{bmatrix} \frac{\psi_{a_t, x} - x}{\tau} & \frac{\psi_{a_t, y} - y}{\tau} \\ -\frac{\psi_{a_t, y} - y}{\tau^2} & \frac{\psi_{a_t, x} - x}{\tau^2} \end{bmatrix} \quad (2.35)$$

Fazendo  $d_x = \psi_{a_t,x} - x$  e  $d_y = \psi_{a_t,y} - y$  temos:

$$G_{\theta_{a_t}} = \begin{bmatrix} \frac{d_x}{\tau} & \frac{d_y}{\tau} \\ -\frac{d_y}{\tau^2} & \frac{d_x}{\tau^2} \end{bmatrix} \quad (2.36)$$

A prova da correção de cada um dos termos da equação 2.36 é como se segue:

$$\begin{aligned} - \frac{\partial \tau}{\partial \psi_{a_t,x}} &= \frac{1}{2} ((\psi_{a_t,x} - x)^2 + \psi_{a_t,y} - y)^2)^{-\frac{1}{2}} (2)(\psi_{a_t,x} - x) = \frac{d_x}{\tau} \\ - \frac{\partial \tau}{\partial \psi_{a_t,y}} &= \frac{1}{2} ((\psi_{a_t,x} - x)^2 + \psi_{a_t,y} - y)^2)^{-\frac{1}{2}} (2)(\psi_{a_t,y} - y) = \frac{d_y}{\tau} \\ - \frac{\partial \phi}{\partial \psi_{a_t,x}} &= \frac{\frac{\partial \left( \frac{\psi_{a_t,y}-y}{\psi_{a_t,x}-x} \right)}{\partial \psi_{a_t,x}}}{1 + \left( \frac{\psi_{a_t,y}-y}{\psi_{a_t,x}-x} \right)^2} = \frac{\frac{(\psi_{a_t,x}-x)(0) - (\psi_{a_t,y}-y)(1)}{(\psi_{a_t,x}-x)^2}}{(\psi_{a_t,x}-x)^2 + (\psi_{a_t,y}-y)^2} = -\frac{d_y}{\tau^2} \\ - \frac{\partial \phi}{\partial \psi_{a_t,y}} &= \frac{\frac{\partial \left( \frac{\psi_{a_t,y}-y}{\psi_{a_t,x}-x} \right)}{\partial \psi_{a_t,y}}}{1 + \left( \frac{\psi_{a_t,y}-y}{\psi_{a_t,x}-x} \right)^2} = \frac{\frac{(\psi_{a_t,x}-x)(1) - (\psi_{a_t,y}-y)(0)}{(\psi_{a_t,x}-x)^2}}{(\psi_{a_t,x}-x)^2 + (\psi_{a_t,y}-y)^2} = \frac{d_x}{\tau^2} \end{aligned}$$

### 2.4.3 Calculando o peso das partículas

Amostras de uma distribuição proposta são distribuídas de acordo com  $p(\chi^t | z^{t-1}, u^t, a^{t-1})$ , e portanto não corresponde à distribuição posterior desejada  $p(\chi^t | z^t, u^t, a^t)$ . Esta diferença é corrigida através da amostragem por importância. Um exemplo de amostragem por importância é mostrada na figura 2.5. Ao invés de se amostrar diretamente a distribuição meta (linha sólida), amostras são obtidas de uma distribuição proposta mais simples, uma Gaussiana (linha tracejada). Nas regiões onde a distribuição meta é maior do que a distribuição proposta, as amostras recebem pesos maiores. Em consequência, amostras nesta região serão escolhidas com maior frequência. Em regiões onde a distribuição meta é menor do que a distribuição proposta, as amostras receberão pesos menores. No limite de um conjunto infinito de amostras, este procedimento produzirá amostras distribuídas de acordo com a distribuição meta.

No FastSLAM, o peso de cada partícula  $w_t^{[i]}$  é igual à razão entre a distribuição de probabilidade posterior do SLAM e a distribuição de probabilidade proposta descrita anteriormente.

$$w_t^{[m]} = \frac{\text{distribuição meta}}{\text{distribuição proposta}} = \frac{p(\chi^{t,[m]} | z^t, u^t, a^t)}{p(\chi^{t,[m]} | z^{t-1}, u^t, a^{t-1})} \quad (2.37)$$

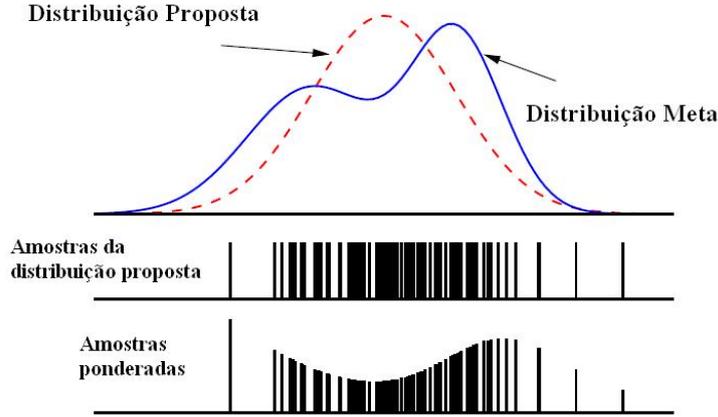


Figura 2.5: O amostrador por importância projeta amostras da distribuição proposta (Gaussiana - linha pontilhada) com quantidades proporcionais aos seus pesos, que são maiores na região onde a distribuição meta (linha sólida) é maior que a proposta [18].

O numerador da equação 2.37 pode ser expandido usando a regra de Bayes. A constante de normalização da regra de Bayes pode ser seguramente ignorada porque os pesos das partículas serão normalizados antes da reamostragem (seção 2.4.4):

$$w_t^{[m]} \propto \frac{p(z_t | \chi^{t,[m]}, z^{t-1}, u^t, a^t) p(\chi^{t,[m]} | z^{t-1}, u^t, a^t)}{p(\chi^{t,[m]} | z^{t-1}, u^t, a^{t-1})} \quad (2.38)$$

O segundo termo do numerador não está condicionado à última observação  $z_t$ , tal que a associação de dados  $a_t$  não pode fornecer qualquer informação sobre o caminho do veículo. Portanto, pode ser retirado:

$$w_t^{[m]} = \frac{p(z_t | \chi^{t,[m]}, z^{t-1}, u^t, a^t) p(\chi^{t,[m]} | z^{t-1}, u^t, a^{t-1})}{p(\chi^{t,[m]} | z^{t-1}, u^t, a^{t-1})} = p(z_t | \chi^{t,[m]}, z^{t-1}, u^t, a^t) \quad (2.39)$$

O estimador do marco é um EKF, tal que esta probabilidade da observação  $z_t$  pode ser computada de forma fechada. Esta probabilidade é normalmente computada em termos de *inovação*, ou a diferença entre a observação real  $z_t$  e a observação predita  $\hat{z}_t$ . A seqüência de inovações no EKF é uma Gaussiana com média zero e covariância  $Z_{a_t,t}$ , onde  $Z_{a_t,t}$  é a matriz de covariância de inovação definida em 2.29. A probabilidade da observação  $z_t$

ser igual à probabilidade da inovação  $z_t - \hat{z}_t$  ser gerada por esta Gaussiana, pode ser escrita como:

$$w_t^{[m]} = \frac{\exp \left\{ -\frac{1}{2} (z_t - \hat{z}_{a_t,t})^T Z_{a_t,t}^{-1} (z_t - \hat{z}_{a_t,t}) \right\}}{\sqrt{|2\pi Z_{a_t,t}|}} \quad (2.40)$$

Calcular o peso é uma operação de tempo constante por partícula.

#### 2.4.4 Reamostragem por importância

Uma vez que foram atribuídos pesos às partículas do conjunto  $S_{t-1}$ , um novo conjunto de partículas  $S_t$  é projetado deste conjunto (por substituição), com probabilidade proporcional ao peso das partículas.

A necessidade de se reamostrar o conjunto de partículas, se deve ao fato de que algumas partículas no conjunto não correspondem à distribuição posterior desejada. A reamostragem é uma técnica comum em filtragem de partículas para corrigir tais erros de correspondência e evitar a degeneração das partículas. Reamostrando as partículas de acordo com seus respectivos pesos, o conjunto de partículas resultante se aproxima da distribuição meta (Fig.2.5).

Na seção 2.8, é apresentada uma técnica de reamostragem utilizada neste trabalho.

#### 2.4.5 Distribuição de probabilidade posterior sobre o caminho do veículo (revisitada)

À primeira vista, fatorar o problema do SLAM usando o caminho do veículo pode parecer uma má idéia, uma vez que o tamanho das partículas crescerá com o tempo (Eq.2.16). Contudo, nenhuma das equações de atualização do FastSLAM depende do tamanho do caminho no instante  $t$ . De fato, somente a pose mais recente  $\chi_{t-1}^{[m]}$  é usada para atualizar o conjunto de partículas. Em conseqüência, considera-se somente a pose mais recente de cada partícula (Eq.2.41). Isto evita o problema computacional que resultaria se a dimensão do filtro de partícula crescesse ao longo do tempo.

$$S_t^{[m]} = \left[ \underbrace{\chi_{t,[m]}, \psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}}_{\text{marco.1}}, \chi_{t,[m]}, \overbrace{\psi_{2,t}^{[m]}, \Sigma_{2,t}^{[m]}}^{\text{marco.2}}, \dots, \psi_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \right] \quad (2.41)$$

## 2.5 FastSLAM - Associação de dados desconhecida

A maior limitação do algoritmo FastSLAM descrito até agora é a assunção de que a associação de dados  $a_t$  (entre um marco observado e um marco no mapa  $\Theta$ ) é conhecida. Na prática, este é raramente o caso. Esta seção estende o algoritmo FastSLAM para o domínio em que a correspondência entre observações e marcos não é conhecida. A solução clássica para o problema da associação de dados no FastSLAM é escolher  $a_t$  tal que maximize a probabilidade da observação do sensor  $z_t$  a partir de todos os dados disponíveis:

$$a_t = \underbrace{\arg\max}_{n_t} p(z_t | n_t, \chi^t, z^{t-1}, u^t) \quad (2.42)$$

O termo  $p(z_t | n_t, a^{t-1}, \chi^t, z^{t-1}, u^t)$  é referenciado como verossimilhança, e esta abordagem é um exemplo de um estimador de máxima verossimilhança. Este tipo de associação de dados é também denominada de associação de dados pelo *vizinho mais próximo*, caso se interprete o *log* negativo da verossimilhança como uma função distância. Para Gaussianas, o *log* negativo da verossimilhança é a distância de *Mahalanobis*.

### 2.5.1 Incerteza na associação de dados

Dois fatores contribuem para a incerteza na probabilidade posterior do SLAM: ruído na medição e ruído no movimento. A medida que o ruído na medição aumenta, a distribuição das possíveis observações de cada marco se torna mais incerta. Se o ruído de medição é suficientemente alto, as distribuições de observações de marcos próximos começarão a se sobrepor de maneira substancial. Esta sobreposição conduz a ambiguidade na identidade  $a_t$  dos marcos. A ambiguidade de associação de dados causada pelo ruído na medição é conhecido como *ambiguidade de medição*. Um exemplo de ambiguidade de medição é mostrado na figura 2.6. As duas elipses descrevem o alcance de prováveis observações de dois diferentes marcos. A observação, mostrada como um círculo negro, poderia de maneira plausível ter vindo de um dos dois marcos.

Atribuir uma observação para um marco errado devido a ambiguidade de medição aumentará o erro do mapa e da pose do veículo, mas seu impacto será relativamente pequeno (em relação à ambigüidade devido a um ruído de movimento). Dado que a observação pode ter sido gerada por qualquer um dos marcos com alta probabilidade, o efeito da observação sobre as posições dos marcos e a pose do veículo é pequena. A covariância de um marco será

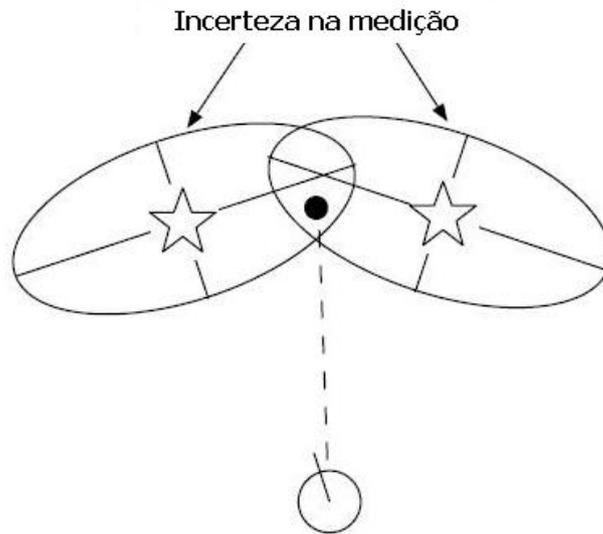


Figura 2.6: Ambiguidade na medição. Grande erros de medição conduzem a uma ambiguidade entre marcos próximos [18].

ligeiramente sobreestimada, enquanto a covariância do segundo será ligeiramente subestimada. Se múltiplas observações são incorporadas por ação de controle  $u_t$ , um erro de associação de dados devido a ambiguidade de medição terá pouco impacto sobre a decisão de associação de dados para as outras observações.

A ambiguidade de associação de dados devido a um ruído no movimento (*ambiguidade de movimento*) pode ter conseqüências mais graves sobre a acurácia da estimação. Ruídos de movimento maiores conduzirão a maiores incertezas na pose após a incorporação de um controle. Uma incerteza na pose do veículo bastante alta, implicará em hipóteses de associação de dados por máxima verossimilhança drasticamente diferentes para observações subseqüentes. Esta ambiguidade no movimento, mostrada na figura 2.7, é facilmente induzida se há um significativo erro rotacional no movimento do veículo. Além disso, se múltiplas observações do veículo são incorporadas por controle, a pose do veículo correlacionará a decisão de associação de dados para todas as observações.

### 2.5.2 Associação de dados por partícula

O FastSLAM adota uma abordagem de hipótese múltipla de associação de dados para o problema de associação de dados. Cada partícula representa uma diferente hipótese para o caminho do veículo, tal que a decisão de associação de dados pode ser feita por partícula. Partículas que escolhem a correta

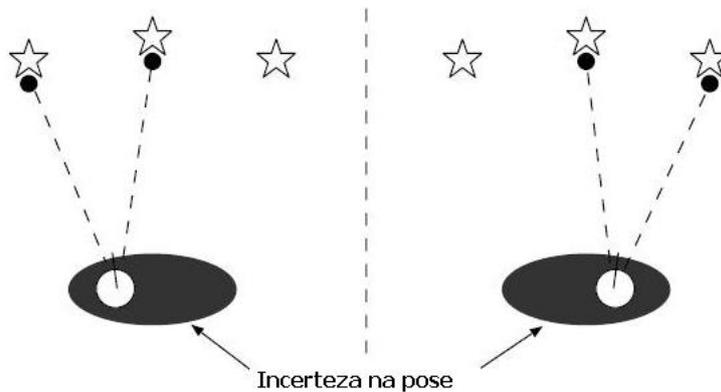


Figura 2.7: Incerteza por ambiguidade do movimento devido ao movimento do veículo poder resultar em diferentes hipóteses de associação de dados para diferentes poses do veículo [18].

associação de dados receberão pesos mais altos porque elas explicam bem a observação. Partículas que escolhem associação de dados errônea receberão baixos pesos e serão removidas no passo de reamostragem.

Associação de dados por partículas tem muitas vantagens sobre a associação de dados pela máxima verossimilhança. Primeiro, ela fatora o problema da incerteza da pose do veículo fora do problema de associação de dados. Ambiguidade do movimento é a mais grave forma de ambiguidade de associação de dados. Assim sendo, condicionar a decisão sobre a associação de dados aos hipotéticos caminhos do veículo parece ser uma escolha lógica. Dado um cenário como o da figura 2.7, algumas partículas projetarão novas poses do veículo consistentes com a hipótese de associação de dados à esquerda enquanto outras projetariam poses de forma consistente com a hipótese de associação de dados à direita.

Fazer a associação de dados com base nas partículas também torna o problema da associação de dados mais fácil. No FastSLAM, a incerteza na pose do veículo é representada por todo conjunto de partículas. Os filtros de marcos em uma única partícula não são afetados pelo ruído de movimento porque eles estão condicionados a um caminho específico do veículo. Isto é especialmente útil se o veículo tem movimento ruidoso e um sensor preciso.

Outra consequência da associação de dados por partícula que está implícita é o retardo na tomada de decisão. Em um dado instante, alguma fração de partículas receberá, ainda que de forma errada, associações de dados. No futuro, o veículo pode receber uma nova observação que claramente refuta essas atribuições anteriores. Neste ponto, as partículas com associação de dados erradas receberão pesos baixos e provavelmente serão removidas do filtro. Como resultado deste processo, o efeito de uma decisão de associ-

ação de dados errada feita no passado pode ser removida do filtro. Além disso, nenhuma heurística é necessária para remover antigas associações de dados errôneas do filtro. Isto é feito estatisticamente, de maneira válida, simplesmente como uma consequência do passo de reamostragem.

### 2.5.3 Associação de dados por máxima verossimilhança por partícula

A abordagem mais simples de associação de dados por partículas é feita pela aplicação da heurística de associação de dados por máxima verossimilhança (Eq.2.42), por partícula. Desde que os estimadores de marcos são EKF's, a verossimilhança na equação 2.42 pode ser calculada usando inovações. Esta probabilidade é igual ao fator de importância calculado na equação 2.40. Se o valor desta probabilidade (reescrita na Eq.2.43) cair abaixo de algum limite  $P_0$ , um novo marco é adicionado para a partícula.

$$p(z_t | \chi^{t,[m]}, z^{t-1}, u^t, a_{t-1}) = \frac{\exp \left\{ -\frac{1}{2} (z_t - \hat{z}_{n,t})^T Z_{n,t}^{-1} (z_t - \hat{z}_{n,t}) \right\}}{\sqrt{|2\pi Z_{n,t}|}} \quad (2.43)$$

## 2.6 Adicionando novos marcos

Se a função de observação  $g(\theta_{a_t}, \chi_t)$  é inversível, uma única observação é suficiente para inicializar um novo marco. Cada observação define uma Gaussiana:

$$N(z_t; \hat{z}_t + G_{\theta_{a_t}}(\theta_{a_t} - \psi_{a_t,t-1}^{[m]}), R_t) \quad (2.44)$$

Esta Gaussiana pode ser escrita como<sup>6</sup>:

$$\frac{\exp \left\{ -\frac{1}{2} (z_t - \hat{z}_t - G_{\theta_{a_t}}(\theta_{a_t} - \psi_{a_t,t-1}^{[m]}))^T R_t^{-1} (z_t - \hat{z}_t - G_{\theta_{a_t}}(\theta_{a_t} - \psi_{a_t,t-1}^{[m]})) \right\}}{|2\pi \Sigma|^{\frac{1}{2}}} \quad (2.45)$$

Definindo  $J$  como o negativo do expoente desta Gaussiana:

$$J = \left\{ \frac{1}{2} (z_t - \hat{z}_t - G_{\theta_{a_t}}(\theta_{a_t} - \psi_{a_t,t-1}^{[m]}))^T R_t^{-1} (z_t - \hat{z}_t - G_{\theta_{a_t}}(\theta_{a_t} - \psi_{a_t,t-1}^{[m]})) \right\} \quad (2.46)$$

---

<sup>6</sup>A distribuição normal multivariada de uma variável  $x$  com média  $\psi$  e covariância  $R_t$  é da forma  $\frac{\exp \left\{ -\frac{1}{2} (x-\psi)^T R_t^{-1} (x-\psi) \right\}}{|2\pi R_t|^{\frac{1}{2}}}$

A segunda derivada de  $J$  com relação a  $\theta_{a_t}$  será a inversa da matriz de covariância da Gaussiana das coordenadas do marco.

$$\frac{\partial^2 J}{\partial \theta_{a_t}^2} = G_{\theta_{a_t}}^T R_t^{-1} G_{\theta_{a_t}} \quad (2.47)$$

Em conseqüência, uma observação inversível pode ser usada para a criação de um novo marco como segue:

$$\psi_{a_t,t}^{[m]} = g^{-1}(\chi_t^{[m]}, z_t) \quad (2.48)$$

$$\Sigma_{a_t,t}^{[m]} = (G_{\theta_{a_t}}^T R_t^{-1} G_{\theta_{a_t}})^{-1} \quad (2.49)$$

O Jacobiano utilizado na equação 2.49 é computado como se segue:

$$G_{\theta_{a_t}} = \begin{bmatrix} \cos(\phi + \alpha) & \sin(\phi + \alpha) \\ -\frac{\sin(\phi + \alpha)}{\tau} & \frac{\cos(\phi + \alpha)}{\tau} \end{bmatrix} \quad (2.50)$$

A correção de cada um dos termos da equação 2.50 é descrita abaixo. Eles são decorrentes da equação 2.36 e do modelo de observação (Fig.2.2):

- $\cos(\phi + \alpha) = \frac{d_x}{\tau}$
- $\sin(\phi + \alpha) = \frac{d_y}{\tau}$
- $-\frac{\sin(\phi + \alpha)}{\tau} = -\frac{d_y}{\tau^2}$
- $\frac{\cos(\phi + \alpha)}{\tau} = \frac{d_x}{\tau^2}$

## 2.7 Algoritmo FastSLAM (marcos 2D)

Na seqüência, apresentamos o algoritmo FastSLAM (Algoritmo 2.1) com associação de dados desconhecida. As partículas no algoritmo têm a seguinte forma:

$$\left[ \chi_t^{[m]}, N_t^{[m]}, \psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \psi_{N_t^{[m]},t}^{[m]}, \Sigma_{N_t^{[m]},t}^{[m]} \right] \quad (2.51)$$

Em adição à pose do veículo  $\chi_t^{[m]}$  e às estimativas do marco,  $\psi_{n,t}^{[m]}$  e  $\Sigma_{n,t}^{[m]}$ , cada partícula mantém o número de marcos  $N_t^{[m]}$  no mapa. É também interessante notar que cada partícula pode ter um número diferente de marcos.

FastSLAM ( $S_{t-1}, u_t, R, z_t$ )

1.  $S_t = S_{aux} = \emptyset$
2. Para  $m = 1$  até  $M$  fazer *{Loop: todas partículas}*
  - (a) Recupere  $\left[ \chi_{t-1}^{[m]}, N_{t-1}^{[m]}, \psi_{1,t-1}^{[m]}, \Sigma_{1,t-1}^{[m]}, \dots, \psi_{N_{t-1}^{[m]},t-1}^{[m]}, \Sigma_{N_{t-1}^{[m]},t-1}^{[m]}, w_{t-1}^{[m]} \right]$  de  $S_{t-1}$
  - (b) Amostre  $\chi_t^{[m]} \sim p(\chi_t | \chi_{t-1}^{[m]}, u_t)$  *{Amostragem de nova pose (Eq.2.2)}*
  - (c) Para  $n = 1$  até  $N_{t-1}^{[m]}$  fazer *{Loop sobre potenciais associações de dados}*
    - a.  $\hat{z}_{n,t} = g(\chi_t^{[m]}, \psi_{n,t-1}^{[m]})$  (Eq.2.27)
    - b.  $G_{\theta,n} = \nabla_{\theta} g(\theta_n, \chi_t) |_{\theta_n = \psi_{n,t-1}^{[m]}, \chi_t = \chi_t^{[m]}}$  (Eq.2.28)
    - c.  $Z_{n,t} = G_{\theta,n} \Sigma_{n,t-1}^{[m]} G_{\theta,n}^T + R$  (Eq.2.29)
    - d.  $p_{n,t}^{[m]} = |2\pi Z_{n,t}|^{-\frac{1}{2}} \exp \left\{ \frac{1}{2} (z_t - \hat{z}_{n,t})^T Z_{n,t}^{-1} (z_t - \hat{z}_{n,t}) \right\}$  (Eq.2.43)
  - (d) Fim Para
  - (e)  $a_t = \arg\_max_n p_{n,t}^{[m]}$  *{Associação de dados}*
  - (f) Se  $p_{n,t}^{[m]} > P_0$ 
    - a.  $N_t^{[m]} = N_{t-1}^{[m]}$  *{É um Marco conhecido: atualização}*
    - b.  $K_{a_t,t}^{[m]} = \Sigma_{a_t,t-1} G_{\theta,a_t}^T Z_{a_t,t}^{-1}$  (Eq.2.30)
    - c.  $\psi_{a_t,t}^{[m]} = \psi_{a_t,t-1}^{[m]} + K_{a_t,t} (z_t - \hat{z}_{a_t,t})$  (Eq.2.31)
    - d.  $\Sigma_{a_t,t}^{[m]} = (I - K_{a_t,t} G_{\theta,a_t}) \Sigma_{a_t,t-1}^{[m]}$  (Eq.2.32)
    - e.  $w_t^{[m]} = p_{a_t,t}^{[m]}$
  - (g) Senão
    - a.  $N_t^{[m]} = N_{t-1}^{[m]} + 1$  *{É Novo Marco: inicialização}*
    - b.  $\psi_{a_t,t}^{[m]} = g^{-1}(\chi_t^{[m]}, \hat{z}_{a_t,t})$  (Eq.2.48)
    - c.  $\Sigma_{a_t,t}^{[m]} = (G_{\theta,a_t}^T R^{-1} G_{\theta,a_t})^{-1} = G_{\theta,a_t}^{-1} R G_{\theta,a_t}^{-T}$  (Eq.2.49)
  - (h) Fim Se
  - (i) Fim Para
  - (j) Adicionar  $\left[ \chi_t^{[m]}, N_t^{[m]}, \psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \dots, \psi_{N_t^{[m]},t}^{[m]}, \Sigma_{N_t^{[m]},t}^{[m]}, w_t^{[m]} \right]$  a  $S_{aux}$
3. Fim Para
4.  $id = \text{melhor\_partícula}(S_{aux})$
5.  $S_t = \text{Reamostragem}(S_{aux})$
6. Retornar  $S_t$

algoritmo 2.1: Algoritmo FastSLAM

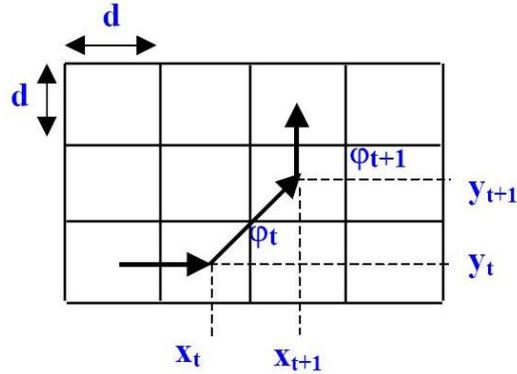


Figura 2.8: Modelo de movimento usado

### 2.7.1 Explicação do Algoritmo 2.1

Para cada partícula  $S_{t-1}^{[m]}$  de  $S_{t-1}$ , é amostrado o modelo de movimento do veículo (passo 2.(b), Eq.2.2). Um exemplo de modelo de movimento com a respectiva amostragem é apresentado na figura 2.8 e tabela 2.1.

Por outro lado, para cada marco  $\theta_n^{[m]}$  no mapa  $\Theta^{[m]}$  é calculada a qualidade da associação de  $\theta_n^{[m]}$  com a observação  $z_t$  de um marco no ambiente, feita através de função do FastSLAM (Passo 2.(c).d).

A estratégia de associação de dados do FastSLAM escolhe como associação correta o marco  $\theta_n$  com probabilidade de associação máxima (passo 2.(e)). Se  $p_{a_t,t}^{[m]}$  é maior que um certo limite  $P_0$  (passo 2.(f)), a associação de dados é considerada correta e a estimativa das coordenadas do marco  $\theta_{a_t}^{[m]}$  é atualizada com equações do EKF (passos 2.(f).a até 2.(f).d). Caso contrário, um novo marco é criado no mapa (passos 2.(g).a até 2.(g).c).

O peso da partícula  $w_t$  corresponde à qualidade da associação  $p_{a_t,t}^{[m]}$  do marco  $\theta_{a_t}^{[m]}$  que é associado com a observação  $z_t$  (passo 2.(f).e).

A melhor partícula, que corresponde à estimativa de localização obtida, é a que possui maior peso (passo 4).

O algoritmo descrito considera apenas uma observação a cada passo. Caso haja mais observações, o peso da partícula (passo 2.(f).e) é calculado como o produto normalizado das respectivas probabilidades de associação.

Na reamostragem (passo 5), partículas com maior peso  $w_t^{[i]}$  são replicadas.

Tabela 2.1: Amostragem dos modelo de movimento (Fig.2.8) e observação (Fig.2.2).  $N(0, \sigma)$  é uma distribuição normal com média zero e desvio padrão  $\sigma$ .

Variável	Erro Médio
$x_{t+1} = x_t + \Delta x_t + \bar{d}N(0, \sigma)$	$\bar{d}$
$y_{t+1} = y_t + \Delta y_t + \bar{d}N(0, \sigma)$	$\bar{d}$
$\varphi_{t+1} = \varphi_t + \Delta \varphi_t + \bar{g}N(0, \sigma)$	$\bar{g}$
$\tau = \tau + \bar{\tau}N(0, \sigma)$	$\bar{\tau}$
$\phi = \phi + \bar{\phi}N(0, \sigma)$	$\bar{\phi}$

## 2.7.2 Detalhes do Algoritmo 2.1

No passo 2(c)a e 2(f)c, a computação da observação predita é como descrito na equação 2.34.

O Jacobiano dos passos 2(c)b, 2(f)b e 2(f)d são calculados a partir da equação 2.35.

A variável  $R$  (passo 2(c)c e 2(g)c) é como descrito na equação 2.52 (Tab.2.1):

$$R = \begin{bmatrix} \bar{\tau}N(0, \sigma) & 0 \\ 0 & \bar{\phi}N(0, \sigma) \end{bmatrix} \quad (2.52)$$

O Jacobiano do passo 2(g)c é calculado a partir da equação 2.50.

## 2.8 Reamostragem Estratificada

A necessidade de se reamostrar o conjunto de partículas, de acordo com os seus respectivos pesos, se deve ao fato de que algumas partículas no conjunto não correspondem à distribuição de probabilidade posterior desejada. A reamostragem é uma técnica comum em filtragem de partículas que é utilizada para corrigir tais erros de correspondência e evitar a degeneração das partículas. Na reamostragem, as partículas do conjunto são replicadas de acordo com seus respectivos pesos.

É importante que se se tenha um critério para a reamostragem das partículas a fim de que ela produza um conjunto de partículas que se aproxime da distribuição meta. A reamostragem deve ocorrer apenas quando o número efetivo de partículas,  $N_{efetivo} = 1 / \sum_{i=1}^M (w_i^{[m]})^2$ , é pequeno [10]. Este número representa o inverso do coeficiente de variação do peso normalizado do conjunto de partículas, ou seja,  $N_{efetivo}$  representa o quanto o peso das partículas

do conjunto são similares: um valor baixo (alto) indica que a variação nos valores dos pesos é grande (pequena).

A estratégia de reamostragem estratificada (Algoritmo 2.2) é apresentada a seguir.

A estratificação é utilizada quando a população se divide em subpopulações (estratos) razoavelmente homogêneos. Ela consiste em se especificar quantos itens da amostra serão retirados de cada estrato, sendo a seleção em cada estrato aleatória. A estratificação pode resultar em um sensível ganho nas estimativas [12].

Após a reamostragem, os pesos das partículas são ajustados para  $w_t^{[m]} = 1/M$  (passo 14.2). Reamostramos cada vez que  $N_{efetivo}$  fica abaixo de um certo limite que configuramos como  $0,5M$  (passo 14). Um limite maior poderia excluir boas partículas. Por outro lado, um limite menor poderia manter no conjunto, uma partícula que não representa a correta distribuição de probabilidade posterior.

A tabela 2.2 ilustra o funcionamento do algoritmo de reamostragem. Na 9ª coluna temos as partículas reamostradas em função das partículas que tiveram marcos atualizados (1ª coluna).

## 2.9 Conclusão

Neste capítulo, apresentamos a técnica FastSLAM em que os marcos 2D são obtidos através de um sensor a laser. No próximo capítulo, apresentamos o algoritmo de visão computacional SIFT (*Scale-Invariant Feature Transform*) que fornece descritores estáveis de características em imagens. Estas características serão utilizadas como marcos no algoritmo FastSLAM Visual (capítulo 4).

Reamostragem ( $S_t$ )

01.  $w_{soma} = \sum_{m=1}^M w_m$
02.  $w_m = \frac{w_m}{w_{soma}}$  ( $\forall m \in M$ )
03.  $N_{efetivo} = \frac{1}{\sum_{m=1}^M w_m^2}$
04.  $w_m = w_m + w_{m-1}$  ( $\forall m > 1 \in M$ )
05.  $k = \frac{1}{M}$
06.  $q = \frac{k}{2}$
07. Para  $m = 1$  até  $M$ 
  1.  $d[m] = q$
  2.  $q = q + k$
08. Fim Para
09. Para  $m = 1$  até  $M$ 
  1.  $selecao[m] = d[m] + k * funcao\_dist\_uniforme - \frac{k}{2}$
10. Fim Para
11.  $controle = 1$
12. Para  $m = 1$  até  $M$  fazer
  1. Enquanto ( $(controle < M)$  e ( $selecao[controle] < w_m$ ))
    1.  $manter[controle] = m$
    2.  $controle = controle + 1$
  2. Fim Enquanto
13. Fim Para
14. Se ( $N_{efetivo} < 0.5M$ )
  1. Para  $m = 1$  até  $N$ 
    1.  $S[m] = S[manter[m]]$
    2.  $w_m = \frac{1}{M}$
  2. Fim Para
15. Fim Se
16. Retornar  $S_t$

algoritmo 2.2: Algoritmo de reamostragem

Tabela 2.2: Reamostragem considerando 1 veículo ( $M = 40$ )

Exemplo 01

m	w	$\frac{w_m}{w_{soma}}$ (passo 2)	d[m] (passo 7.1)	f_dist_unif	selecao[m] (passo 9.1)	w_m (passo 4)	m	manter[m] (passo 12.1.1)
1	0,9	0,2483	0,0125	0,1	0,0025	0,2483	1	1
2	0,9	0,2483	0,0375	0,9	0,0475	0,4966	2	1
3	0,9	0,2483	0,0625	0,3	0,0575	0,7448	3	1
4	0,025	0,0069	0,0875	0,4	0,085	0,7517	4	1
5	0,025	0,0069	0,1125	0,5	0,1125	0,7586	5	1
6	0,025	0,0069	0,1375	0,4	0,135	0,7655	6	1
7	0,025	0,0069	0,1625	0,6	0,165	0,7724	7	1
8	0,025	0,0069	0,1875	0,1	0,1775	0,7793	8	1
9	0,025	0,0069	0,2125	0,1	0,2025	0,7862	9	1
10	0,025	0,0069	0,2375	0,1	0,2275	0,7931	10	1
11	0,025	0,0069	0,2625	0,1	0,2525	0,8000	11	2
12	0,025	0,0069	0,2875	0,1	0,2775	0,8069	12	2
13	0,025	0,0069	0,3125	0,3	0,3075	0,8138	13	2
14	0,025	0,0069	0,3375	0,1	0,3275	0,8207	14	2
15	0,025	0,0069	0,3625	0,1	0,3525	0,8276	15	2
16	0,025	0,0069	0,3875	0,2	0,38	0,8345	16	2
17	0,025	0,0069	0,4125	0,1	0,4025	0,8414	17	2
18	0,025	0,0069	0,4375	0,1	0,4275	0,8483	18	2
19	0,025	0,0069	0,4625	0,1	0,4525	0,8552	19	2
20	0,025	0,0069	0,4875	0,1	0,4775	0,8621	20	2
21	0,025	0,0069	0,5125	0,1	0,5025	0,8690	21	3
22	0,025	0,0069	0,5375	0,9	0,5475	0,8759	22	3
23	0,025	0,0069	0,5625	0,3	0,5575	0,8828	23	3
24	0,025	0,0069	0,5875	0,4	0,585	0,8897	24	3
25	0,025	0,0069	0,6125	0,5	0,6125	0,8966	25	3
26	0,025	0,0069	0,6375	0,4	0,635	0,9034	26	3
27	0,025	0,0069	0,6625	0,6	0,665	0,9103	27	3
28	0,025	0,0069	0,6875	0,1	0,6775	0,9172	28	3
29	0,025	0,0069	0,7125	0,1	0,7025	0,9241	29	3
30	0,025	0,0069	0,7375	0,1	0,7275	0,9310	30	3
31	0,025	0,0069	0,7625	0,1	0,7525	0,9379	31	5
32	0,025	0,0069	0,7875	0,1	0,7775	0,9448	32	8
33	0,025	0,0069	0,8125	0,3	0,8075	0,9517	33	13
34	0,025	0,0069	0,8375	0,1	0,8275	0,9586	34	15
35	0,025	0,0069	0,8625	0,1	0,8525	0,9655	35	19
36	0,025	0,0069	0,8875	0,2	0,88	0,9724	36	23
37	0,025	0,0069	0,9125	0,1	0,9025	0,9793	37	26
38	0,025	0,0069	0,9375	0,1	0,9275	0,9862	38	30
39	0,025	0,0069	0,9625	0,1	0,9525	0,9931	39	34
40	0,025	0,0069	0,9875	0,1	0,9775	1,0000	40	37

$w_{soma}$	$N_{efetivo}$	$k$	$q$
3,625	5,35	0,025	0,0125

## CAPÍTULO 3

# Extração de características em Imagens

### 3.1 Introdução

Neste capítulo, apresentamos o algoritmo SIFT (*Scale-Invariant Feature Transform*) de extração de características de imagens (Fig.3.1(b)) que pode se usada para se conseguir uma correspondência confiável entre diferentes visões de uma cena ou objeto. As características são invariantes à escala e rotação da imagem, e parcialmente invariantes à mudança na iluminação e ponto de vista 3D da câmera. Este método gera um grande número de características por imagens. Além disso, elas são muito distintas, o que permite uma única característica ser corretamente correspondida com alta probabilidade em um grande banco de dados de características.



Figura 3.1: (a) Imagem original e (b) Imagem com características SIFT identificadas (setas brancas).

A quantidade de características geradas é importante para o reconhecimento de objetos, que requer que no mínimo 3 características (de cada objeto) sejam corretamente correspondidas para uma identificação confiável.

Embora as características SIFT possam ser usadas no reconhecimento de objetos, neste trabalho usamo-as como marcos no FastSLAM (capítulo 4).

As etapas principais do algoritmo SIFT são: detecção de extremos, localização exata dos pontos-chave <sup>1</sup>, definição da orientação e a descrição dos descritores dos pontos-chave.

## 3.2 Detecção de extremos no espaço-escala

Objetos no mundo real são significativos somente sob uma determinada escala. Podemos ver um cubo de açúcar perfeitamente sobre uma mesa, mas se formos procurá-lo em toda Via Láctea, então ele não existe (não é percebido). Esta natureza multi-escala de objetos é bastante comum na natureza. O espaço-escala tenta replicar este conceito em imagens digitais.

A etapa definida nesta seção consiste em buscar pontos-chave (*keypoints*) que sejam invariantes a mudanças de escala da imagem, possibilitando - desta forma - sua detecção com uma câmera próxima ou distante do objeto de interesse.

Vamos supor que temos a imagem de um objeto (Fig.3.1(a)). Se desejássemos identificar somente o objeto (e não seus detalhes), então, os detalhes do objeto deveriam ser intencionalmente removidos (Fig.3.2). Computacionalmente, a maneira de se fazer isto é com um borramento Gaussiano - também conhecido como suavização Gaussiana - que é o resultado do borramento de uma imagem por uma função Gaussiana com o objetivo de reduzir o ruído e detalhes na imagem .

Para se criar um espaço-escala, deve-se tomar a imagem original e gerar progressivamente imagens borradas [3] [4]. O espaço-escala de uma imagem é definida como uma função  $L(x, y, \sigma)$  gerada da convolução de uma Gaussiana variável em escala  $G(x, y, \sigma)$  com uma imagem de entrada,  $I(x, y)$ :

$$L(x, y, \sigma) = \left( \overbrace{\frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}}^{G(x,y,\sigma)} \right) * I(x, y) \quad (3.1)$$

---

<sup>1</sup>Os termos *ponto-chave* SIFT, *keypoint* SIFT e *feature* SIFT são outras denominações para uma característica SIFT.



Figura 3.2: Imagem da figura 3.1(a) com borrimento Gaussiano aplicado (Eq.3.1).

Detectar localizações que são invariantes à troca na escala da imagem ( $\sigma$ ) pode ser conseguido pela pesquisa de características estáveis em todas as possíveis escalas.

Para detectar eficientemente localizações de pontos-chave estáveis no espaço-escala, usa-se o extremo do espaço-escala na função de diferença-de-Gaussiana (DoG) – que pode ser calculada a partir da diferença de duas escalas próximas separadas por um fator multiplicativo constante  $k$  – convoluída com imagem ( $D(x, y, \sigma)$ ):

$$D(x, y, \sigma) = \overbrace{(G(x, y, k\sigma) - G(x, y, \sigma))}^{DoG} * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (3.2)$$

A escolha da função  $D(x, y, \sigma)$  se deve a dois fatores principais: o seu cálculo é eficiente e, além disso, a função DoG provê uma boa aproximação para o Laplaciano ( $\nabla^2$ ) da Gaussiana normalizado em escala<sup>2</sup>,  $\nabla^2 \sigma^2 G$  (Eq.3.3)

$$DoG = G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G \quad (3.3)$$

A operação Laplaciano de Gaussiana ( $\nabla^2 G$ ) toma uma imagem, borra-a e então calcula a derivada de segunda ordem sobre a mesma ( ou o laplaciano

<sup>2</sup>A a normalização do Laplaciano com um fator  $\sigma^2$  é exigido para uma verdadeira invariância na escala [4]. Note que  $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$  e que  $\sigma^2 G(x, y, \sigma) = \frac{1}{2\pi} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$

-  $\nabla^2$ ). O laplaciano localiza arestas e cantos em imagens, que são úteis para se encontrar pontos-chave. No entanto a segunda derivada é extremamente sensível ao ruído. O borramento suaviza o ruído e estabiliza a derivada de segunda ordem.

A referência [15] mostra que o máximo e o mínimo de  $\sigma^2 \nabla^2 G$  produzem características de imagens mais estáveis comparados a uma gama de outras possíveis funções de imagens.

A equação 3.3 demonstra que quando a função DoG tem escalas diferindo por um fator constante  $k$  ela já incorpora a normalização de escala  $\sigma^2$  exigida pelo Laplaciano invariante à escala. O fator  $(k - 1)$  é uma constante sobre todas as escalas e portanto não influencia a localização do extremo.

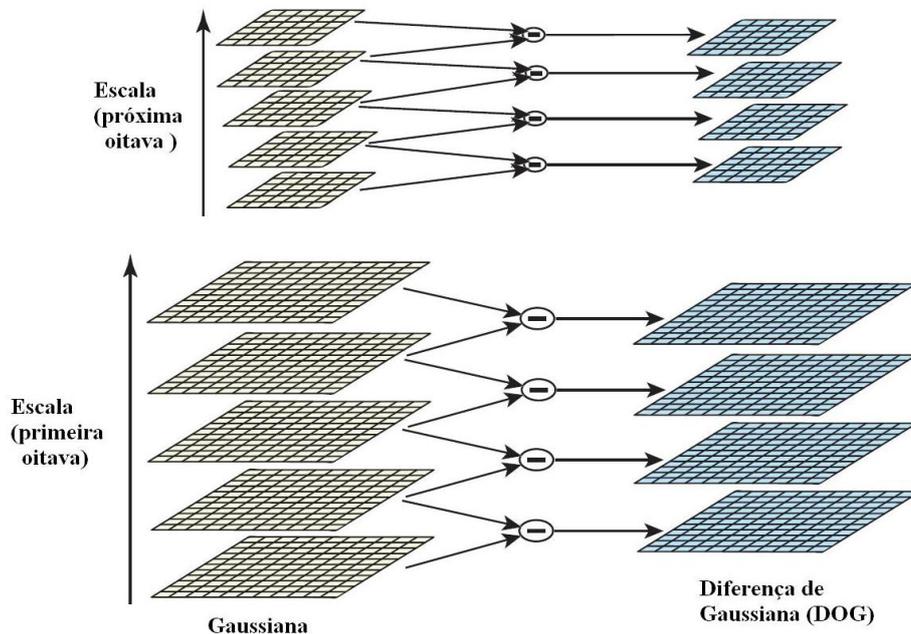


Figura 3.3: Representação do procedimento de obtenção das DoG para diversas oitavas de uma imagem [21].

Uma modo eficiente para a criação de DoG é esquematizado na figura 3.3, cujos passos são descritos a seguir.

1. A imagem inicial sofre convoluções incrementais com filtros Gaussianos para produzir imagens separadas por um fator de escala  $k$  no espaço-escala, representados na coluna esquerda.
2. A referência [21] considera que é necessário fazer convoluções da imagem até  $2\sigma$  para ser possível a construção de descritores invariantes à escala.

3. Imagens em escalas adjacentes são subtraídas para produzir as imagens DoG mostrada à direita.
4. Uma vez processada a oitava, a resolução da imagem é reduzida (*down-sample*) tomando-se cada segundo *pixel* da imagem no centro da oitava, gerando-se uma nova oitava (alteração da taxa de amostragem por um fator dois), e voltando-se ao passo 1.

A figura 3.4 a aplicação dos filtros Gaussianos e a geração de uma imagem DoG.



Figura 3.4: Duas imagens suavizadas em escalas diferindo por um fator  $k$  e a respectiva imagem DoG.

Os extremos são dados por valores de máximo e mínimo locais para cada  $D(x, y, \sigma)$ , que podem ser obtidos comparando-se a intensidade de cada ponto com as intensidades de seus oito vizinhos na sua escala, os nove vizinhos na escala superior, e os nove vizinhos na escala inferior (Fig.3.5). Na figura 3.5, o ponto marcado com **X** é comparado com seus vizinhos marcados com **O**. As 3 imagens DoG apresentadas na figura correspondem à diferença entre imagens adjacentes da pirâmide Gaussiana.

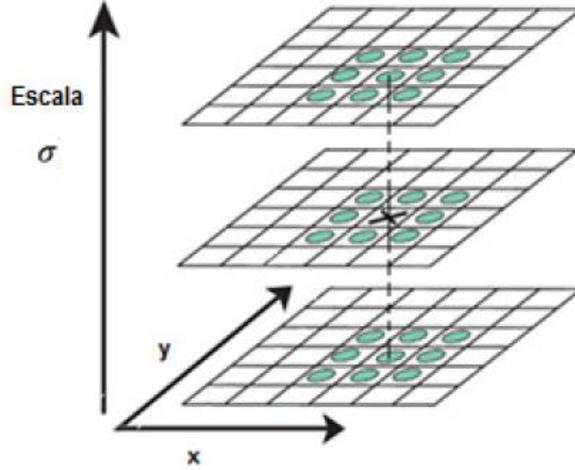


Figura 3.5: Detecção de extremos no espaço-escala [21].

O número de oitavas e escalas depende do tamanho da imagem original. Lowe[21] sugere que 4 oitavas e 5 níveis de borrimento (escala) são ideais para o algoritmo. A quantidade de borrimento  $k$  é importante. Lowe [21] utilizou em todos os seus experimentos o  $k = \sqrt{2}$  e  $\sigma = 1.6$ .

A próxima etapa é definir a localização dos ponto-chave e fazer o descarte de pontos instáveis.

### 3.3 Localização precisa dos pontos-chave

Todos os pontos detectados como extremos são candidatos a pontos-chave. Deseja-se calcular a localização exata destes pontos-chave.

O método consiste em ajustar uma função quadrática 3D do ponto de amostragem local de modo a determinar uma localização interpolada no máximo.

Isto é feito utilizando uma expansão de *Taylor* da função  $D(x, y, \sigma)$ , deslocada de modo que a origem desta expansão esteja localizada no ponto de amostragem [14]:

$$D(\bar{x}) = D + \frac{\partial D^T}{\partial \bar{x}} \bar{x} + \frac{1}{2} \bar{x}^T \frac{\partial^2 D}{\partial \bar{x}^2} \bar{x} \quad (3.4)$$

onde o valor de  $D$  e da sua primeira e segunda derivada são calculadas no ponto de amostragem e  $\bar{x} = (x, y, \sigma)^T$  representa o *offset* deste ponto.

A localização do extremo,  $\hat{x}$ , é determinada ao se calcular a derivada de  $D(\bar{x})$  em relação a  $\bar{x}$  e igualando o resultado a zero:

$$\hat{x} = -\frac{\partial^2 D^{-1}}{\partial \bar{x}^2} \frac{\partial D}{\partial \bar{x}} \quad (3.5)$$

O valor da função no extremo,  $D(\hat{x})$ , é útil para rejeição de extremos instáveis com baixo contraste, que seriam sensíveis a ruído. Substituindo-se a equação 3.5 na equação 3.4 obtém-se:

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial x} \hat{x} \quad (3.6)$$

É aconselhável que se trabalhe com  $|D(\hat{x})| \leq 0.03$  (assumindo que os valores dos *pixels* da imagem estejam normalizados em valores entre 0 e 1) [21].

Além do procedimento apresentado para se descartar pontos, Lowe [21] ainda aponta que a função DoG possui uma forte resposta nas arestas, mesmo que a localização ao longo da borda seja mal determinada, isto é, pontos em arestas poderiam ser escolhidos como pontos de interesse, o que não é desejável. Estes pontos podem ser detectados e eliminados, como será discutido a seguir.

### 3.3.1 Eliminação de respostas nas bordas

Para a estabilidade, não é suficiente rejeitar pontos-chave com baixo contraste. A função de DoG terá uma forte resposta ao longo das bordas, mesmo se um local junto a borda é pobremente determinado e, portanto, instável a pequenas quantidades de ruído.

Um pico mal definido na função DoG terá uma grande curvatura principal ao longo da borda, mas uma pequena curvatura na direção perpendicular. As curvaturas principais podem ser calculadas a partir de uma matriz Hessiana 2x2,  $H$ , computada na localização e escala dos pontos-chaves na função  $D$ :

$$H = \begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \quad (3.7)$$

onde  $D_{xy}$  é a derivada de  $D(x, y, \sigma)$  em relação a  $x$  e  $y$  e  $D_{xx}$  e  $D_{yy}$  são, respectivamente, a derivada segunda em relação a  $x$  e  $y$ .

As derivadas são estimadas através de diferenças de pontos vizinhos amostrados.

A matriz Hessiana representa assim um segunda derivada, permitindo mensurar as magnitudes das curvaturas de  $D$  a partir de seus autovalores.

Seja  $\alpha$  o autovalor com a maior magnitude e  $\beta$  o de menor magnitude. Pode-se calcular a soma dos autovalores a partir do traço de  $H$  e o produto pelo seu determinante:

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (3.8)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta \quad (3.9)$$

No caso em que o determinante for negativo, as curvaturas possuem sinais diferentes, e o ponto é descartado, não sendo considerado um extremo. Seja  $r$  a razão entre o autovalor de maior magnitude e o de menor, tal que  $\alpha = r\beta$ . Então,

$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r} \quad (3.10)$$

A equação 3.10 só depende da razão entre os autovalores, sendo independente dos seus valores individuais. O valor  $(r + 1)^2/r$  oferece uma medida de quanto os autovalores são distintos, ou seja, é mínimo quando são idênticos e cresce a medida que  $r$  aumenta. Desta forma, eliminam-se pontos (indesejáveis) próximos a extremidades descartando-se pontos abaixo de um determinado limiar:

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r} \quad (3.11)$$

A condição de desigualdade 3.11 pode ser verificada de forma muito eficiente. Lowe[21] propõe  $r = 10$ , assim eliminam-se pontos-chave que não são estáveis apesar de estarem próximos de extremidades.

### 3.4 Atribuição da Orientação dos Descritores

Atribuindo uma orientação consistente para cada ponto-chave baseado nas propriedades da imagem local, o descritor do ponto-chave (vetor com 128

posições - seção 3.5) pode ser representado em relação a esta orientação e portanto conseguir invariância a rotação da imagem.

Calcula-se para cada amostragem da imagem na escala,  $L(x, y, \sigma)$ , a magnitude  $m(x, y)$ , e a orientação,  $\theta(x, y)$ , do gradiente usando as diferenças de *pixels*:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.12)$$

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \quad (3.13)$$

Monta-se um histograma das orientações para *pixels* em uma região vizinha ao redor do ponto-chave. O histograma possui 36 regiões, cobrindo todas as orientações possíveis (0 a  $2\pi$ )(Fig.3.6).

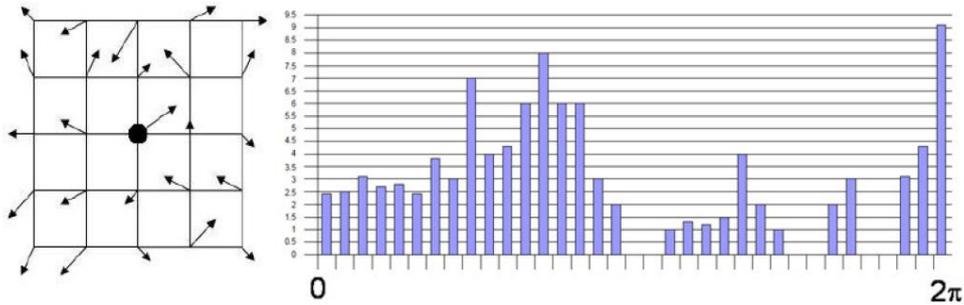


Figura 3.6: Histograma de orientações de um ponto-chave.

Cada ponto na vizinhança do ponto-chave é adicionado ao histograma com um valor de peso determinado (Eq.3.15). O primeiro peso é o valor da magnitude  $m(x, y)$  de cada ponto adicionado. O segundo peso é dado por uma janela Gaussiana circular com  $\sigma^*$  igual a 1.5 vezes maior que a escala do ponto-chave. Esta janela é definida pela equação Gaussiana

$$g(\Delta x, \Delta y, \sigma^*) = \frac{1}{2\pi(\sigma^*)^2} e^{-(\Delta x^2 + \Delta y^2)/2(\sigma^*)^2} \quad (3.14)$$

onde  $\Delta x$  e  $\Delta y$  são as distâncias entre cada ponto verificado e o ponto-chave.

O valor dos pesos calculados para cada ponto na vizinhança em  $(x, y)$  é atualizado na expressão:

$$h_{\theta}^* = h_{\theta} + \alpha m(x, y)g(\Delta x, \Delta y, \sigma^*) \quad (3.15)$$

com  $\alpha = \frac{d}{i}$  ( $d < i$ ) ou  $\alpha = 0$  ( $d > i$ ) onde  $h_{\theta}^*$  é a atualização de  $h_{\theta}$ ,  $d$  é a distância em graus entre a orientação do ponto e o  $\theta$  discretizado e  $i$  é o intervalo em graus entre os  $\theta$  discretizados.

Picos no histograma de orientações (Fig.3.6) correspondem a direções dominantes dos gradientes locais. Além do máximo, são considerados também os picos que correspondem a pelo menos 80% do valor deste máximo. Portanto, um mesmo ponto chave poderá ter mais de uma orientação associada.

O pico deste histograma é utilizado para definir a sua orientação. No caso de múltiplos picos de elevada amplitude, o ponto-chave receberá múltiplas orientações, tornando-se ainda mais estável para futura identificação. No final, uma parábola é usada para interpolar os três valores do histograma mais próximos ao pico, de forma a se obter uma melhor exatidão na sua posição.

Cada ponto-chave tem agora quatro dimensões: posição  $x$  e  $y$ , magnitude e orientação.

### 3.5 Construção do Descritor Local

Nesta seção, a cada ponto-chave é atribuído um descritor invariante à iluminação e ponto de vista 3D, tornando-o bem distinguível. É importante lembrar que os procedimentos a seguir são feitos com valores normalizados em relação à orientação e magnitude definidos na seção anterior para cada ponto-chave.

Para que os descritores tenham invariância à rotação, as orientações dos gradientes destes pontos são giradas de um ângulo correspondente à orientação do ponto-chave definido na seção anterior.

O descritor do ponto-chave é então criado calculando-se as magnitudes e orientações dos gradientes que são amostradas ao redor da localização do ponto-chave. Este procedimento é exibido na figura 3.8, onde os gradientes são representados pelas pequenas setas em cada amostra da localização. São definidas  $n \times n$  regiões de amostragem cada uma com  $k \times k$  pixels ao redor da localização do ponto-chave.

Uma função Gaussiana é utilizada para dar peso à magnitude do gradiente em cada ponto na vizinhança do ponto-chave, com uma janela de suavização Gaussiana de escala  $\sigma$  igual à metade da largura da janela do

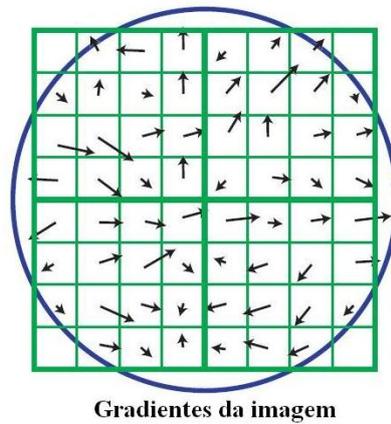


Figura 3.7: Mapa de gradientes para  $n = 2$  regiões com  $k = 4$  *pixels* [21].

descriptor. Esse Gaussiano evita mudanças súbitas do descriptor a pequenas mudanças na posição da janela, e também reduz a ênfase nos gradientes longe do centro do descriptor, que são mais afetados por erros.

Uma vez efetuada a suavização dos gradientes, o descriptor consiste de um vetor contendo os valores do histograma. No exemplo da 3.8, o histograma tem 8 valores de orientação, cada um criado ao longo de uma janela de apoio de  $4 \times 4$  *pixels*. Com uma janela de apoio de  $16 \times 16$  *pixels*, o vetor característico resultante tem 128 elementos .

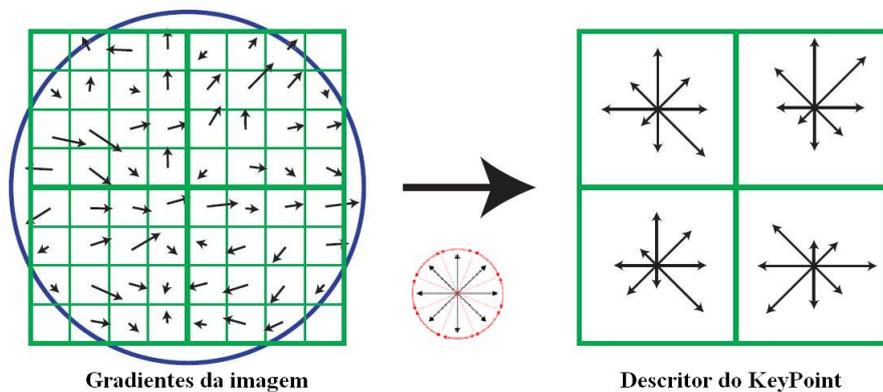


Figura 3.8: Construção do descriptor para um ponto-chave de  $2 \times 2$  com 48 elementos [21].

No entanto, duas imagens de um mesmo objeto podem possuir variações de luminosidade que modifiquem sensivelmente os descritores obtidos. Assim, para que o descriptor tenha invariância à iluminação, este é normalizado.

Os descritores são invariantes a mudanças homogêneas de brilho da imagem, uma vez que esta variação representa uma adição a todos os *pixels*

da imagem de uma constante, e os descritores são calculados por diferenças de *pixels*. Quanto a mudanças homogêneas de contraste, representadas pela multiplicação de todos os *pixels* por uma constante, elas são corrigidas com a normalização dos descritores.

Variações não-lineares, causadas por saturação das câmeras ou por efeito de iluminação de superfícies tridimensionais em diferentes orientações, podem influenciar grandemente as magnitudes dos descritores, mas têm pouca influência em sua orientação. Reduz-se este efeito impondo um valor máximo às magnitudes. Após a normalização, todos os valores acima de um determinado limiar são ajustados para este limiar. Isto é feito para que direções com magnitude muito grande não dominem a representação do descritor. Lowe[21] sugere usar um limiar 0.2. Isto significa que a correspondência para as grandes magnitudes dos gradientes não é tão importante se comparada com a distribuição das orientações.

Para cada imagem, são construídos diversos descritores, cada um referente a um ponto-chave. Tem-se como resultado, portanto, um conjunto de descritores robustos que podem ser usados para fazer a correspondência da imagem em outra imagem, como será detalhado na seção 3.5.3.

### 3.5.1 Sensibilidade a trocas afins

A sensibilidade do descritor a trocas afins é examinada na figura 3.9. Pode ser verificado que a repetibilidade reduziu com o incremento na distorção afim, mas que a precisão na correspondência final permanece acima de 50% para uma mudança de até  $50^\circ$  no ponto de visada.

### 3.5.2 Correspondência para grandes bases de dados

Uma questão importante para medição da distinguibilidade de características é como a confiabilidade da correspondência varia em função do número de características no banco de dados. A figura 3.10 mostra que a confiabilidade da correspondência varia em função do tamanho do banco de dados. Neste caso, esta figura foi gerada usando banco de dados de 112 imagens, com uma transformação afim de  $30^\circ$  e 2% de ruído na imagem em adição a usual mudança aleatória na rotação e escala da imagem.

A linha sólida é o percentual de pontos-chave em que foi identificado uma correta correspondência na imagem transformada, tal que somente estes pontos têm a chance de terem descritores correspondentes no banco de dados.

A linha tracejada mostra a porção das características da imagem em que o vizinho mais próximo no banco de dados teve uma correspondência correta, como uma função do tamanho do banco de dados mostrado em

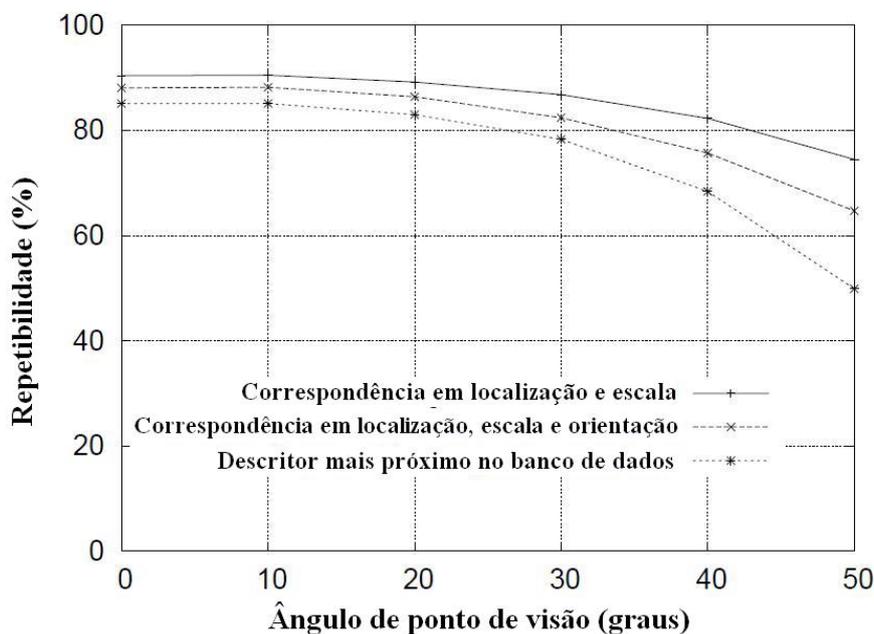


Figura 3.9: Este gráfico mostra a estabilidade da detecção para localização e orientação de pontos-chave e a correspondência final para um banco de dados como uma função da distorção afim [21].

escala logarítmica.

### 3.5.3 Encontrando pontos em comum (*Matching*)

A idéia de *matching* é extrair pontos-chave de duas imagens, e procurar os pontos correspondentes em cada uma, como exemplificado na Figura 3.11. A comparação de pontos é baseada na similitude dos descritores correspondentes.

Quando se trabalha com SIFT, pontos de interesse são detectados pelo método e representados por descritores. Os descritores são vetores que podem ser comparados, por exemplo, utilizando a distância Euclidiana. Normalmente, os melhores candidatos à correspondência são pontos próximos, de maneira que o melhor candidato é o ponto que apresenta a menor distância Euclidiana.

Lowe [21] utilizou uma modificação do algoritmo *k-d tree* chamado de método de *Best-Bin-First (BBF)*, que pode identificar os vizinhos mais próximos com elevada probabilidade, utilizando apenas uma quantidade limitada de esforço computacional.

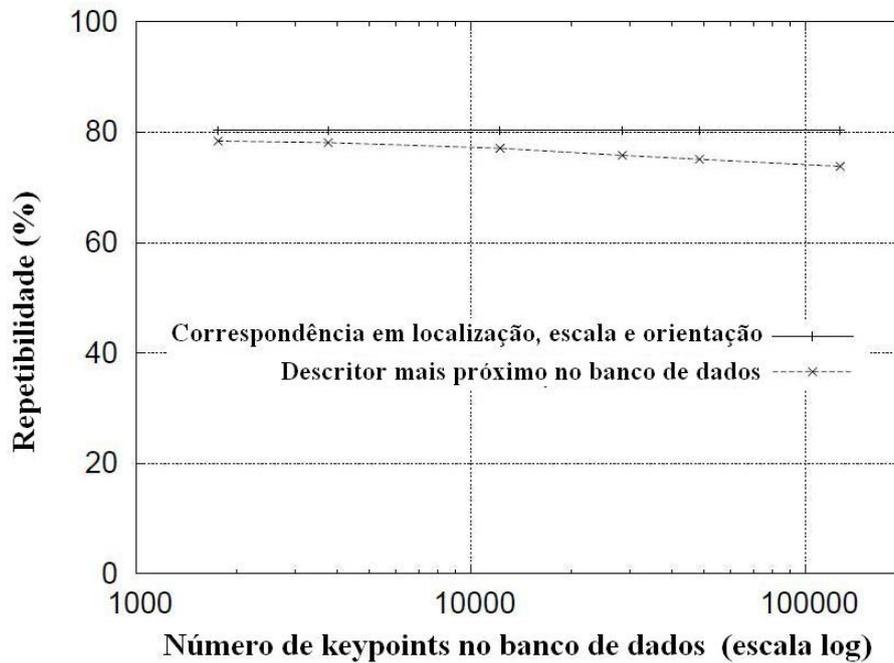


Figura 3.10: A linha pontilhada exibe o percentual de pontos-chave corretamente correspondido em função do tamanho do banco (usando escala logarítmica). A linha sólida mostra o percentual de pontos-chave que recebe uma atribuição correta em orientação, escala e orientação [21].

### 3.6 O SIFT em relação a outras abordagens

Em [26], o SIFT foi comparado com as abordagens GLOH (*Gradient Location and Orientation Histogram*) [23], PCA-SIFT (*Principal Components Analysis - Scale invariant feature transform*) [22] e o SURF (*Speeded Up Robust Feature*) [26] em relação aos seguintes parâmetros: repetibilidade (encontrar o mesmo ponto de interesse sobre diferentes visões - até 50°), distinguibilidade do descritor e rapidez.

SIFT <i>versus</i>	GLOH	PCA_SIFT	SURF
Repetibilidade	similar	menor	menor
Distinguibilidade do descritor	menor	maior	menor
Rapidez	maior	menor	menor

Tabela 3.1: SIFT *versus* Outras abordagens.

Portanto, embora tenhamos utilizado o SIFT, o SURF tem uma performance melhor e é ao mesmo tempo mais rápido.

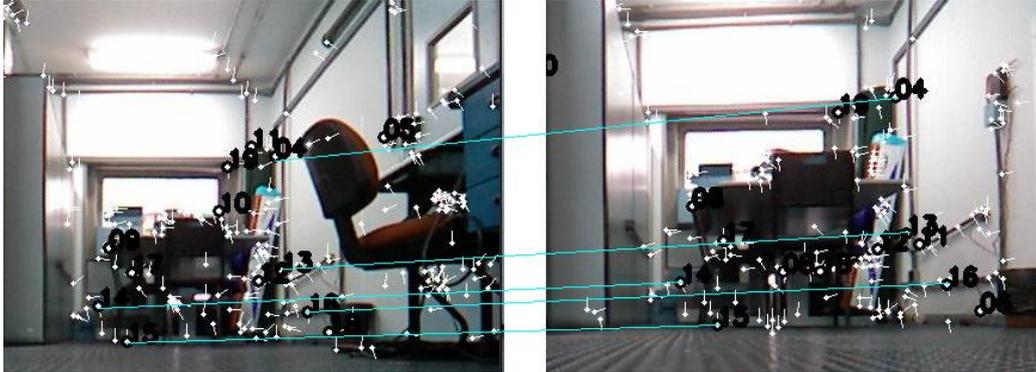


Figura 3.11: Processo de correspondência entre duas imagens com o SIFT.

### 3.7 Conclusão

Neste capítulo, apresentamos uma visão geral do algoritmo SIFT que fornece descritores estáveis de características (*keypoints*) em imagens. No capítulo 4 trataremos da abordagem FastSLAM Visual em que os marcos visuais 3D são características (*keypoints*) SIFT.

## 4.1 Introdução

Uma abordagem típica de SLAM utiliza sensor a laser na extração de características de marcos observados no ambiente (suas coordenadas  $2D$ ) como foi mostrado no capítulo 2. No entanto, o interesse pelo uso de câmeras tem crescido e muitos autores têm concentrado sua atenção na construção de mapas  $3D$  de marcos detectados no ambiente usando informações obtidas através de câmeras. Esta abordagem é usualmente denotada SLAM Visual. As razões para o crescente interesse no SLAM Visual são o custo dos sensores visuais (normalmente mais baratos que sensores a laser), o provimento de informação  $3D$  do ambiente e a possibilidade de integração de outras técnicas, tais como o reconhecimento de objetos.

Deste ponto em diante, consideramos os marcos como características (*keypoints*) SIFT (capítulo 3) de imagens obtidas com o sensor visual *kinect* [52] embarcado no veículo (Fig.4.1). Este sensor visual permite que se obtenham as coordenadas  $3D$  dos respectivos marcos em relação ao veículo (seção 5.3).

Como foi visto na seção 2.1, o FastSLAM [18] decompõe o problema do SLAM em um problema de localização do veículo e uma coleção de problemas de estimação de marcos que estão condicionados à estimativa da pose do veículo.

Seja  $u_t$  uma ação de controle responsável pela troca de estado de um veículo no instante  $t$ . Em robótica, a pose  $\chi_t$  do veículo e a observação  $z_t$  de

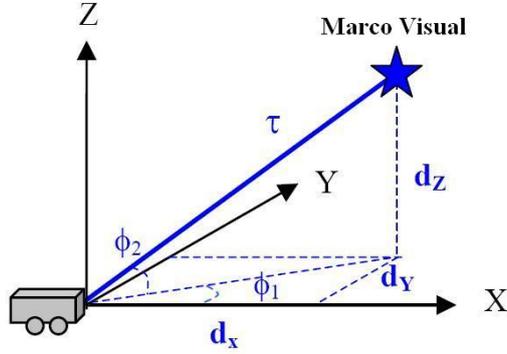


Figura 4.1: Na observação de um marco visual 3D (característica (*keypoint*) SIFT)  $z_t = \{d_x, d_y, d_z\}$  e  $d_t$  (descriptor da característica (*keypoint*)) devem ser obtidos.

um marco  $\theta_j$  são modelados por leis probabilísticas:

$$p(\chi_t | \chi_{t-1}, u_t) \text{ e } p(z_t | \chi_t, \theta_j) \quad (4.1)$$

com valores amostrados por funções usualmente não lineares,  $h$  e  $g$ , em seus argumentos, contaminadas com ruídos brancos Gaussianos de média nula e covariância  $Q_t$  e  $R_t$ , respectivamente:

$$\chi_t = h(u_t, \chi_{t-1}) + \overbrace{N(0, Q_t)}^{\delta_t} \text{ e } z_t = g(\chi_t, \theta_j) + \overbrace{N(0, R_t)}^{\varepsilon_t} \quad (4.2)$$

O FastSLAM estima a distribuição de probabilidade posterior do caminho do veículo  $\chi^t = \{\chi_1, \dots, \chi_t\}$  e do mapa  $\Theta$  (Eq.4.3), considerando as observações  $z^t = \{z_1, \dots, z_t\}$ , as ações de controle  $u^t = \{u_1, \dots, u_t\}$  e as associações  $a^t = \{a_1, \dots, a_t\}$  entre as características dos marcos<sup>1</sup> que foram observados e as dos marcos no mapa:

$$p(\chi^t, \Theta | z^t, u^t, a^t) \quad (4.3)$$

É demonstrado que se o caminho  $\chi^t$  é conhecido, então a posição do marco  $\theta_i$  em  $\Theta$  são condicionalmente independentes, o que permite fatorar o problema da estimação da distribuição de probabilidade posterior de  $\chi^t$  e  $\Theta$

<sup>1</sup>As características do marco visual (característica (*keypoint*) SIFT) no mapa de características são suas coordenadas 3D, bem como o seu descriptor.

como um produto de termos simples (Eq.4.4), como descrito na representação (parcial) em rede de Bayes da figura 2.3:

$$p(\chi^t, \Theta | z^t, u^t, a^t) = \underbrace{p(\chi^t | z^t, u^t, a^t)}_{\text{Caminho}} \prod_{n=1}^N \underbrace{p(\theta_n | \chi^t, z^t, u^t, a^t)}_{\text{Marco}} \quad (4.4)$$

A distribuição de probabilidade posterior de  $\chi^t$  é estimada usando um filtro de partículas [25] e a distribuição de probabilidade posterior sobre os  $N$  marcos  $\theta_i$  de cada partícula são estimados por meio de  $N$  filtros de Kalman Estendidos [25] condicionados ao caminho  $\chi^t$ .

O filtro de partículas representa a distribuição usando um conjunto  $S_t = \{S_t^{[1]}, \dots, S_t^{[M]}\}$  de partículas:

$$S_t^{[m]} = \left[ \chi_t^{[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, \dots, \psi_{n,t}^{[m]}, \Sigma_{n,t}^{[m]}, d_n^{[m]}, \dots, w_t^{[m]}}_{\Theta^{[m]}} \right] \quad (4.5)$$

onde  $\psi_{n,t}^{[m]}$  e  $\Sigma_{n,t}^{[m]}$  são a média e a covariância das coordenadas 3D de  $\theta_n$  (marco visual) condicionadas a pose  $\chi_t^{[m]}$ ;  $d_n^{[m]}$  é o descritor da característica (*keypoint*) SIFT de  $\theta_n$  e  $w_t^{[m]}$  é o peso da partícula.

Quando o veículo realiza uma observação ele deve atualizá-la no seu mapa. Dado que os sensores estão sujeitos a erros, as informações incorporadas no mapa podem ter uma certa quantidade de incerteza que pode conduzir a erros de associação de dados e de atualização.

O *erro de atualização* pode ser minimizado através de sucessivas observações e, portanto, para melhorar a precisão na pose do veículo, é necessário *fechar o loop*, ou seja, reobservar um marco previamente observado. Este erro ocorre também devido à linearização executada pelo EKF.

O *erro na associação de dados*, causado pela associação (correspondência) errônea de uma característica presente no mapa a uma característica de um marco observado no ambiente, pode ser reduzida com a utilização de um conjunto de características mais robusto, como é o caso das características SIFT, cujo descritor, um vetor de dimensão 128, é invariante à escala da imagem e sua rotação e parcialmente invariante à mudança na iluminação e ao ponto de vista 3D da câmera.

## 4.2 FastSLAM para marcos 3D

O algoritmo FastSLAM usado neste trabalho é o Algoritmo 4.1. Nós consideramos uma observação  $v_t = \{z_t, d_t\}$  onde  $d_t$  é o descritor SIFT de um

marco visual e  $z_t = [d_x, d_y, d_z]^T$  é a distância entre o marco e o veículo obtido através de um sensor visual (Fig.4.1).

### Explicação do Algoritmo 4.1

Para cada partícula  $S_{t-1}^{[m]}$  de  $S_{t-1}$ , é amostrado o modelo de movimento do veículo (Passo 2.(b), Eq.4.1 e Eq.4.2). Por outro lado, para cada marco  $\theta_n^{[m]}$  no mapa  $\Theta^{[m]}$ , é computada a distância de Mahalanobis  $E_n^{[m]}$  (distância Euclidiana ao quadrado) entre o descritor SIFT  $d_t$  do marco observado e o descritor  $d_n^{[m]}$  de  $\theta_n^{[m]}$  (passo 2.(c).a), bem como a qualidade da associação de  $\theta_n^{[m]}$  com  $z_t$  (Passo 2.(c).e).

Na estratégia de associação de dados adotada, escolhe-se um conjunto  $j$  de marcos com probabilidade de associação maior do que  $P_0$  (passo 2.(e)) e seleciona-se, neste conjunto, aquele que minimize  $E_j^{[m]}$  (passo 2.(f)). Se  $E_{a_t}^{[m]}$  é menor ou igual que um certo limite  $E_0$  (passo 2.(g)), a associação de dados é considerada correta e a estimativa das coordenadas do marco  $\theta_{a_t}^{[m]}$  são atualizadas com equações do EKF (passos 2.(g).a até 2.(g).d). Caso contrário, um novo marco é criado no mapa (passos 2.(h).a até 2.(h).c) [39].

O peso da partícula  $w_t$  corresponde à qualidade da associação  $p_{a_t,t}^{[m]}$  do marco  $\theta_{a_t}^{[m]}$  que é associado com a observação  $z_t$  (passo 2.(g).e).

A melhor partícula, que corresponde à estimativa de localização obtida, é a que possui maior peso (passo 4).

O algoritmo descrito considera apenas uma observação a cada passo. Caso haja mais observações, o peso da partícula (passo 2.(g).e) é calculado como o produto normalizado das respectivas probabilidades de associação.

Na reamostragem (passo 5), partículas com maior peso  $w_t^{[i]}$  são replicadas.

O algoritmo FastSLAM Visual, no caso de um veículo, requer tempo  $MN$ . Isto ocorre porque  $M$  partículas no filtro necessitam ser processadas, enquanto que, para cada partícula, a associação de dados necessita iterar sobre  $N$  marcos no mapa. Entretanto, se cada partícula for armazenada em uma estrutura *kd-tree*<sup>2</sup> de dimensão do descritor SIFT (128), a pesquisa em cada estrutura por uma lista de marcos (vizinhos mais próximos a uma distância  $E_0$ ) custa  $\log N$ , o que acelera a associação de dados, implicando em um tempo  $M \log N$ . Nesta lista, escolhe-se aqueles que possuem probabilidade

<sup>2</sup>*K-d tree*[57] é uma estrutura de particionamento de espaço de dados para organização de pontos em um espaço k-dimensional. Estruturas *k-d tree* são úteis em pesquisas envolvendo uma chave de pesquisa multidimensional (por exemplo, pesquisas de intervalos e pesquisa de vizinhos mais próximos).

FastSLAM Visual 3D ( $S_{t-1}, u_t, R, z_t$ )

1.  $S_t = S_{aux} = \emptyset$
2. Para  $m = 1$  até  $M$  fazer *{Loop: todas partículas}*
  - (a) Recupere  $[\chi_{t-1}^{[m]}, N_{t-1}^{[m]}, \psi_{1,t-1}^{[m]}, \Sigma_{1,t-1}^{[m]}, d_1^{[m]}, \dots, \psi_{n,t-1}^{[m]}, \Sigma_{n,t-1}^{[m]}, d_n^{[m]}, \dots, w_{t-1}^{[m]}]$  de  $S_{t-1}$
  - (b) Amostre  $\chi_t^{[m]} \sim p(\chi_t | \chi_{t-1}^{[m]}, u_t)$  *{Amostragem de uma nova pose (ver equação 4.2)}*
  - (c) Para  $n = 1$  até  $N_{t-1}^{[m]}$  fazer *{Loop sobre potenciais associações de dados}*
    - a.  $E_n^{[m]} = \text{Mahalanobis}(d_n^{[m]}, d_t) = (d_n^{[m]} - d_t)(d_n^{[m]} - d_t)^T$
    - b.  $\hat{z}_{n,t} = g(\chi_t^{[m]}, \psi_{n,t-1}^{[m]})$  (Eq.2.27)
    - c.  $G_{\theta,n} = \nabla_{\theta_n} g(\theta_n, \chi_t) |_{\theta_n = \psi_{n,t-1}^{[m]}, \chi_t = \chi_t^{[m]}}$  (Eq.2.28)
    - d.  $Z_{n,t} = G_{\theta,n} \Sigma_{n,t-1}^{[m]} G_{\theta,n}^T + R$  (Eq.2.29)
    - e.  $p_{n,t}^{[m]} = |2\pi Z_{n,t}|^{-\frac{1}{2}} \exp\{\frac{1}{2}(z_t - \hat{z}_{n,t})^T Z_{n,t}^{-1} (z_t - \hat{z}_{n,t})\}$  (Eq.2.43)
  - (d) Fim Para
  - (e)  $j = \{n\}$  onde  $p_{n,t}^{[m]} \geq P_0$
  - (f)  $a_t = \arg\min_j E_j^{[m]}$  *{Associação de dados}*
  - (g) Se  $E_{a_t}^{[m]} \leq E_0$ 
    - a.  $N_t^{[m]} = N_{t-1}^{[m]} + 1$  *{É um Marco conhecido: atualização}*
    - b.  $K_{a_t,t}^{[m]} = \Sigma_{a_t,t-1} G_{\theta,a_t}^T Z_{a_t,t}^{-1}$  (Eq.2.30)
    - c.  $\psi_{a_t,t}^{[m]} = \psi_{a_t,t-1}^{[m]} + K_{a_t,t} (z_t - \hat{z}_{a_t,t})$  (Eq.2.31)
    - d.  $\Sigma_{a_t,t}^{[m]} = (I - K_{a_t,t} G_{\theta,a_t}) \Sigma_{a_t,t-1}^{[m]}$  (Eq.2.32)
    - e.  $w_t^{[m]} = p_{a_t,t}^{[m]}$
  - (h) Senão
    - a.  $N_t^{[m]} = N_{t-1}^{[m]} + 1$  *{É Novo Marco: inicialização}*
    - b.  $\psi_{a_t,t}^{[m]} = g^{-1}(\chi_t^{[m]}, \hat{z}_{a_t,t})$  (Eq.2.48)
    - c.  $\Sigma_{a_t,t}^{[m]} = (G_{\theta,a_t}^T R^{-1} G_{\theta,a_t})^{-1} = G_{\theta,a_t}^{-1} R G_{\theta,a_t}^{-T}$  (Eq.2.49)
  - (i) Fim Se
  - (j) Fim Para
  - (k) Adicionar  $[\chi_t^{[m]}, N_t^{[m]}, \psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, \dots, \psi_{n,t}^{[m]}, \Sigma_{n,t}^{[m]}, d_n^{[m]}, \dots, w_t^{[m]}]$  a  $S_{aux}$
3. Fim Para
4.  $id = \text{melhor-partícula}(S_{aux})$
5.  $S_t = \text{Reamostragem}(S_{aux})$
6. Retornar  $S_t$

de associação maior que  $P_0$ . Neste conjunto é selecionado o marco que possui distância de *Mahanalobis* mínima (que é menor que ou igual a  $E_0$ ).

#### Detalhes do Algoritmo 4.1

O modelo de observação utilizado neste caso é ilustrado pela figura 4.1 e é similar ao modelo *distância e orientação* de um sensor laser (Fig.2.2). Como os valores  $\tau$ ,  $\phi_1$  e  $\phi_2$  usados no FastSLAM Visual 3D são derivados de  $z_t = \{d_x, d_y, d_z\}$  obtidos com o sensor *kinect* (seção 5.3), na amostragem das variáveis  $\phi_1$  e  $\phi_2$  (Tab.4.1),  $\bar{\phi}_i$  deve ser pequeno a fim de que os valores das medidas  $d_x$ ,  $d_y$  e  $d_z$  dependam somente de  $\tau$ , similarmente ao que ocorre quando se utiliza o *kinect*.

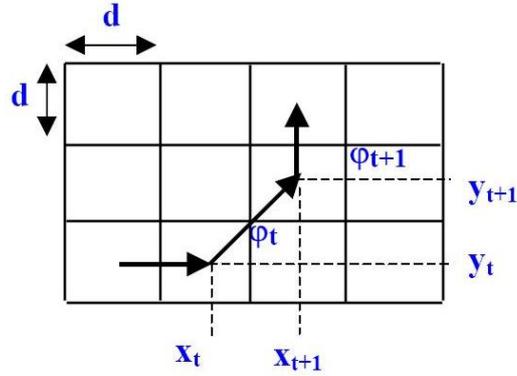


Figura 4.2: Modelo de movimento usado

Tabela 4.1: Amostragem do modelo de movimento (Fig.4.2) e observação (Fig.4.1).  $N(0, \sigma)$  é uma distribuição normal com média zero e desvio padrão  $\sigma$ .

Variável	Erro Médio
$x_{t+1} = x_t + \Delta x_t + \bar{d}N(0, \sigma)$	$\bar{d}$
$y_{t+1} = y_t + \Delta y_t + \bar{d}N(0, \sigma)$	$\bar{d}$
$\varphi_{t+1} = \varphi_t + \Delta \varphi_t + \bar{g}N(0, \sigma)$	$\bar{g}$
$\tau = \tau + \bar{\tau}N(0, \sigma)$	$\bar{\tau}$
$\phi_i = \phi_i + \bar{\phi}_iN(0, \sigma) \quad i = \{1, 2\}$	$\bar{\phi}_i$

Nos passos 2(c)b e 2(g)c, a observação predita  $\hat{z}_n = \hat{g}(\chi, \psi_n) = [\hat{\tau}, \hat{\phi}_1, \hat{\phi}_2]^T$  - que é função da pose estimada do veículo  $\chi$  e das coordenadas estimadas do marco  $\psi_n (\psi_{n,x}, \psi_{n,y}, \psi_{n,z})$  - é calculada de acordo com as equações 4.6, 4.7 e 4.8.

$$\hat{\tau} = \sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2 + (\psi_{n,z})^2} \quad (4.6)$$

$$\hat{\phi}_1 = \tan^{-1} \left[ \frac{\psi_{n,y} - y}{\psi_{n,x} - x} \right] - \alpha \quad (4.7)$$

$$\hat{\phi}_2 = \tan^{-1} \left[ \frac{\psi_{n,z}}{\sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2}} \right] \quad (4.8)$$

O Jacobiano  $G_{\theta_n}$  nos passos 2(c)c e 2(g)d é calculado como descrito na equação 4.9:

$$G_{\theta_n} = \begin{bmatrix} \frac{\partial \hat{\tau}}{\partial \psi_{n,x}} & \frac{\partial \hat{\tau}}{\partial \psi_{n,y}} & \frac{\partial \hat{\tau}}{\partial \psi_{n,z}} \\ \frac{\partial \hat{\phi}_1}{\partial \psi_{n,x}} & \frac{\partial \hat{\phi}_1}{\partial \psi_{n,y}} & \frac{\partial \hat{\phi}_1}{\partial \psi_{n,z}} \\ \frac{\partial \hat{\phi}_2}{\partial \psi_{n,x}} & \frac{\partial \hat{\phi}_2}{\partial \psi_{n,y}} & \frac{\partial \hat{\phi}_2}{\partial \psi_{n,z}} \end{bmatrix} = \begin{bmatrix} \frac{\psi_{n,x} - x}{\hat{\tau}} & \frac{\psi_{n,y} - y}{\hat{\tau}} & \frac{\psi_{n,z}}{\hat{\tau}} \\ -\frac{\psi_{n,y} - y}{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2} & \frac{\psi_{n,x} - x}{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2} & 0 \\ -\frac{\psi_{n,z}(\psi_{n,x} - x)}{\hat{\tau}^2 \sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2}} & -\frac{\psi_{n,z}(\psi_{n,y} - y)}{\hat{\tau}^2 \sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2}} & \frac{\hat{d}_z}{(\sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2})^3} \end{bmatrix} \quad (4.9)$$

Fazendo  $\hat{d}_x = \psi_{n,x} - x$ ,  $\hat{d}_y = \psi_{n,y} - y$ ,  $\hat{d}_z = \psi_{n,z}$  e  $\hat{d} = \sqrt{(\psi_{n,x} - x)^2 + (\psi_{n,y} - y)^2}$  temos:

$$G_{\theta_n} = \begin{bmatrix} \frac{\hat{d}_x}{\hat{\tau}} & \frac{\hat{d}_y}{\hat{\tau}} & \frac{\hat{d}_z}{\hat{\tau}} \\ -\frac{\hat{d}_y}{(\hat{d})^2} & \frac{\hat{d}_x}{(\hat{d})^2} & 0 \\ -\frac{(\hat{d}_z)(\hat{d}_x)(\hat{d})}{\hat{\tau}^2} & -\frac{(\hat{d}_z)(\hat{d}_y)(\hat{d})}{\hat{\tau}^2} & \frac{(\hat{d})^3}{\hat{\tau}^2} \end{bmatrix} \quad (4.10)$$

Explicitando o cálculo de cada um dos termos da equação 4.10 temos <sup>3</sup>:

$$\begin{aligned} -\frac{\partial \hat{\tau}}{\partial \psi_{n,x}} &= \left(\frac{1}{2}\right) \left(\frac{1}{\hat{\tau}}\right) (2)(\psi_{n,x} - x) = \frac{\hat{d}_x}{\hat{\tau}} \\ -\frac{\partial \hat{\tau}}{\partial \psi_{n,y}} &= \left(\frac{1}{2}\right) \left(\frac{1}{\hat{\tau}}\right) (2)(\psi_{n,y} - y) = \frac{\hat{d}_y}{\hat{\tau}} \\ -\frac{\partial \hat{\tau}}{\partial \psi_{n,z}} &= \left(\frac{1}{2}\right) \left(\frac{1}{\hat{\tau}}\right) (2)(\psi_{n,z}) = \frac{\hat{d}_z}{\hat{\tau}} \end{aligned}$$

3

- $y = u^n \rightarrow y' = n(u^{n-1})u'$
- $y = \frac{u}{v} \rightarrow y' = \frac{(vu' + uv')}{v^2}$
- $y = \tan^{-1} u \rightarrow u' = \frac{u'}{(1+u^2)}$

$$\begin{aligned}
-\frac{\partial \hat{\phi}_1}{\partial \psi_{n,x}} &= \frac{\frac{\partial \left( \frac{\psi_{n,y-y}}{\psi_{n,x-x}} \right)}{\partial \psi_{n,x}}}{1 + \left( \frac{\psi_{n,y-y}}{\psi_{n,x-x}} \right)^2} = \frac{(\psi_{n,x-x})(0) - (\psi_{n,y-y})(1)}{(\psi_{n,x-x})^2} = -\frac{\hat{d}_y}{(\hat{d})^2} \\
-\frac{\partial \hat{\phi}_1}{\partial \psi_{n,y}} &= \frac{\frac{\partial \left( \frac{\psi_{n,y-y}}{\psi_{n,x-x}} \right)}{\partial \psi_{n,y}}}{1 + \left( \frac{\psi_{n,y-y}}{\psi_{n,x-x}} \right)^2} = \frac{(\psi_{n,x-x})(1) - (\psi_{n,y-y})(0)}{(\psi_{n,x-x})^2} = \frac{\hat{d}_x}{(\hat{d})^2} \\
-\frac{\partial \hat{\phi}_1}{\partial \psi_{n,z}} &= 0 \\
-\frac{\partial \hat{\phi}_2}{\partial \psi_{n,x}} &= \frac{\frac{\partial \left( \frac{\psi_{n,z}}{\sqrt{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} \right)}{\partial \psi_{n,x}}}{1 + \frac{(\psi_{n,z})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \frac{-\psi_{n,z} \left( \frac{1}{2} \right) \left( (\psi_{n,x-x})^2 + (\psi_{n,y-y})^2 \right)^{-\frac{1}{2}} (2)(\psi_{n,x-x})}{\frac{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2 + (\psi_{n,z})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \\
&\quad -\frac{(\hat{d}_z)(\hat{d}_x)(\hat{d})}{\hat{\tau}^2} \\
-\frac{\partial \hat{\phi}_2}{\partial \psi_{n,y}} &= \frac{\frac{\partial \left( \frac{\psi_{n,z}}{\sqrt{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} \right)}{\partial \psi_{n,y}}}{1 + \frac{(\psi_{n,z})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \frac{-\psi_{n,z} \left( \frac{1}{2} \right) \left( (\psi_{n,x-x})^2 + (\psi_{n,y-y})^2 \right)^{-\frac{1}{2}} (2)(\psi_{n,y-y})}{\frac{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2 + (\psi_{n,z})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \\
&\quad -\frac{(\hat{d}_z)(\hat{d}_y)(\hat{d})}{\hat{\tau}^2} \\
-\frac{\partial \hat{\phi}_2}{\partial \psi_{n,z}} &= \frac{\frac{\partial \left( \frac{\psi_{n,z}}{\sqrt{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} \right)}{\partial \psi_{n,z}}}{1 + \frac{(\psi_{n,z})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \frac{\frac{\sqrt{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2 + (\psi_{n,z})^2}}{\frac{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}{(\psi_{n,x-x})^2 + (\psi_{n,y-y})^2}} = \frac{(\hat{d})^3}{\hat{\tau}^2}
\end{aligned}$$

Nos passos 2(h)b e 2(h)c do algoritmo é calculado, para cada partícula, as estimativas  $\psi_a$  e  $\Sigma_a$  do marco  $\theta_a$  inserido no mapa. Expandindo  $\psi_a$  (Fig.??) obtemos:

$$\psi_{a,x} = x + \tau \cos(\phi_2) \cos(\alpha + \phi_1) \quad (4.11)$$

$$\psi_{a,y} = y + \tau \cos(\phi_2) \sin(\alpha + \phi_1) \quad (4.12)$$

$$\psi_{a,z} = \tau \sin(\phi_2) \quad (4.13)$$

A fim de se obter  $\Sigma_a$  (passo 2(h)c), o Jacobiano é computado como na equação 4.14:

$$G_{\theta_a} = \begin{bmatrix} (\cos(\alpha + \phi_1)) \cos(\phi_2) & (\sin(\alpha + \phi_1)) \cos(\phi_2) & \sin(\phi_2) \\ -\frac{\sin(\alpha + \phi_1)}{\tau \cos(\phi_2)} & \frac{\cos(\alpha + \phi_1)}{\tau \cos(\phi_2)} & 0 \\ -(\cos(\phi_2))(\sin(\phi_2))(\tau \cos(\phi_2))(\cos(\alpha + \phi_1)) & -(\sin(\phi_2))(\sin(\alpha + \phi_1))(\cos(\phi_2))(\tau \cos(\phi_2)) & \frac{(\tau \cos(\phi_2))^2 (\cos(\phi_2))}{\tau} \end{bmatrix}$$

(4.14)

Fazendo  $d = \tau \cos(\phi_2)$ ,  $\sin_1 = \sin(\alpha + \phi_1)$ ,  $\cos_1 = \cos(\alpha + \phi_1)$ ,  $\sin_2 = \sin(\phi_2)$  e  $\cos_2 = \cos(\phi_2)$ .

$$G_{\theta_a} = \begin{bmatrix} \cos_1 \cos_2 & \sin_1 \cos_2 & \sin_2 \\ -\frac{\sin_1}{d} & \frac{\cos_1}{d} & 0 \\ -\cos_2 \sin_2 d \cos_1 & -\sin_2 \sin_1 \cos_2 d & d(\cos_2)^2 \end{bmatrix} \quad (4.15)$$

A prova da corretude de cada um dos termos da equação 4.15 é descrita abaixo (ver Fig.4.1). Compare os resultados obtidos com os termos da equação 4.10

$$\begin{aligned} - \cos_1 \cos_2 &= \left(\frac{d_x}{d}\right) \left(\frac{d}{\tau}\right) = \frac{d_x}{\tau} \\ - \sin_1 \cos_2 &= \left(\frac{d_y}{d}\right) \left(\frac{d}{\tau}\right) = \frac{d_y}{\tau} \\ - \sin_2 &= \frac{d_z}{\tau} \\ - \frac{\sin_1}{d} &= -\left(\frac{d_y}{d}\right) \left(\frac{1}{d}\right) = -\frac{d_y}{(d)^2} \\ - \frac{\cos_1}{d} &= \left(\frac{d_x}{d}\right) \left(\frac{1}{d}\right) = \frac{d_x}{(d)^2} \\ - -\cos_2 \sin_2 d \cos_1 &= -\left(\frac{d}{\tau}\right) \left(\frac{d_z}{\tau}\right) (d) \left(\frac{d_x}{d}\right) = -\frac{(d_z)(d_x)(d)}{\tau^2} \\ - -\sin_2 \sin_1 \cos_2 d &= -\left(\frac{d_z}{\tau}\right) \left(\frac{d_y}{d}\right) \left(\frac{d}{\tau}(d)\right) = -\frac{(d_z)(d_y)(d)}{\tau^2} \\ - \frac{(d)^3}{\tau^2} &= d \frac{d}{\tau} \frac{d}{\tau} = d \cos_2 \cos_2 = d(\cos_2)^2 \end{aligned}$$

A inversa da matriz 4.15 - utilizado no passo 2(h)c é como descrito na equação 4.16:

$$G'_{\theta_a} = \begin{bmatrix} \cos_1 \cos_2 & -d \sin_1 & -\frac{\sin_2 \cos_1}{d \cos_2} \\ \sin_1 \cos_2 & d \cos_1 & -\frac{\sin_2 \sin_1}{d \cos_2} \\ \sin_2 & 0 & \frac{1}{d} \end{bmatrix} \quad (4.16)$$

Nos passos 2(c)d e 2(h)c do algoritmo a matriz de covariância do erro de observação  $R$  (Eq.4.2) é como descrito na equação 4.17 (Tab.4.1):

$$R = \begin{bmatrix} \bar{\tau}N(0, \sigma) & 0 & 0 \\ 0 & \bar{\phi}_1 N(0, \sigma) & 0 \\ 0 & 0 & \bar{\phi}_2 N(0, \sigma) \end{bmatrix} \quad (4.17)$$

Vale a pena destacar que o peso da associação  $p_n$  (passo 2(c)e) depende da diferença entre a observação  $z = g(\chi) = [\tau, \phi_1, \phi_2]^T$  e a observação predita pela partícula,  $\hat{z}_n$  (passo 2(c)b). No caso de marcos 3D, a observação predita  $\hat{z}_n$  incorpora  $\hat{d}_z$  (Eq. 4.6 e Eq. 4.8). Note também que o peso da associação  $p_n$  depende do Jacobiano (Eq.4.9), devido à linearização do modelo de observação.

### 4.3 Conclusão

Neste capítulo, tratamos da abordagem FastSLAM Visual. No capítulo 5, apresentamos uma visão geral da técnica de extração de coordenadas 3D de um sensor visual estéreo convencional, bem como a técnica de extração de coordenadas 3D do sensor visual estéreo ativo *kinect*.

## CAPÍTULO 5

# Obtenção de Coordenadas 3D

O objetivo deste capítulo é apresentar aspectos relacionados com a obtenção das coordenadas 3D por intermédio de um sistema estéreo convencional (seção 5.2) por meio do sensor visual estéreo ativo *kinect* (seção 5.3)[52]. Primeiramente apresentaremos alguns conceitos fundamentais (seção 5.1) necessários ao entendimento das seções 5.2 e 5.3.

## 5.1 Conceitos Fundamentais

### 5.1.1 Introdução

Quando a luz atinge um objeto, uma parte é absorvida, e a que não é absorvida é refletida e a percebemos como a cor da luz. A luz refletida que caminha até nossos olhos (ou a uma câmera) é coletada na retina (ou no *gerador de imagens* da câmera). Toda geometria deste arranjo - particularmente a viagem do raio do objeto, através da lente para nossos olhos (ou câmera), e para a retina (ou *gerador de imagens* da câmera) - é particularmente importante para a visão computacional (Fig.5.1).

Um simples modelo de como isto acontece é o modelo de câmera *pinhole*<sup>1</sup>(Fig. 5.2). Um *pinhole* é uma parede imaginária, com um pequeno orifício no centro, que bloqueia todos os raios exceto aqueles que passam através da pequena abertura no centro. Infelizmente, um *pinhole* real não é uma boa maneira de produzir imagens porque ele não recolhe bastante luz para exposições rápidas. Devido a isto, nossos olhos (e câmeras) usam lentes para obter mais luz do que aquela que seria disponível em um único ponto.

<sup>1</sup>A tradução literal de *pinhole* é *um pequeno orifício*

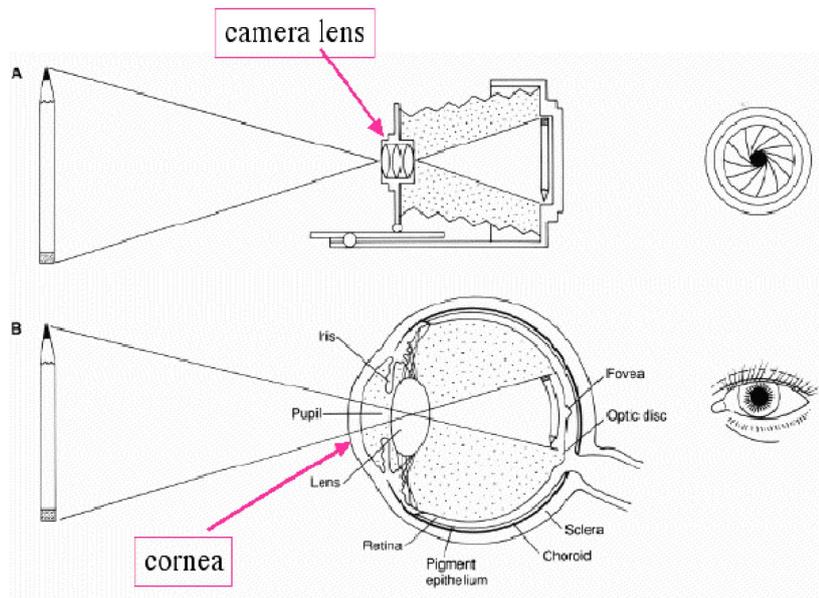


Figura 5.1: Formação de imagem em uma câmera e no olho humano [33].

Entretanto, a obtenção de mais luz com o uso de lentes exige um modelo de câmera mais complexo do que o modelo *pinhole* em que as distorções devidas às lentes são introduzidas.

### 5.1.2 Modelo *Pinhole*

No modelo *pinhole*, a luz é imaginada como vindo da cena ou de um objeto distante, mas somente um único raio, partindo de qualquer ponto, entra no *pinhole*. Este ponto é então projetado sobre a superfície de imageamento. Como resultado, a imagem sobre este *plano imagem* (ou *plano projetivo*) está sempre em foco, e o tamanho da imagem em relação à distância do objeto é controlado por um único parâmetro da câmera, a *distância focal*. Neste modelo idealizado, a distância do *pinhole* para a tela (ou *plano imagem*) é a *distância focal*. Isto é mostrado na figura 5.2, onde  $f$  é a *distância focal* da câmera,  $Z$  é a distância da câmera para o objeto,  $X$  é o tamanho do objeto, e  $x$  é o tamanho da imagem do objeto sobre o *plano imagem*. Na figura 5.2, pode ser visto, por semelhança de triângulos que

$$-x = f \left( \frac{X}{Z} \right) \quad (5.1)$$

O modelo da figura 5.2 pode ser rearranjado para formar um modelo

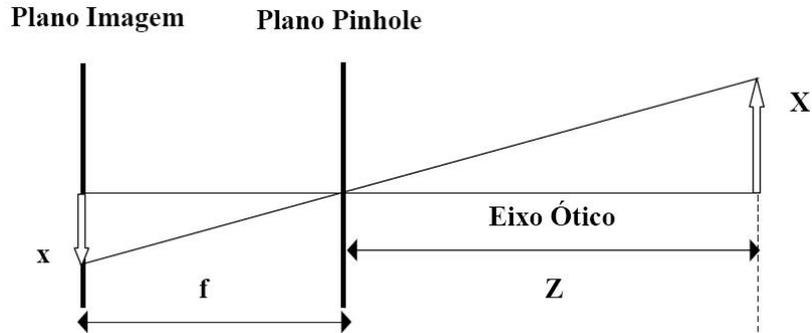


Figura 5.2: Modelo de *pinhole*: um *pinhole* (a abertura de *pinhole*) deixa somente atravessar aqueles raios de luz que interceptam um ponto particular no espaço; estes raios então formam uma imagem projetada no plano da imagem.

equivalente, mas matematicamente mais simples (Fig.5.3). Na figura 5.3, a posição do *plano pinhole* e o *plano da imagem* são trocados. A principal diferença é a de que agora o objeto aparece do lado direito superior do *plano imagem*. O ponto no *plano pinhole* é reinterpretado como o *centro de projeção*. Neste modelo, cada raio deixa um ponto sobre o objeto e se dirige para o *centro de projeção*. A imagem é gerada pela intersecção destes raios com o *plano da imagem*, que ocorre exatamente a uma distância  $f$  do *centro de projeção*. Isto torna a relação de semelhança de triângulos  $\frac{x}{f} = \frac{X}{Z}$  mais evidente do que antes.

O ponto na intersecção do *plano imagem* e o *eixo ótico* é conhecido como *ponto principal*. Poderíamos pensar que o *ponto principal* seja o centro do *gerador de imagens*. Na verdade, por questão de precisão, ao se colocar o *gerador de imagens* na câmera, o centro do *gerador de imagens* usualmente não está sobre o *eixo ótico*. Portanto, introduz-se dois novos parâmetros  $c_x$  e  $c_y$ , para modelar um possível deslocamento em relação ao *eixo ótico*. O resultado é que um modelo relativamente simples em que um ponto  $Q$  no mundo físico, cujas coordenadas são  $(X, Y, Z)$ , é projetado sobre a tela (*plano imagem*) em alguma localização em *pixel*  $(x_{tela}, y_{tela})$  de acordo com as seguintes equações:

$$x_{tela} = f_x \left( \frac{X}{Z} \right) + c_x \quad ; \quad y_{tela} = f_y \left( \frac{X}{Z} \right) + c_y \quad (5.2)$$

Note que foram introduzidas duas distâncias focais devido aos *pixels* individuais sobre um *gerador de imagens* de baixo custo serem retangulares ao invés de quadrados. A *distância focal*  $f_x$  (por exemplo) é realmente o

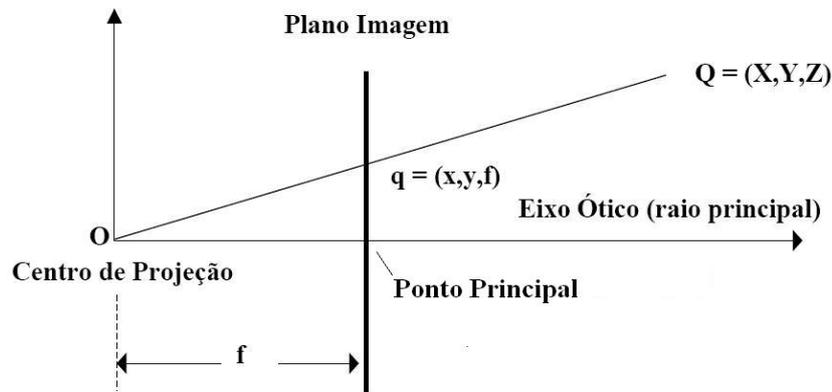


Figura 5.3: Um ponto  $Q = (X, Y, Z)$  é projetado sobre o *plano imagem* através de um raio passando pelo *centro de projeção*, e o ponto resultante sobre a imagem é  $q = (x, y, f)$ .

produto da *distância focal* física da lente  $F$  e o tamanho  $s_x$  dos elementos individuais do *gerador de imagens* (isto faz sentido por que  $s_x$  tem unidade de *pixel* por milímetro enquanto que  $F$  tem unidade de milímetro, o que significa que  $f_x$  possui a requerida unidade de *pixels*). Naturalmente, o mesmo se mantém para  $f_y$  e  $s_y$ . É importante ter em mente que  $s_x$  e  $s_y$  não podem ser medidos diretamente via qualquer processo de calibração de câmera, e nem a *distância focal* física  $F$  é diretamente medida. Somente a combinação  $f_x = F s_x$  e  $f_y = F s_y$  pode ser derivada.

### 5.1.3 Distorções da lente

Na prática, as lentes não são perfeitas devido à sua manufatura e à dificuldade de se alinhar mecanicamente as lentes e o *gerador de imagens*. As duas principais distorções da lente são a *distorção radial* e a *distorção tangencial*. A distorção radial surge da forma da lente e a tangencial do processo de montagem da câmera como um todo.

#### Distorção radial

As lentes de uma câmera real distorcem a localização do *pixel* perto das bordas do *gerador de imagens* (Fig.5.5). Este fenômeno é a fonte do *efeito barril* ou *olho de peixe*. A figura 5.4 fornece uma intuição de como a distorção radial ocorre. Com algumas lentes, raios mais distantes do centro das lentes são mais inclinados do que aqueles mais próximos.

Para a distorção radial, a distorção é nula no *centro ótico* do *gerador de*

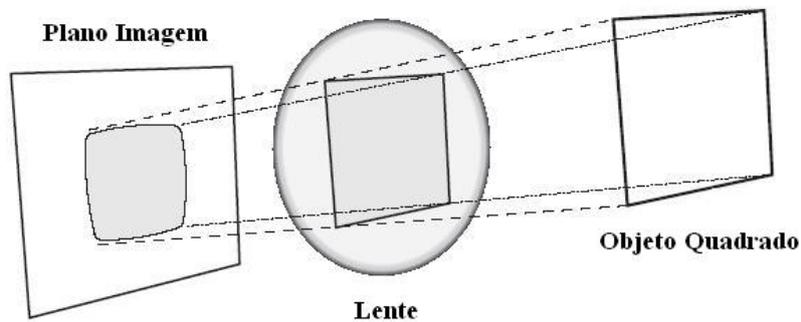


Figura 5.4: Distorção radial: raios mais distantes do centro de uma lente são mais inclinados se comparado com raios que passam perto do centro; portanto, os lados de um quadrado parecem curvar no *plano imagem* (distorção do tipo barril) [34].



Figura 5.5: Imagem da câmera distorcida (esquerda) e corrigida (direita) [53].

*imagens* e aumenta a medida que se move em direção à periferia. Na prática, esta distorção é pequena e pode ser caracterizada pelos primeiros termos da série de *Taylor* entorno de  $r = 0$ . Em geral, a localização radial de um ponto sobre o *gerador de imagens* será reescalado de acordo com as seguintes equações

$$x_{\text{corrigido}} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (5.3)$$

$$y_{\text{corrigido}} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad (5.4)$$

onde  $(x, y)$  é a localização original (sobre o *gerador de imagens*) do ponto distorcido e  $(x_{\text{corrigido}}, y_{\text{corrigido}})$  é a nova localização resultante da correção.

## Distorção tangencial

A distorção tangencial é devida a defeitos de fabricação resultante das lentes não estarem exatamente paralelas ao *plano imagem*. Ela é minimamente caracterizada por dois parâmetros adicionais,  $p_1$  e  $p_2$ , tal que:

$$x_{\text{corrigido}} = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (5.5)$$

$$y_{\text{corrigido}} = y + [2p_2x + p_1(r^2 + 2y^2)] \quad (5.6)$$

Na correção das distorções radial e tangencial cinco coeficientes são exigidos, a saber,  $k_1, k_2, k_3, p_1$  e  $p_2$ , e, que juntos, constituem o *vetor de distorção*.

## 5.2 Imageamento estéreo

Computadores podem realizar a tarefa de encontrar a correspondência entre pontos  $P$  que são vistos por um gerador de imagem ( $[x^E, y^E]$  na câmera esquerda) e os mesmos pontos  $P$  que são vistos por outro gerador de imagem ( $[x^D, y^D]$  na câmera direita) (Fig.5.6). Com tal correspondência e a separação ( $T$ ) entre as câmeras conhecida, pode-se computar a localização 3D dos pontos ( $X, Y, Z$ ). Embora a pesquisa pela correspondência de pontos seja computacionalmente custosa, pode-se utilizar o conhecimento da geometria do sistema para diminuir ao máximo o espaço de pesquisa. Na prática, o imageamento estéreo envolve quatro passos, quando se usam duas câmeras:

- Remover matematicamente a distorção radial e tangencial da lente (seção 5.1.3) em que a saída é uma imagem não distorcida.
- Ajustar o ângulo e a distância entre as câmeras, por um processo denominado **retificação** (subseção 5.2.2), em que a saída são imagens linha-alinhadas<sup>2</sup> e retificadas.
- Encontrar os mesmos *pontos de interesse* nas visões da câmera esquerda e direita, por um processo conhecido como **correspondência** (subseção 5.2.3), em que a saída é o mapa de disparidade, ou seja, as diferenças nas coordenadas  $x$  sobre os planos de imagem dos mesmos *pontos de interesse* visualizados nas câmeras esquerda e direita:  $x^E - x^D$  (Fig.5.6).

---

<sup>2</sup>Quando dizemos que as imagens são linha-alinhadas, isto significa que os planos das duas imagem são coplanares e que as linhas da imagem são alinhadas de forma exata (na mesma direção e tendo a mesma coordenada  $y$ ).

- Conhecendo o arranjo geométrico das câmeras, transformar o mapa de disparidade em distâncias por **triangulação** (subseção 5.2.1), por um processo conhecido como reprojeção, em que a saída é o mapa de profundidade.

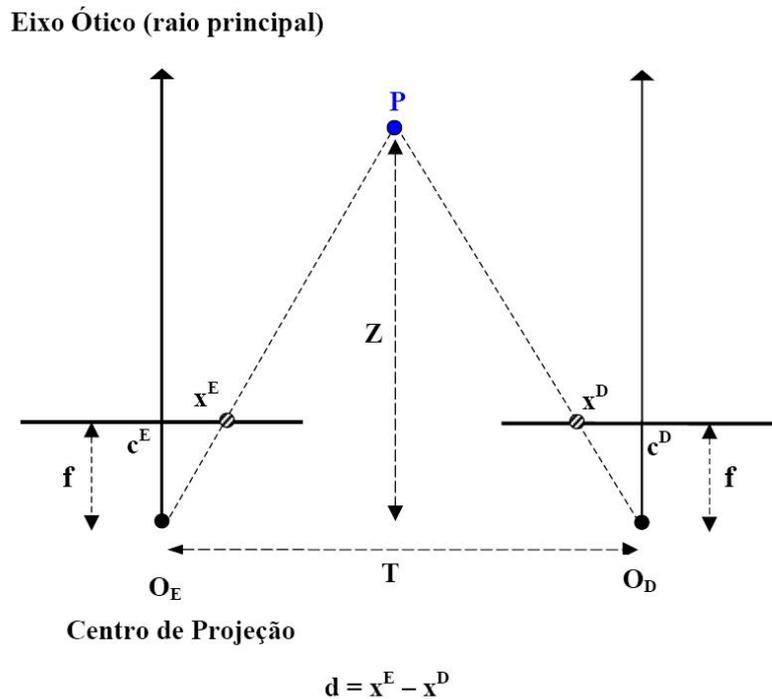


Figura 5.6: Em um equipamento estéreo perfeitamente alinhado, a profundidade  $Z$  é obtida por semelhança de triângulos.

### 5.2.1 Triangulação

Vamos assumir um equipamento estéreo perfeitamente alinhado e medido e sem distorção, como mostrado na figura 5.6: duas câmeras cujo planos de imagem são exatamente coplanares um em relação ao outro, com eixos óticos exatamente paralelos que estão a uma distância conhecida  $T$ , e com distâncias focais iguais  $f_E = f_D$ . Vamos também assumir que os *pontos principais*  $c^E$  e  $c^D$  tenham sido calibrados para terem a mesma coordenada em *pixel* nas respectivas imagens esquerda e direita.

Vamos também assumir que as imagens são linha-alinhadas e toda linha de *pixel* de uma câmera alinha-se exatamente com a linha correspondente na outra câmera. Nós denominaremos este arranjo de *Paralelo Frontal*. Vamos também assumir que podemos encontrar um ponto  $P$  nas coordenadas

mundo nas visões das imagens esquerda e direita em  $p^E$  e  $p^D$  que possuem respectivamente coordenadas horizontais  $x^E$  e  $x^D$ .

Neste caso simplificado, pode-se mostrar por semelhança de triângulos que a profundidade  $Z$  é inversamente proporcional a disparidade  $d$  entre as visões, ou seja,  $d = x^E - x^D$ . Pela figura temos:

$$\frac{T - (x^E - x^D)}{Z - f} = \frac{T}{Z} \Rightarrow Z = \frac{fT}{x^E - x^D} = \frac{fT}{d} \quad (5.7)$$

Uma vez que a profundidade  $Z$  é inversamente proporcional à disparidade, existe obviamente uma relação não linear entre estes dois termos ( $Z$  e  $d$ ). Quando a disparidade está próxima de 0, pequenas diferenças na disparidade implicam em grandes diferenças na profundidade. Quando a disparidade é grande, pequenas diferenças na disparidade não alteram muito a profundidade. Em conseqüência, sistemas de visão estéreo possuem alta resolução na profundidade somente para objetos que estão relativamente pertos da câmera, como a figura 5.7 torna claro.

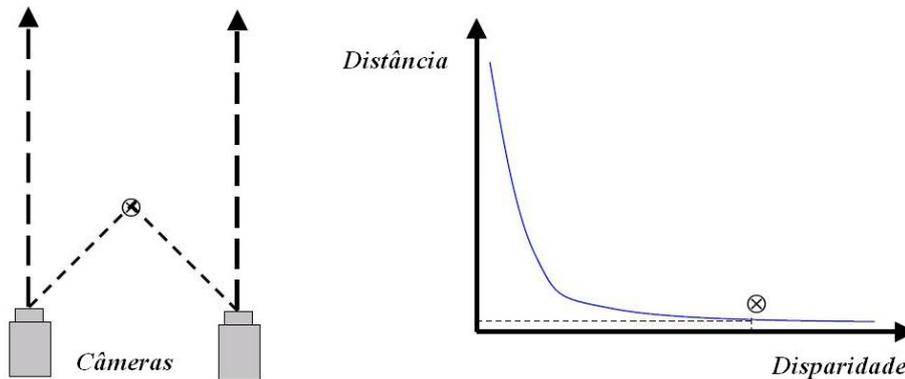


Figura 5.7: Profundidade ( $Z$ ) e disparidade ( $d$ ) são inversamente relacionados, tal que uma medição precisa da profundidade é restrita somente a objetos próximos.

Na figura 5.8 temos um exemplo de uma câmera estéreo, da marca “Bumblebee 2”, fabricada pela empresa “Pont Grey” [53]. Em uma de suas configurações, esta câmera tem um campo visual de  $97^\circ$  (horizontal e vertical).

### 5.2.2 Retificação estéreo

É mais fácil calcular a disparidade estéreo quando os dois planos da imagem se alinham de maneira exata (Fig.5.6). Infelizmente, câmeras raramente são perfeitamente alinhadas na configuração paralelo frontal (Fig.5.6). Ao invés disto, mapas de distorção e projeção que retificam matematicamente

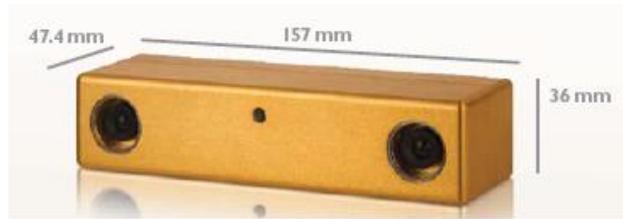


Figura 5.8: Câmera estereoscópica “Bumblebee 2” [53]

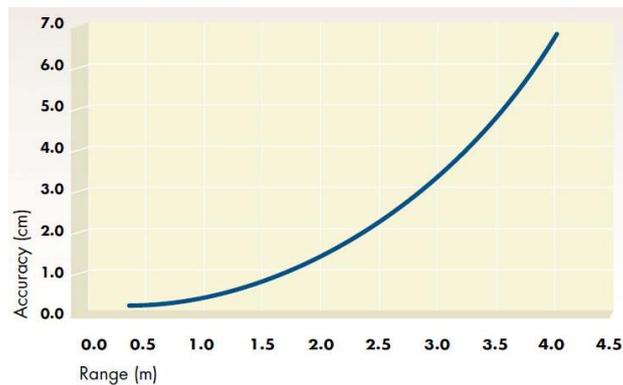


Figura 5.9: Alcance *versus* precisão da câmera estereoscópica ”Bumblebee 2” [53]

as imagens esquerda e direita na configuração paralelo frontal 5.10 devem ser obtidos.

A retificação, ou seja, o alinhamento das linhas das imagens nas duas câmeras, visa facilitar a correspondência estéreo (encontrar o mesmo ponto nas duas diferentes visões das câmeras), tornando-a mais confiável e computacionalmente tratável. Note que confiabilidade e a eficiência computacional são ambas reforçadas pela necessidade de se pesquisar somente uma linha para uma correspondência com um ponto na outra imagem. O algoritmo descrito em [7], entre outros, implementa a retificação estéreo.

A figura 5.11 exhibe os resultados da retificação e eliminação da distorção de um par de imagens estéreo. Note que os pontos se tornam horizontalmente alinhados nas imagens retificadas e não distorcidas.

### 5.2.3 Correspondência estéreo

A correspondência estéreo, ou seja, a correspondência de um ponto  $3D$  na visão de duas câmeras (por exemplo  $x^E$  e  $x^D$  na figura 5.6), pode ser calculada somente sobre áreas visuais em que a visão das duas câmeras se sobrepõem. Por este motivo, melhores resultados tendem a ser obtidos se as câmeras

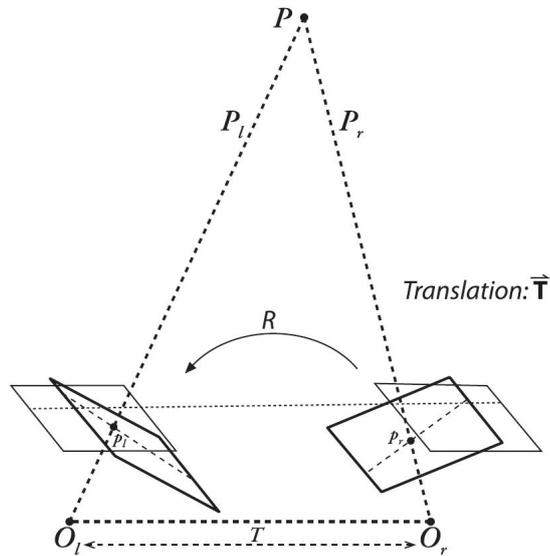


Figura 5.10: Alinhamento matemático (e não físico) de duas câmeras em um plano de visualização tal que linhas de *pixels* entre as câmeras estejam perfeitamente alinhadas [34].

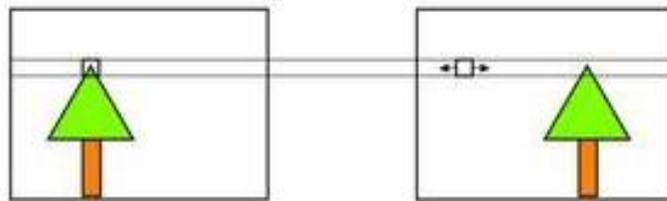


Figura 5.11: Espaço de pesquisa após a retificação estéreo.

estiverem na configuração paralelo frontal.

Existem alguns algoritmos que calculam a correspondência estéreo tal como o descrito em [5], que é rápido e eficaz. Este algoritmo encontra somente pontos com forte correspondência (alta textura) entre duas imagens. Portanto, em uma cena altamente texturizada tal como pode ocorrer em uma floresta, todo *pixel* pode ter a sua profundidade calculada. Por outro lado, em uma cena com baixa textura, tal como um corredor interno, muito poucos pontos podem ter sua profundidade computada.

### 5.3 Coordenadas 3D obtidas de um sensor *Kinect*

O sensor *Kinect* consiste de um emissor de laser infravermelho, uma câmera infravermelha e uma câmera RGB (Fig.5.12). Os inventores descrevem a medição de profundidade, como um processo de triangulação (seção 5.2.1). A fonte de laser emite um feixe único, que é dividido em múltiplos feixes por uma grade de difração para criar um padrão constante de pontos projetados na cena correspondente a cada *pixel* da imagem (Fig.5.13 - esquerda). Este padrão é capturado pela câmera infravermelha e correlacionado com um padrão de referência o que permite que se recupere a distância de cada ponto em relação ao sensor através do passos descritos mais adiante (Fig.5.13 - direita).

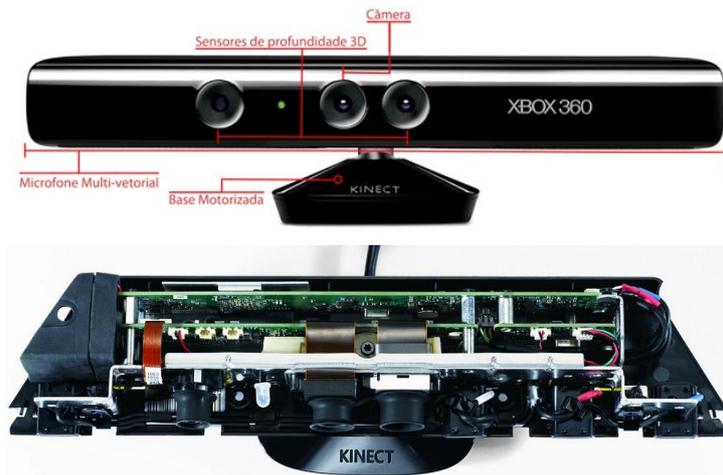


Figura 5.12: Sensor Visual *kinect* [52].

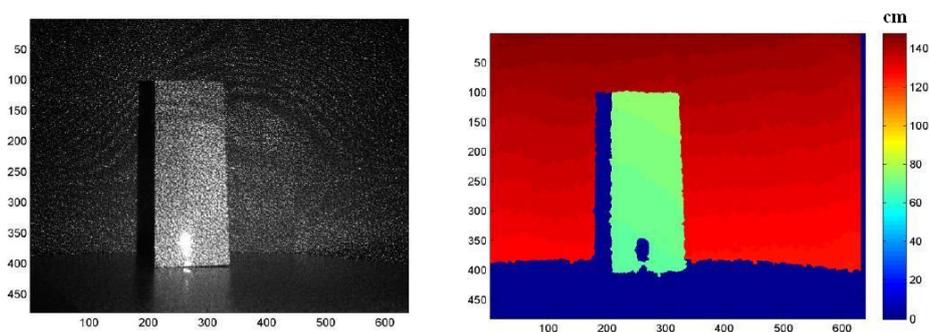


Figura 5.13: Imagem infravermelha do padrão de pontos projetados sobre uma cena (esquerda) e imagem resultante - mapa de profundidade - (direita)[52].

A figura 5.14 ilustra a relação entre a distância  $Z_k$  de um ponto  $k$  para o sensor em função do plano referência e da disparidade medida  $d$ . Para expressar as coordenadas 3D dos pontos, considera-se um sistema de coordenadas de profundidade com origem no centro perspectivo da câmera infravermelha. O eixo  $Z$  é ortogonal ao *plano imagem* em relação ao objeto, o eixo  $X$  é perpendicular ao eixo  $Z$  na direção da linha de base  $b$  entre o centro da câmera infravermelha e o projetor laser, e o eixo  $Y$  é ortogonal a  $X$  e  $Z$ .

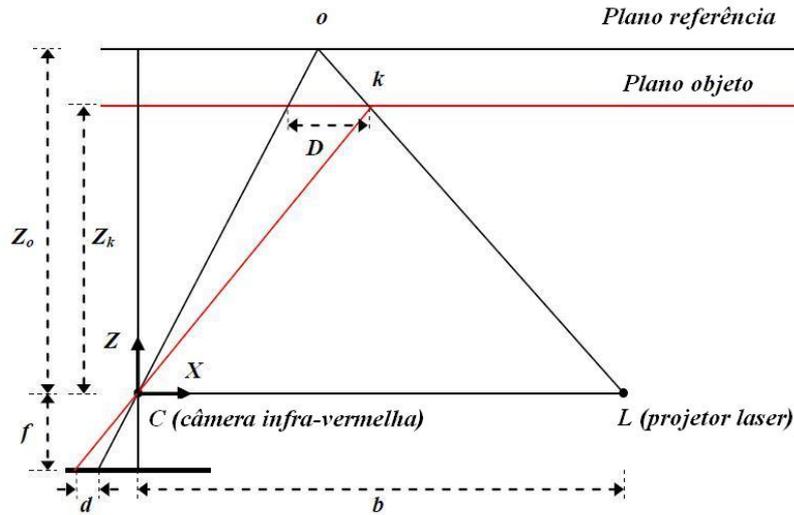


Figura 5.14: Relação entre a profundidade relativa ( $Z_k$ ) e a disparidade medida ( $d$ ) [52].

Assuma que um ponto  $k$  está sobre o plano referência a uma distância  $Z_o$  do sensor, e o referido ponto é capturado no *plano imagem* da câmera infravermelha. Se o ponto é deslocado para mais perto (ou para mais longe) do sensor, o ponto no *plano imagem* será deslocado na direção  $X$ . Este deslocamento é medido no *plano imagem* como uma disparidade  $d$  correspondente ao ponto  $k$ . A partir da semelhança de triângulos temos:

$$\frac{D}{b} = \frac{Z_o - Z_k}{Z_o} \quad (5.8)$$

$$\frac{d}{f} = \frac{D}{Z_k} \quad (5.9)$$

onde  $Z_k$  denota a distância (profundidade) de um ponto  $k$ ,  $b$  é comprimento base,  $f$  é a *distância focal* da câmera infravermelha,  $D$  é o deslocamento do ponto  $k$  e  $d$  é a disparidade observada no *plano imagem*. Substituindo

$D$  (Eq.5.8) na equação 5.9 e expressando  $Z_k$  em termo das outras variáveis temos:

$$Z_k = \frac{Z_o}{1 + \frac{Z_o}{fb}d} \quad (5.10)$$

A equação 5.10 é o o modelo matemático básico para a derivação de profundidade a partir da disparidade observada uma vez que os parâmetros constantes  $Z_o$ ,  $f$ , e  $b$  podem ser determinados por calibração. As coordenadas no plano de cada ponto  $k$  podem então ser calculadas:

$$X_k = -\frac{Z_k}{f}(x_k - x_o - \delta x) \quad (5.11)$$

$$Y_k = -\frac{Z_k}{f}(y_k - y_o - \delta y) \quad (5.12)$$

onde  $x_k$  e  $y_k$  são as coordenadas imagem do ponto  $k$ ,  $x_o$  e  $y_o$  são as coordenadas do *ponto principal*, e  $\delta x$  e  $\delta y$  são as correções para distorções na lente, para o que muitos modelos com coeficientes diferentes existem.

O sensor *kinect* (sem calibração) possui os seguintes parâmetros: alcance efetivo de 0.8 a 3.5 metros, um campo visual igual a  $58^\circ$  (horizontal) e  $45^\circ$  (vertical) [52]. A precisão do *kinect* é de 5 cm para uma distância de 2 m.

*OpenKinect* [54] é uma comunidade aberta de pessoas interessadas em usar o *amazing Xbox Kinect hardware* com computadores e outros periféricos. Neste trabalho, as distâncias  $d_x$ ,  $d_y$  e  $d_z$  de cada descritor SIFT (seção 3, Fig.4.1) para o sensor *kinect* foram obtidos com a biblioteca *Openkinect*. Nesta biblioteca,  $d_y$  e  $d_z$  são obtidos a partir de  $d_x$  de acordo com as equações 5.13 e 5.14.

$$d_y = (x_{pixel} - \left(\frac{640}{2}\right))(d_x - 10) * 0,0021 * \left(\frac{640}{480}\right) \quad (5.13)$$

$$d_z = (y_{pixel} - \left(\frac{640}{2}\right))(d_x - 10) * 0,0021 \quad (5.14)$$

onde  $x_{pixel} \in \{0 - 630\}$  e  $y_{pixel} \in \{0 - 480\}$  são as coordenadas do *pixel*.

## 5.4 Conclusão

Neste capítulo, apresentamos uma visão geral da técnica de extração de coordenadas 3D de um sensor visual estéreo convencional, bem como a técnica de extração de coordenadas 3D do sensor visual estéreo ativo *kinect*. No capítulo 4.2, tratamos da abordagem FastSLAM Visual com múltiplos veículos em que os mapas de características dos marcos detectados é comum aos mesmos.

## CAPÍTULO 6

# FastSLAM Visual com um Mapa Comum aos Veículos

### 6.1 Introdução

Quando múltiplos veículos têm a possibilidade de se comunicarem com um agente central, eles podem trabalhar juntos para reduzir o tempo de exploração e permitir uma cooperação na observação de marcos no ambiente. A motivação para utilização da abordagem FastSLAM com um mapa comum aos veículos é a de que o veículo não necessita retornar a uma área anteriormente explorada a fim de reduzir a incerteza de sua pose (Fig.1.5). O veículo pode reduzir a incerteza de sua pose quando reobserva um marco anteriormente mapeado pelo outro veículo.

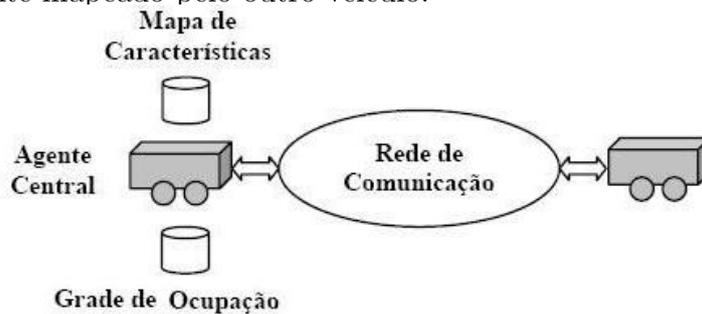


Figura 6.1: Esquema da exploração com dois veículos em que um agente central é responsável pela construção do mapa de características (localização) e da grade de ocupação (exploração).

Se  $k$  veículos exploram o ambiente construindo um mapa comum (Fig. 6.1) e no instante  $t$  o veículo ( $i$ ) na pose  $\chi_{(i),t}$  executa uma única observação  $z_{(i),t}$ , a distribuição de probabilidade posterior sobre os caminhos  $\chi_{(1:k)}^t$  dos  $k$  veículos e o mapa  $\Theta$  podem ser estimados a partir da seguinte função [39]:

$$p(\chi_{(1:k)}^t, \Theta | z_{(1:k)}^t, u_{(1:k)}^t, a^t) = \underbrace{p(\chi_{(1:k)}^t | z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{\text{Caminho}} \prod_{n=1}^N \underbrace{p(\theta_n | \chi_{(1:k)}^t, z_{(1:k)}^t, u_{(1:k)}^t, a^t)}_{\text{Marco}}$$

onde  $\chi_{(1:k)}^t = \{\chi_{(1)}^t, \dots, \chi_{(k)}^t\}$ ,  $u_{(1:k)}^t = \{u_{(1)}^t, \dots, u_{(k)}^t\}$  e  $z_{(1:k)}^t = \{z_{(1)}^t, \dots, z_{(k)}^t\}$  são respectivamente o conjunto de caminhos, ações e observações dos  $k$  veículos e  $a^t = \{a_1, \dots, a_t\}$  é o histórico de associações de dados.

A a figura 6.2 descreve graficamente o processo de aquisição de dados, na forma de uma rede Bayesiana dinâmica.

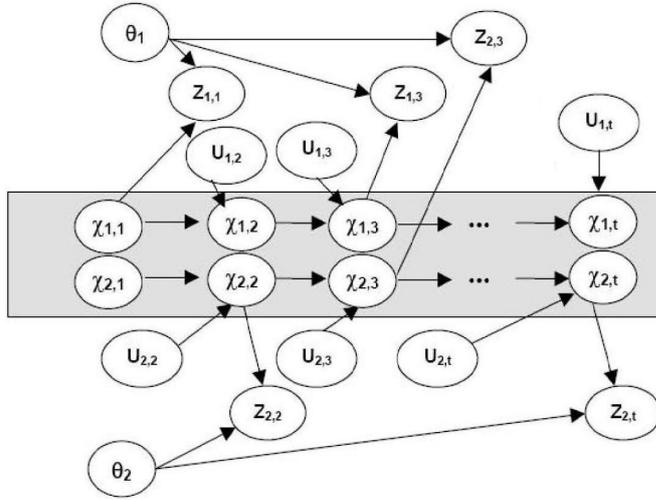


Figura 6.2: Rede de Bayes para SLAM com dois veículos.

A distribuição de probabilidade posterior sobre  $\chi_{(1:k)}^t$  é estimada usando  $k$  filtros de partículas. Por outro lado, a distribuição de probabilidade posterior sobre os  $N$  marcos  $\theta_i$  de cada partícula são estimados por meio de  $kN$  Filtros de Kalman Estendidos independentes condicionados aos caminhos  $\chi_{(1:k)}^t$ .

Uma vez que o mapa é comum aos veículos, os  $k$  filtros de partículas produzem um mesmo conjunto  $S_t$  de partículas

$$S_t^{[m]} = \left[ \chi_{(1:k)}^{[m]}, \underbrace{\psi_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, d_1^{[m]}, \dots, \psi_{N,t}^{[m]}, \Sigma_{N,t}^{[m]}, d_N^{[m]}, w_t^{[m]}}_{\Theta^{[m]}} \right] \quad (6.1)$$

O estado a ser estimado é composto pela pose  $\chi_{t,(1:k)}$  dos  $k$  veículos. O número de partículas necessárias para se obter uma boa estimativa cresce exponencialmente com a dimensão do estado a ser estimado [25].

Um algoritmo para o FastSLAM Visual com Múltiplos Veículos é apresentado a seguir [39]. A rotina FastSLAM\* no Algoritmo 6.1 corresponde ao algoritmo FastSLAM da seção 4.2 considerando apenas os passos 1 – 3.

<b>FastSLAM com Múltiplos (dois) Veículos e Mapa Comum</b>
--

1.  $S_t = \emptyset$
2. Para  $t = 1$  até o Fim fazer
  1.  $[z_{(1),t}, d_{(1),t}, z_{(2),t}, d_{(2),t}] = \text{Observações}()$
  2.  $[S_t, w_{(1),t}] = \text{FastSLAM}^*(S_{t-1}, z_{(1),t}, d_{(1),t}, u_{(1),t})$
  3.  $[S_t, w_{(2),t}] = \text{FastSLAM}^*(S_{t-1}, z_{(2),t}, d_{(2),t}, u_{(2),t})$
  4.  $w_t = w_{(1),t} w_{(2),t}$
  5.  $S_t = \text{Reamostragem}(S_t, w_t)$
3. Fim Para

algoritmo 6.1: FastSLAM com Múltiplos (dois) Veículos e Mapa Comum

Este algoritmo considera que os veículos iniciam a exploração a partir de uma pose conhecida pelo agente central. Considerando que os veículos iniciam a exploração de posições próximas, a pose relativa pode ser obtida através do sensor a laser (Fig.2.2).

Cada veículo inclui suas observações em um ordem pré-definida (passos 2 e 3). Desde que as observações sejam independentes e Gaussianas, esta ordem não afeta a estimativa final dos marcos. O algoritmo FastSLAM no caso de  $k$  veículos requer tempo  $kM \log N$ .

Uma vez que temos um mapa comum aos veículos, para cada partícula  $S_t^{[m]}$  definida na equação 6.1,  $k$  pesos são calculados e um peso total associado com a partícula  $S_t^{[m]}$  é definido como

$$w_t^{[m]} = \sqrt[k]{\prod_{i=1}^k w_{(i),t}^{[m]}} \quad (6.2)$$

## 6.2 Reamostragem Estratificada

A tabela 6.1 ilustra o funcionamento do algoritmo de reamostragem. Na 11ª coluna temos as partículas reamostradas em função das partículas que tiveram marcos atualizados (1ª coluna).

## 6.3 Conclusão

Neste capítulo, tratamos da abordagem FastSLAM Visual com múltiplos veículos e um mapa de características comum aos veículos. No capítulo 7,

Tabela 6.1: Reamostragem considerando 2 veículos ( $M = 40$ )

m	$w_1$	$w_2$	$w$	$\frac{w_m}{w_{soma}}$ (passo 2)	$d[m]$ (passo 7.1)	$f_{dist.unif}$	$selecao[m]$ (passo 9.1)	$w_m$ (passo 4)	$m$	$manter[m]$ (passo 12.1.1)
1	0,9	0,025	0,15	0,10909	0,0125	0,1	0,0025	0,109091	1	1
2	0,9	0,025	0,15	0,10909	0,0375	0,9	0,0475	0,218182	2	1
3	0,9	0,025	0,15	0,10909	0,0625	0,3	0,0575	0,327273	3	1
4	0,025	0,025	0,025	0,01818	0,0875	0,4	0,085	0,345455	4	1
5	0,025	0,025	0,025	0,01818	0,1125	0,5	0,1125	0,363636	5	2
6	0,025	0,025	0,025	0,01818	0,1375	0,4	0,135	0,381818	6	2
7	0,025	0,025	0,025	0,01818	0,1625	0,6	0,165	0,400000	7	2
8	0,025	0,025	0,025	0,01818	0,1875	0,1	0,1775	0,418182	8	2
9	0,025	0,025	0,025	0,01818	0,2125	0,1	0,2025	0,436364	9	2
10	0,025	0,025	0,025	0,01818	0,2375	0,1	0,2275	0,454545	10	3
11	0,025	0,025	0,025	0,01818	0,2625	0,1	0,2525	0,472727	11	3
12	0,025	0,025	0,025	0,01818	0,2875	0,1	0,2775	0,490909	12	3
13	0,025	0,025	0,025	0,01818	0,3125	0,3	0,3075	0,509091	13	13
14	0,025	0,025	0,025	0,01818	0,3375	0,1	0,3275	0,527273	14	14
15	0,025	0,025	0,025	0,01818	0,3625	0,1	0,3525	0,545455	15	15
16	0,025	0,025	0,025	0,01818	0,3875	0,2	0,38	0,563636	16	16
17	0,025	0,025	0,025	0,01818	0,4125	0,1	0,4025	0,581818	17	17
18	0,025	0,025	0,025	0,01818	0,4375	0,1	0,4275	0,600000	18	18
19	0,025	0,025	0,025	0,01818	0,4625	0,1	0,4525	0,618182	19	19
20	0,025	0,025	0,025	0,01818	0,4875	0,1	0,4775	0,636364	20	20
21	0,025	0,025	0,025	0,01818	0,5125	0,1	0,5025	0,654545	21	21
22	0,025	0,025	0,025	0,01818	0,5375	0,9	0,5475	0,672727	22	22
23	0,025	0,025	0,025	0,01818	0,5625	0,3	0,5575	0,690909	23	23
24	0,025	0,025	0,025	0,01818	0,5875	0,4	0,585	0,709091	24	24
25	0,025	0,025	0,025	0,01818	0,6125	0,5	0,6125	0,727273	25	25
26	0,025	0,025	0,025	0,01818	0,6375	0,4	0,635	0,745455	26	26
27	0,025	0,025	0,025	0,01818	0,6625	0,6	0,665	0,763636	27	27
28	0,025	0,025	0,025	0,01818	0,6875	0,1	0,6775	0,781818	28	28
29	0,025	0,025	0,025	0,01818	0,7125	0,1	0,7025	0,800000	29	29
30	0,025	0,025	0,025	0,01818	0,7375	0,1	0,7275	0,818182	30	30
31	0,025	0,025	0,025	0,01818	0,7625	0,1	0,7525	0,836364	31	31
32	0,025	0,025	0,025	0,01818	0,7875	0,1	0,7775	0,854545	32	32
33	0,025	0,025	0,025	0,01818	0,8125	0,3	0,8075	0,872727	33	33
34	0,025	0,025	0,025	0,01818	0,8375	0,1	0,8275	0,890909	34	34
35	0,025	0,025	0,025	0,01818	0,8625	0,1	0,8525	0,909091	35	35
36	0,025	0,025	0,025	0,01818	0,8875	0,2	0,88	0,927273	36	36
37	0,025	0,025	0,025	0,01818	0,9125	0,1	0,9025	0,945455	37	37
38	0,025	0,025	0,025	0,01818	0,9375	0,1	0,9275	0,963636	38	38
39	0,025	0,025	0,025	0,01818	0,9625	0,1	0,9525	0,981818	39	39
40	0,025	0,025	0,025	0,01818	0,9875	0,1	0,9775	1,000000	40	40

$w_{soma}$	$N_{efetivo}$	$k$	$q$
1,375	20,86	0,025	0,0125

trataremos da exploração de um ambiente interno utilizando uma grade de ocupação gerada por intermédio de um sensor laser.

## CAPÍTULO 7

# Exploração com Grades de Ocupação

Neste capítulo, são apresentadas técnicas de exploração que utilizam mapas do tipo grade de ocupação. Na seção 7.1, é descrita a técnica de construção das referidas grades. As seções 7.2 e 7.3 tratam, respectivamente, da exploração com um veículo e múltiplos veículos. Na seção 7.4, é descrita uma estratégia de exploração integrada do ambiente [24].

### 7.1 Mapeamento com grade de ocupação

Mapas do tipo grade de ocupação (Fig.7.1) resolvem o problema da geração de mapas consistentes a partir de dados de medições ruidosas e incertas, sob a suposição de que a pose do veículo é conhecida. Com não existe veículo com odometria perfeita, uma grade é utilizada após o problema do SLAM ter sido resolvido através de alguma técnica.

A idéia básica das grades de ocupação é a de representar o mapa como um conjunto de variáveis aleatórias (binárias) dispostas em uma grade uniformemente espaçada. A referida variável corresponde à ocupação do local que ela cobre (ocupada ou desocupada). Os algoritmos de mapeamento do tipo grade de ocupação implementam a estimação da distribuição de probabilidade posterior destas variáveis aleatórias.

Uma grade de ocupação calcula a distribuição de probabilidade posterior sobre o mapa  $G$  a partir dos dados de entrada

$$p(G|z_{1:t}, \chi_{1:t}) \tag{7.1}$$

De longe, grades bidimensionais - que descrevem uma fatia  $2D$  do mundo  $3D$  - são os tipos de mapas mais comuns. Mapas  $2D$  são muitas vezes sufi-

cientes, especialmente quando um veículo navega sobre uma superfície plana e os sensores estão montados de modo que eles capturam apenas uma fatia do mundo. Entretanto, técnicas  $2D$  podem ser generalizadas para  $3D$ .

Considere  $G_i$  a célula da grade com índice  $i$ . Uma grade de ocupação particiona o espaço em células de tamanho finito:

$$G = \sum_i G_i \quad (7.2)$$

Cada célula  $G_i$  está associada a um valor de binário de ocupação, que especifica se uma célula está ocupada ou livre. Consideramos "1" como ocupado e "0" como livre. A notação  $p(G_i = 1)$  ou  $p(G_i)$  refere-se à probabilidade de que a célula  $i$  da grade esteja ocupada.

O problema em estimar a distribuição de probabilidade posterior do mapa  $G$  de acordo com a equação 7.1 é a sua dimensão, pois o número de mapas definidos neste espaço é enorme. Portanto, calcular a distribuição de probabilidade posterior de um único mapa é intratável.

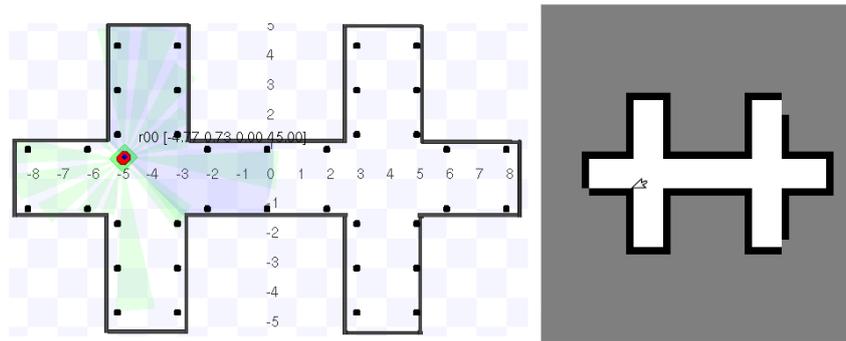


Figura 7.1: Mapa do tipo grade de ocupação (direita) de um ambiente (esquerda): as regiões claras e escuras correspondem, respectivamente, a espaços livres e ocupados.

Uma abordagem para solução deste problema é decompor o problema de estimar o mapa em um conjunto de problemas mais simples, a saber, aquele que estima

$$p(G_i | z_{1:t}, \chi_{1:t}) \quad (7.3)$$

para todas as células  $i$  da grade.

Esta decomposição é conveniente, mas não é isenta de problemas. Em particular, ela não permite representar as dependências entre as células vizinhas. Em vez disso, a distribuição de probabilidade posterior sobre o mapa

é aproximado como o produto das distribuições de probabilidade marginais:

$$p(G|z_{1:t}, \chi_{1:t}) = \prod_i p(G_i|z_{1:t}, \chi_{1:t}) \quad (7.4)$$

O algoritmo 7.1 calcula a ocupação de uma grade a partir do modelo inverso de um sensor laser (Alg.7.2). Este algoritmo utiliza a representação *log-odds* (Eq.7.5) para a ocupação que tem a vantagem de evitar instabilidades para probabilidades próximas de zero.

$$l_{t,i} = \log \left( \frac{p(G_i|z_{1:t}, \chi_{1:t})}{1 - p(G_i|z_{1:t}, \chi_{1:t})} \right) \quad (7.5)$$

O valor da probabilidade é facilmente recuperada a partir da representação *log-odds* (Tab.7.1):

$$p(G_i|z_{1:t}, \chi_{1:t}) = 1 - \left( \frac{1}{1 + \exp(l_{t,i})} \right) \quad (7.6)$$

Tabela 7.1: Probabilidade nas notações *log-odds* (Eq.7.5) e padrão (Eq.7.6).

$l_{t,i}$	$p(G_i z_{1:t}, \chi_{1:t})$
-3	0,047
-2	0,119
-1	0,268
0	0,500
1	0,731
2	0,880
3	0,952

Geração\_de\_grade\_de\_ocupação ( $\chi_t, z_t, l_{t,i}$ )

- 1 Para todas as célula  $m_i$ 
  - a Se  $m_i$  está no campo percentual de  $a_t$ 
    - Então  $l_{t,i} = l_{t,i} + \text{Modelo\_Inverso\_do\_Sensor}(G_i, \chi_t, z_t) - l_0$
    - Senão  $l_{t,i} = l_{t,i}$
  - d Fim Se
- 2 Fim Para
- 3 Retornar  $l_{t,i}$

algoritmo 7.1: Geração de grade de ocupação

O algoritmo 7.1 itera sobre todas as células  $i$  da grade e atualiza aquelas que caem dentro do cone do sensor (Fig.7.2), de acordo com a medida

modelo inverso do sensor (Alg.7.2). A constante  $l_0$  (Eq.7.7) no algoritmo 7.1 é a distribuição de probabilidade anterior na notação *log-odds*.

$$l_0 = \log \left( \frac{p(G_i = 1)}{p(G_i = 0)} \right) = \log \left( \frac{p(G_i)}{1 - p(G_i)} \right) = \log \left( \frac{0,5}{1 - 0,5} \right) = 0 \quad (7.7)$$

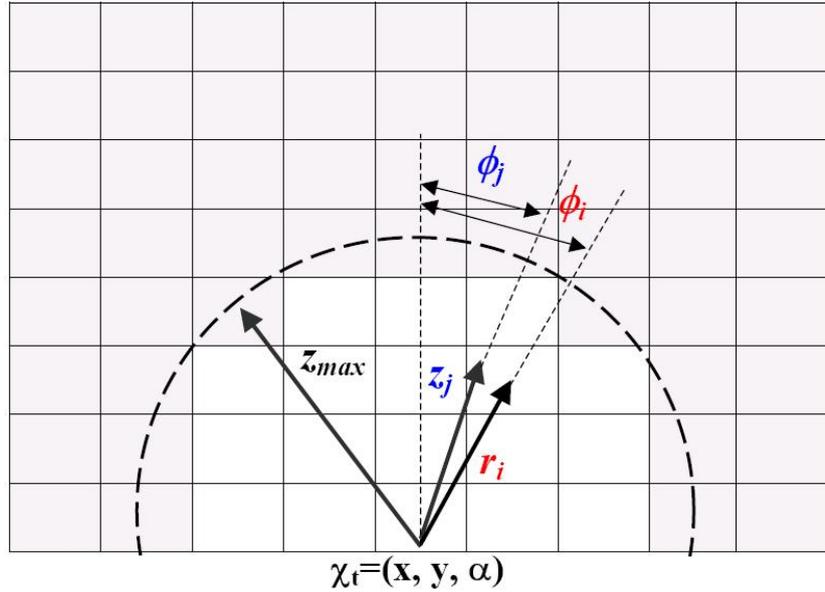


Figura 7.2: Cone de um sensoriamento a laser obtido pelo veículo:  $\chi_t = (x, y, \alpha)^T$  é a pose do veículo;  $\beta$  é o ângulo de abertura do feixe laser ( $\beta = 180^\circ$  neste caso);  $z_{max}$  é o alcance máximo do sensor laser;  $z_{j,sens}$  e  $\phi_{j,sens}$  são, respectivamente a distância e a orientação obtidas pelo sensor laser ( $z_{j,sens} \leq z_{max}$ ;  $-\beta/2 \leq \phi_{j,sens} \leq \beta/2$ ;  $1 \leq j \leq 360$ );  $r_i$  é a distância entre  $\chi_t$  e o centro de massa de uma célula  $G_i$  de coordenadas  $(x_i, y_i)$ .

O algoritmo 7.2 descreve um modelo inverso de um sensor laser. Este modelo atribui para todas as células dentro do cone do sensor, cuja distância é próxima ao alcance da medida  $z_t$ , um valor  $l_{oc} > l_0$ . O tamanho desta região é controlada pelo parâmetro  $\gamma$  e pelo ângulo de abertura do feixe.

A pose do veículo é denotado por  $\chi_t = (x, y, \alpha)^T$ . O modelo inverso (Alg.7.2, Fig.7.2) primeiramente determina as coordenadas  $(x_i, y_i)$  do centro de massa da célula  $G_i$  dado seus índices  $(i, j)$  na grade e o tamanho e escala da grade de ocupação (passo 1).

Nos passos 2 e 3 são calculados respectivamente a distância ( $r_i$ ) e o ângulo ( $\phi_i$ ) entre a pose do veículo ( $\chi_t$ ) e as coordenadas da célula  $m_i$ .

O passo 4 possibilita que se considere apenas as células  $G_i$  que estejam na frente do sensor laser (se considerarmos  $\beta = 180^\circ$ )<sup>1</sup>.

<sup>1</sup>Os passos 4, 4(a) e 4(b) foram modificações que introduzimos a fim de a abordagem,

No passo 4(c) é encontrado o índice  $j$  do feixe do laser que melhor se encaixe à observação da célula  $m_i$ .

O passo 4(d) retorna a probabilidade de ocupação anterior ( $l_0$ ), na notação *log-odds*, quando a célula está fora do alcance de medição ou a uma distância de medição superior a  $z_t^k + \gamma/2$ .

O passo 4(e) retorna  $l_{oc} > l_0$ , se a distância da célula esta à  $\pm\gamma/2$  da distância detectada  $z_t^k$ .

Ele retorna  $l_{livre} < l_0$ , se a distância para a célula é menor ou igual do que a distância detectada  $z_t^k$  (passo 4(f)).

No algoritmo 7.2 (passo 3), a função  $atan_2$  retorna um ângulo  $-180^\circ \leq \phi \leq 180^\circ$  (Tab.7.2). Os passos 4(a) e 4(b) corrigem  $\phi$  na situação em que  $\Delta x = x_i - x < 0$  e  $\Delta y = y_i - y < 0$ .

`Modelo_Inverso_de_Sensor( $m_i, \chi_t, z_t$ )`

1. Seja  $x_i, y_i$  o centro de massa de  $m_i$  ( $x_i = \frac{(i-0.5-\frac{tam}{2})}{escala}$   $y_i = \frac{(j-0.5-\frac{tam}{2})}{escala}$ )
2.  $r_i = \sqrt{(x_i - x)^2 + (y_i - y)^2}$
3.  $\phi_i = atan_2(\frac{y_i - y}{x_i - x}) - \alpha$
4. Se  $((x_i - x) * (\cos(\alpha)) + (y_i - y) * (\sin(\alpha)) > 0)$ 
  - (a) Se  $((abs(\phi_i) > pi) \text{ e } (\alpha > 0))$  então  $\phi_i = +(2pi - abs(\phi_i))$
  - (b) Se  $((abs(\phi_i) > pi) \text{ e } (\alpha < 0))$  então  $\phi_i = -(2pi - abs(\phi_i))$
  - (c)  $k = arg\_min_j |\phi_i - \phi_{j,sens}|$  ( $1 \leq j \leq 360$ )
  - (d) Se  $r > min(z_{max}, z_t^k + \gamma/2)$  ou  $(|\phi_i - \phi_{k,sens}| > \beta/2)$  retorne  $l_0 = 0$
  - (e) Se  $z_t^k < z_{max}$  e  $|r - z_t^k| < \gamma/2$  retorne  $l_{oc}$
  - (f) Se  $r \leq z_t^k$  retorne  $l_{livre}$
5. Fim Se

algoritmo 7.2: Modelo inverso de um sensor a laser

Tabela 7.2: Função  $atan_2$  utilizada no algoritmo 7.2

$\Delta x$	$\Delta y$	$atan_2$
10	10	$45^\circ$
-10	10	$135^\circ$
-10	-10	$-135^\circ$
10	-10	$-45^\circ$

descrita sumariamente em [25] funcionasse corretamente.

## 7.2 Exploração com um veículo

Em [6], foi introduzida uma abordagem para exploração baseada no conceito de fronteiras (Fig.7.3) de uma grade de ocupação .

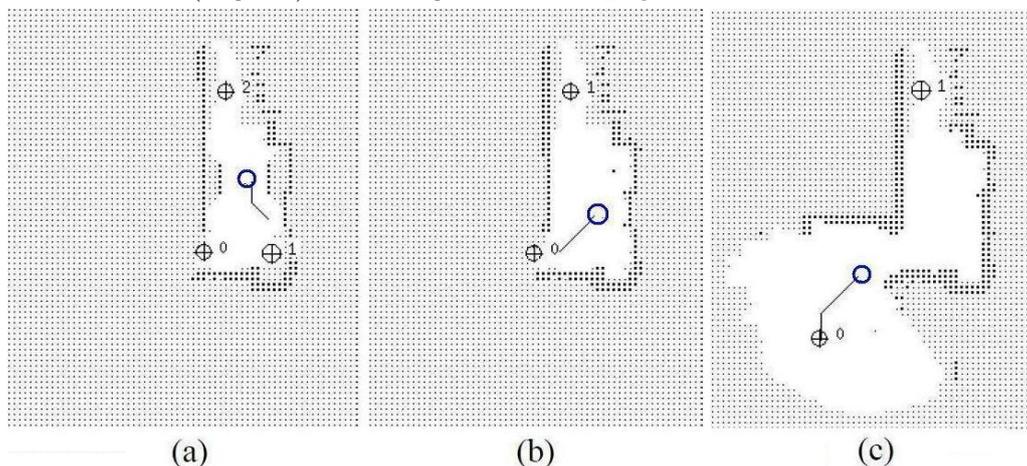


Figura 7.3: Exploração baseada em fronteiras ( $\oplus$ ) de uma grade com 01 veículo ( $\circ$ ) [6].

Cada célula de uma grade é classificada comparando sua probabilidade de ocupação  $p$  com a probabilidade de ocupação inicial ( $p_0 = 0.5$ ) atribuída a todas as células. Elas pertencem a uma das seguintes categorias: livre ( $p < p_0$ ), desconhecida ( $p = p_0$ ) ou ocupada ( $p > p_0$ ). Na figura 7.4, células livres são representadas com espaços brancos, desconhecidas com pequenos pontos e ocupadas com grandes pontos.

Qualquer célula livre adjacente a uma célula desconhecida é rotulada como uma célula-fronteira. Um conjunto de células fronteiras com tamanho acima de um certo limite mínimo é considerada uma fronteira.

A figura 7.4(a) mostra uma grade de ocupação sendo construída por um veículo (não representado) que explora o ambiente. A figura 7.4(b) mostra um conjunto de células-fronteira (borda de fronteira) detectadas na grade e a figura 7.4(c) exhibe fronteiras ( $\oplus$ ) na grade.

A idéia por detrás da exploração baseada em fronteiras de uma grade é a de que para se obter novas informações sobre o mundo real, o veículo deve se mover para o limite entre o espaço livre e um espaço ainda não mapeado (fronteira). Quando um veículo move-se para uma fronteira, ele pode observar um espaço não explorado e adicionar a nova informação para o seu mapa. Movendo-se para sucessivas fronteiras, o veículo pode constantemente aumentar seu conhecimento do ambiente onde está inserido.

Uma vez que fronteiras tenham sido detectadas dentro de uma grade

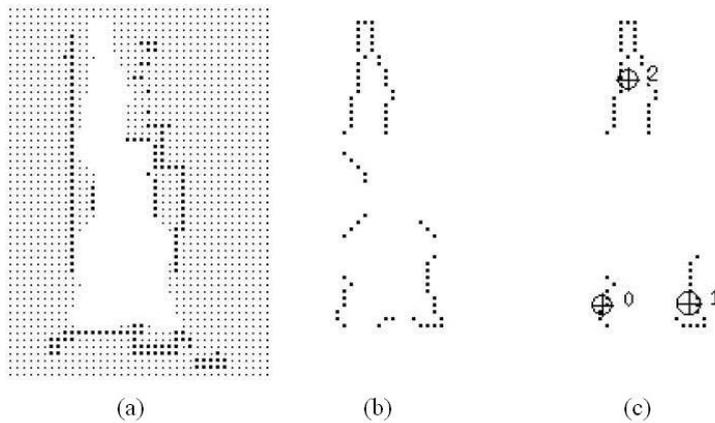


Figura 7.4: (a) grade de ocupação com células livres (branco), ocupadas (grandes pontos) e desconhecidas (pequenos pontos);(b) conjunto de células-fronteira na grade (borda de fronteira); e (c) fronteiras na grade ( $\oplus$ ) [6].

de ocupação, o veículo tenta navegar para a fronteira não visitada acessível mais próxima. O planejador do caminho usa uma busca em profundidade na grade, iniciando da célula corrente do veículo, tentando tomar o caminho mais curto livre de obstáculos para a célula contendo a localização meta. Enquanto o veículo se move em direção ao destino, métodos reativos de desvio de obstáculos podem prever a colisão com qualquer obstáculo não presente enquanto a grade de ocupação é construída. Se após um certa quantidade de tempo o veículo for incapaz de progredir em direção a seu destino, ele determinará que o destino é inacessível, e sua localização será adicionada a lista de fronteiras inacessíveis. O veículo então fará uma varredura com o sensor, atualizará a grade, e tentará navegar para a fronteira não visitada acessível mais próxima.

### 7.3 Exploração com múltiplos veículos

Em [8], o autor mostrou que a exploração baseada em fronteiras pode ser estendida para múltiplos veículos (Fig.7.5). A figura 7.6 provê uma visão geral da abordagem. Cada veículo tem sua própria grade que representa seu conhecimento sobre o ambiente (grade local). Quando o veículo chega a uma nova fronteira, ele efetua uma varredura com seu sensor e constrói uma grade de ocupação representando a corrente vizinhança. Esta grade é integrada com a grade global do veículo, e também propagada para todos os outros veículos.

Cada veículo armazena grades locais recebidas de outros veículos. Quando

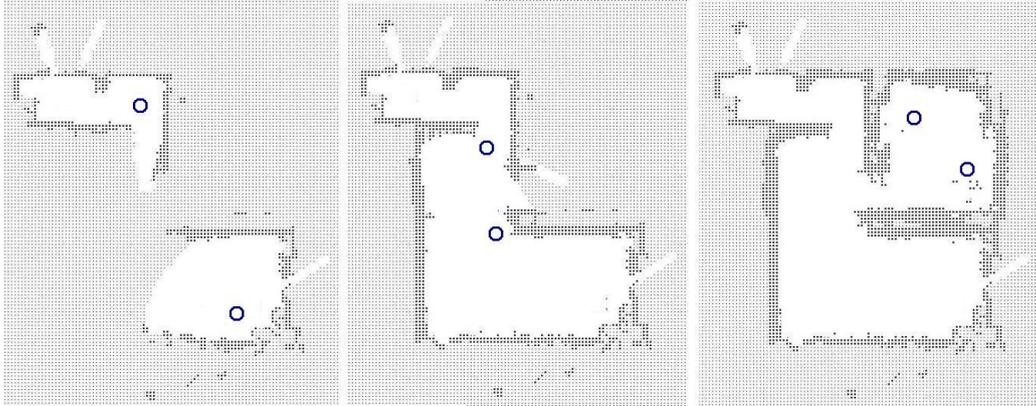


Figura 7.5: Exploração baseada em fronteiras de uma grade com 02 veículos (○) [8].

um veículo chega a uma nova fronteira, ele integra estas grades locais com o sua grade global e a nova grade local que ele construiu na fronteira.

Esta abordagem tem a vantagem de ser tanto cooperativa quanto descentralizada. As informações obtidas por um veículo são disponíveis aos outros, o que permite a um veículo usar os dados de outros veículos para determinar seu novo destino. Baseado nesta informação, um veículo pode determinar que áreas já foram exploradas por outros veículos, e então escolher explorar outra região. Um veículo também pode descobrir que uma fronteira detectada por outro veículo está próxima, e decidir investigar.

Se por um lado a informação é compartilhada, o controle é independente, o que permite a equipe ser robusta a falhas de veículos individuais. Se um veículo torna-se indisponível, ele não reportará nova informação, mas os veículos restantes continuarão a explorar o novo ambiente. Uma vez que este sistema é completamente assíncrono, os veículos nunca estão esperando pelos outros veículos, tal que a perda de um ou mais veículos nunca levará a equipe a parar.

Esta abordagem possui algumas poucas limitações. (a) uma vez que a exploração não é coordenada (ver seção 7.4), ela pode não ser eficiente pois dois ou mais veículos podem perder tempo navegando para a mesma fronteira; (b) embora a navegação reativa previna colisões, é ainda possível um veículo bloquear outro veículo e, neste caso, qualquer veículo bloqueado continuará a explorar outras fronteiras; (c) quando veículos escaneiam um ambiente, eles incorporam outros como obstáculos. No entanto, à medida que os mesmos se movem para novas posições, as antigas ficarão marcadas como espaço-livre em subseqüentes varreduras do sensor.

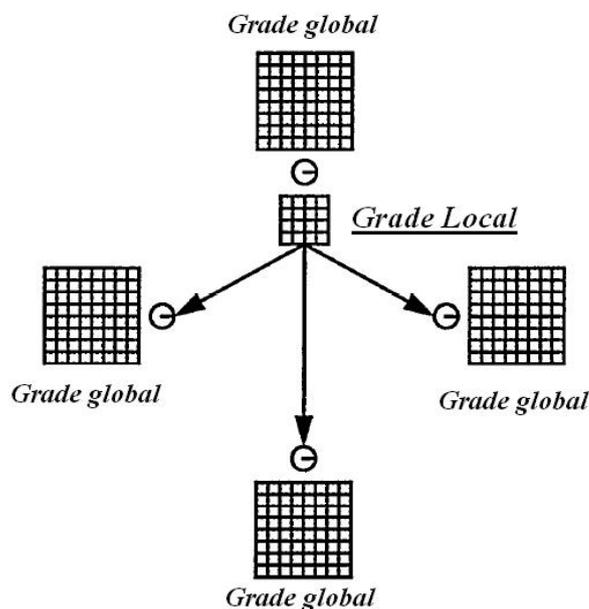


Figura 7.6: Troca de grades locais na exploração com múltiplos veículos (○) [8].

## 7.4 Exploração coordenada

Em uma atividade de exploração, o caminho a ser seguido pelo veículo deve ser controlado para ser eficiente. Na abordagem que adotamos [24] um destino - fronteira de uma grade de ocupação [17] - é escolhido de acordo com uma função que avalia o custo da navegação e a utilidade do destino.

### 7.4.1 Custo da navegação

Para determinar o custo de se alcançar as células-fronteira atuais, calcula-se o caminho ótimo da posição atual do veículo para as células-fronteira com base em um variante determinístico do *value iteration*, um conhecido algoritmo de programação dinâmica [1]. Considera-se que o custo para atravessar uma célula da grade  $(x, y)$  seja proporcional a sua probabilidade de ocupação  $P(occ_{xy})$ . O caminho de custo mínimo é calculado usando dois passos:

1. **Inicialização.** A célula da grade que contém a localização do veículo é inicializada com 0, e todas as outras com  $\infty$

$$V_{x,y} \leftarrow \begin{cases} 0, & \text{se } (x,y) \text{ é a posição do veículo} \\ \infty, & \text{caso contrário} \end{cases}$$

2. **Atualização.** Para todas as células  $(x, y)$  fazer

$$V_{x,y} \leftarrow \min\{V_{x+\Delta x, y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} P(occ_{x+\Delta x, y+\Delta y})\}$$

$$\Delta x, \Delta y \in \{0, \pm 1\} \quad P(occ_{x+\Delta x, y+\Delta y}) \in [0, occ_{max}]$$

onde  $occ_{max}$  é a probabilidade de ocupação máxima de uma célula que é permitida a um veículo atravessar. Esta técnica atualiza o valor de todas as células da grade pelo valor dos melhores vizinhos delas, mais o custo de se mover para este vizinho. Aqui, o custo é a probabilidade  $P(occ_{x,y})$  que a célula esteja ocupada vezes a distância para a célula. A regra de atualização é repetida até a convergência. Então, cada valor  $V_{x,y}$  é o custo cumulativo do veículo se mover de sua posição para  $(x, y)$ .

#### 7.4.2 Utilidade das células-fronteira

Se já existe um veículo que se move para uma determinada célula-fronteira, espera-se que a utilidade daquela célula seja menor para outros veículos.

Suponhamos que, no início, todas as células-fronteira  $t$  tenham uma utilidade  $U_t$  que é igual para todas as referidas células. Então, computamos a utilidade  $U(t_n|t_1, \dots, t_{n-1})$  de uma célula-fronteira  $t_n$  dado que as células  $t_1, \dots, t_{n-1}$  já tenha sido atribuída aos veículos  $1, \dots, n-1$  como

$$U(t_n|t_1, \dots, t_{n-1}) = U_{t_n} - \sum_{i=1}^{n-1} P(\underbrace{\|t_n - t_i\|}_d) \quad (7.8)$$

onde  $P(d)$  é a probabilidade de que o sensor de distância do veículo (sensor a laser em nosso caso) irá cobrir células a uma distância  $d$  (Eq.7.9).

De acordo com a equação 7.8, quanto mais veículos se movem para um local onde  $t_n$  seja provavelmente visível, menor é a utilidade da  $t_n$ . Computamos  $P(d)$  como

$$P(d) \leftarrow \begin{cases} 1 - \frac{d}{dist\_max}, & \text{se } d < dist\_max \\ 0, & \text{caso contrário} \end{cases} \quad (7.9)$$

onde  $dist\_max$  é a distância máxima coberta pelo sensor laser. Se existe um obstáculo entre duas células-fronteira  $t_n$  e  $t'$ , fazemos  $P(\|t_n - t'\|)$  igual a zero.

### Destino em uma Exploração Coordenada

1. Determinar o conjunto de células-fronteira.
2. Computar  $V_t^i$  para cada veículo  $i$  e célula-fronteira  $t$ .
3. Fazer  $U_t = 1$  para todas células-fronteira  $t$ .
4. Enquanto existir veículo sem destino fazer
  1. Determinar veículo  $i$  e célula-fronteira  $t$  tal que
$$(i, t) = \operatorname{argmax}_{(i^*, t^*)} (U_{t^*} - V_{t^*}^{i^*})$$
  2. Reduza  $U_{t^*}$  de cada destino  $t^*$  na área de visibilidade:
$$U_{t^*} \leftarrow U_{t^*} - P(\|t - t^*\|)$$
5. Fim Enquanto

algoritmo 7.3: Atribuição de destinos na exploração coordenada

### 7.4.3 Seleção dos destinos

Na seleção dos destinos, consideramos para cada veículo  $i$ , um balanceamento entre o custo  $V_t^i$  de se mover para um destino  $t$  e a utilidade  $U_t$  deste destino.

As figuras 7.7,7.8,7.9 e 7.10 exibem passos de geração de uma grade de ocupação utilizada para auxiliar a exploração de um ambiente em que a técnica de seleção de destinos na grade é a definida nesta seção. O caminho a ser percorrido desde a posição do veículo até o destino foi calculada de acordo com o algoritmo A estrela (A\*,[2])<sup>2</sup>.

## 7.5 Conclusão

Neste capítulo, tratamos da exploração de um ambiente interno utilizando uma grade de ocupação gerada por intermédio de um sensor laser. No próximo capítulo, são descritos os experimentos e é apresentado uma avaliação da abordagem.

---

<sup>2</sup>Algoritmo A\* é um algoritmo para Busca de Caminho. Ele busca o caminho em um grafo de um vértice inicial até um vértice final. Ele é a combinação de aproximações heurísticas como do algoritmo *Best-first Search*[57] e da formalidade do *Algoritmo de Dijkstra*[57]. O algoritmo foi descrito pela primeira vez em 1968 por Peter Hart, Nils Nilsson, e Bertram Raphael.

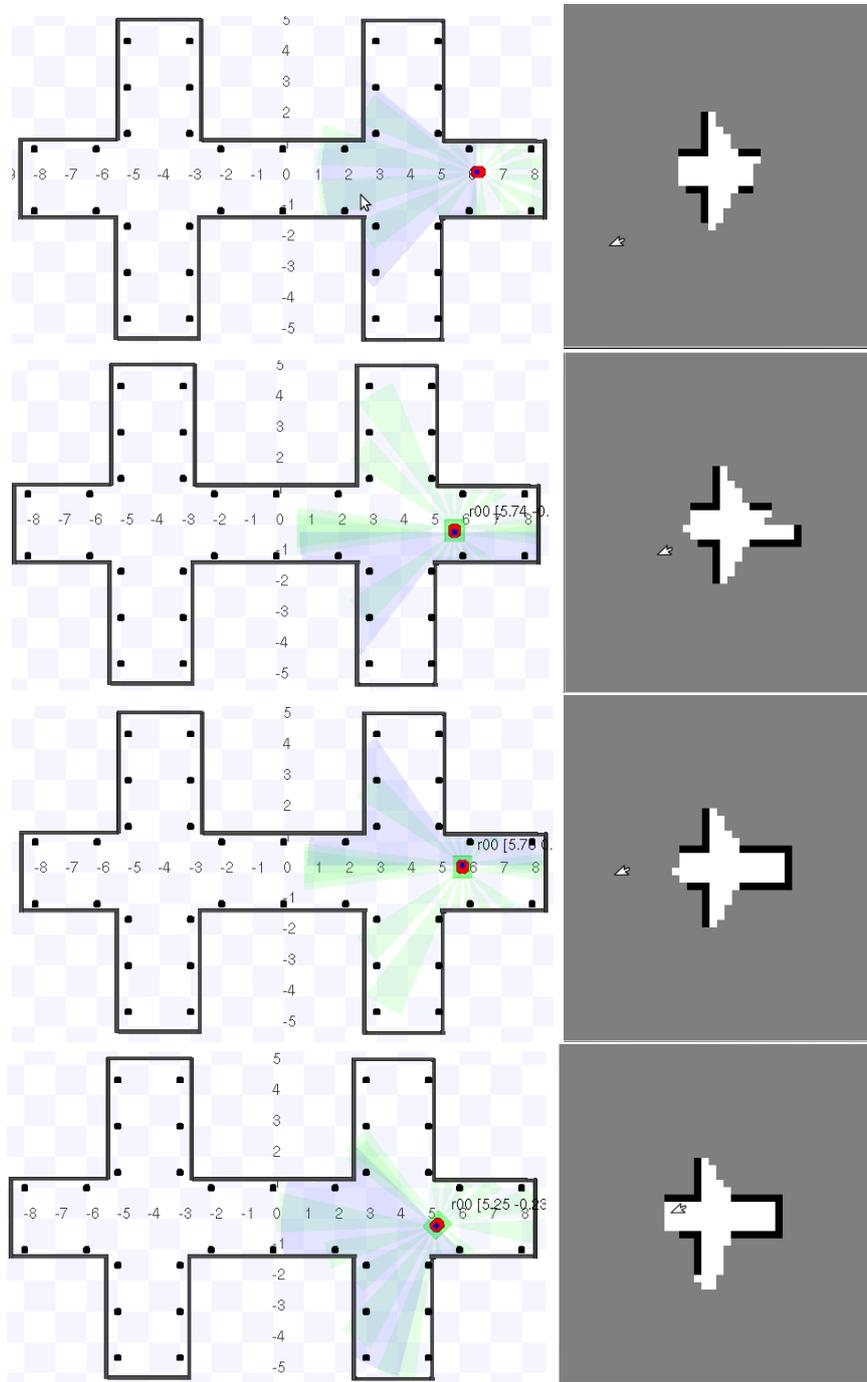


Figura 7.7: Exploração coordenada [01/04].

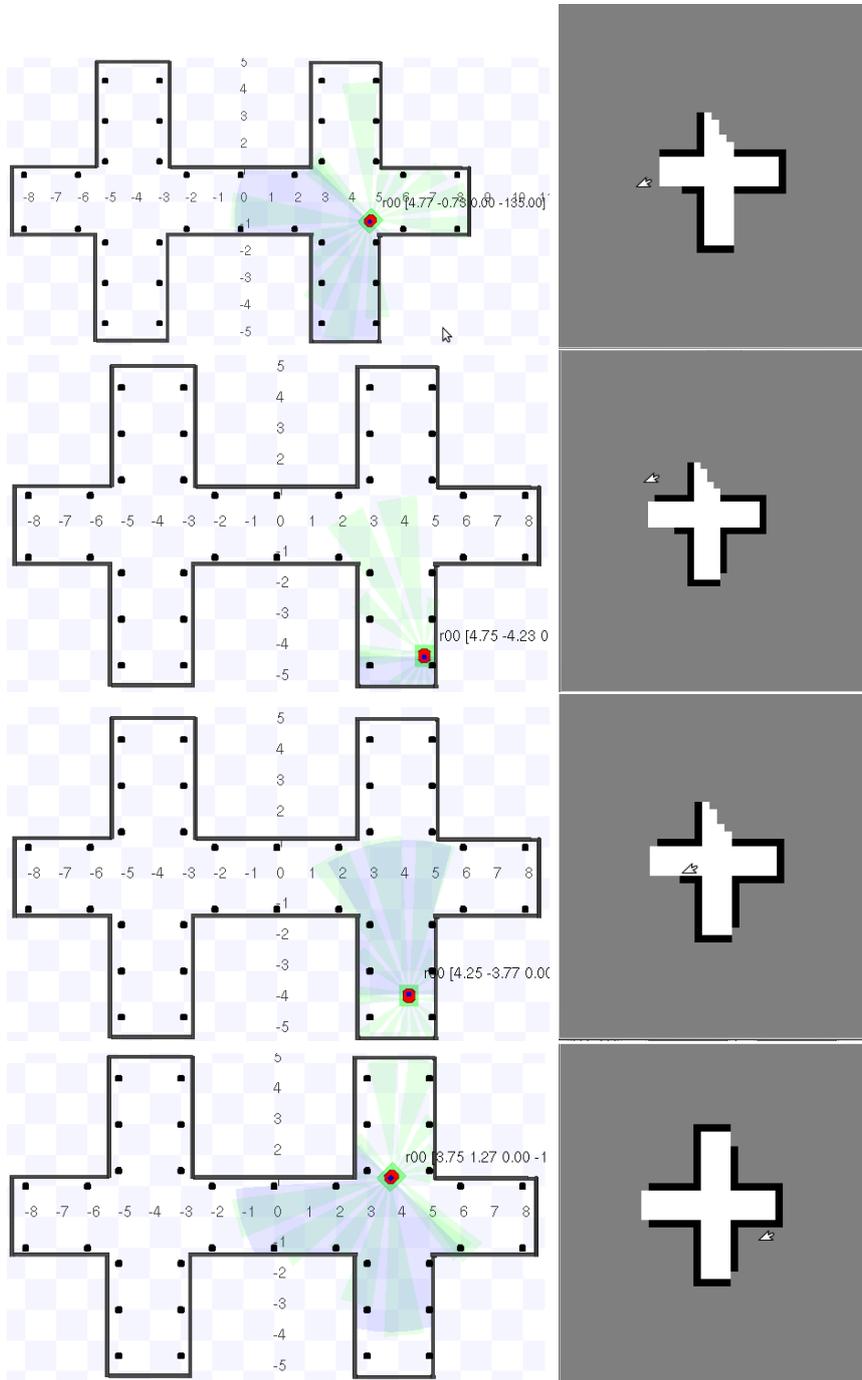


Figura 7.8: Exploração coordenada [02/04].

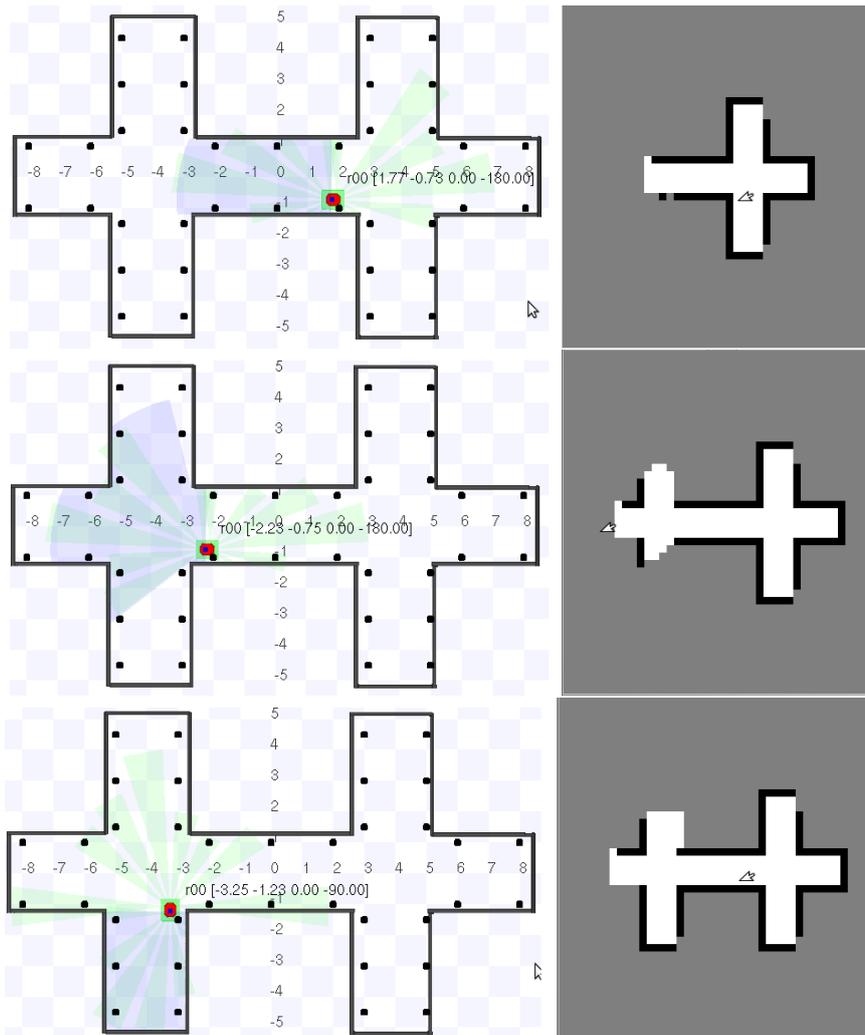


Figura 7.9: Exploração coordenada [03/04].

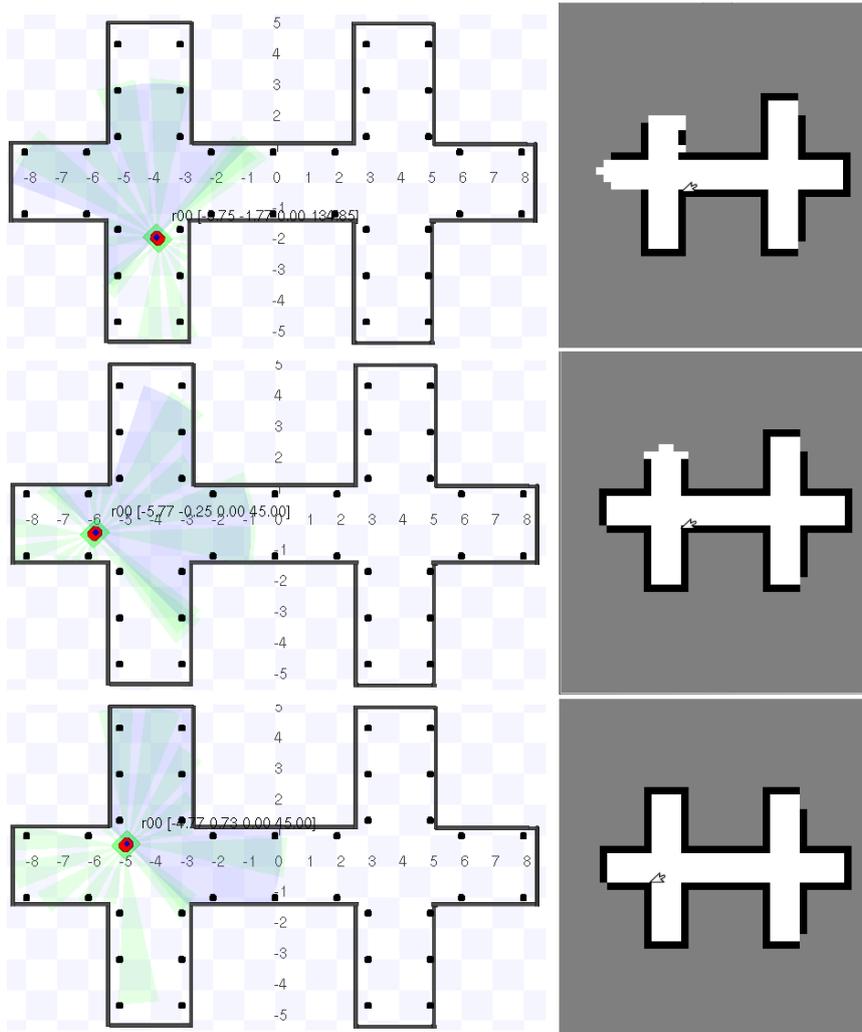


Figura 7.10: Exploração coordenada [04/04].

## CAPÍTULO 8

### Experimentos e avaliação da abordagem

Sete experimentos foram conduzidos. No primeiro, segundo e terceiro, o simulador *Player/Stage* [20] foi usado para testar toda a abordagem (Fig.8.1). No quarto, somente a associação de dados é verificada por meio de imagens obtidas com um sensor *kinect* [52], um sensor estéreo ativo, em um caminho pré-definido de um ambiente real. No quinto experimento, um Pioneer 3DX com os sensores laser (Sick LMS-200) e *kinect* é utilizado. No sexto e sétimo experimentos, a partir das imagens e dos arquivos contendo as distâncias  $[d_x, d_y, d_z]^T$  dos pixels em relação ao *kinect* (Fig.5.13), simulamos dois veículos que utilizam, respectivamente, as abordagens FastSLAM com mapa comum (Fig.8.2) e mapas dedicados. Os resultados do primeiro, segundo, terceiro, sexto e sétimo experimentos foram obtidos considerando o uso de mapas dedicados (caso “a”) e mapa comum (caso “b”).

### 8.1 Experimento 01

No primeiro experimento (Fig.8.3), o sensor laser Sick LMS-200 embarcado em cada um dos dois veículos Pioneer 3DX foi utilizado na construção da grade de ocupação. Por outro lado, para simular um sensor visual estéreo com campo de visão de  $180^\circ$ , marcos 3D representados por imagens foram usados. Cada marco contém somente uma característica SIFT e suas coordenadas  $[\theta_x, \theta_y, \theta_z]^T$  foram atribuídas artificialmente (Fig.4.1). As imagens são diferentes e, portanto, a associação de dados é conhecida e os descritores totalmente invariantes ao ponto de vista 3D da câmera (não como o descrito na figura 3.9). Nosso objetivo é mostrar que, nestas condições, a abordagem FastSLAM com mapa comum aumenta a precisão na pose esti-

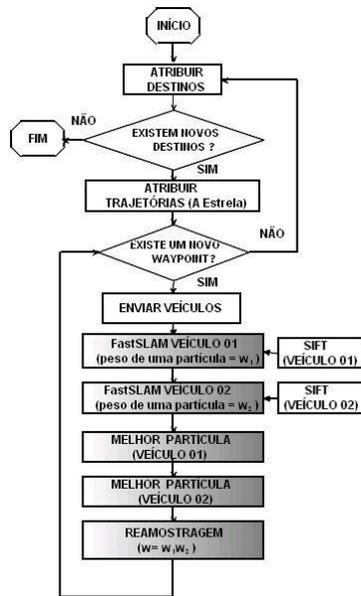


Figura 8.1: Esquema geral da abordagem. A área sombreada corresponde ao algoritmo FastSLAM com mapa comum (caso “b”).

mada dos veículo. Quando um marco visual está dentro do campo de visão do sensor visual, o descritor SIFT da imagem é extraído e a distância  $\tau$  e as orientações  $\phi_1$  e  $\phi_2$  do marco em relação ao veículo são calculadas a partir de  $[\theta_x, \theta_y, \theta_z]^T$  e da pose  $\chi_t$  do veículo (Fig.4.1).

A seleção de destinos foi feita como descrito na seção 7.4<sup>1</sup>, onde uma grade de ocupação com células de tamanho 0,5m x 0,5m foi usada. Após

<sup>1</sup>As figuras 8.5, 8.8 e 8.10 exibem caminhos percorridos por dois veículos, respectivamente, nos ambientes das figuras 8.3,8.7 e 8.9 seguindo estratégia de seleção de destinos definida na seção 7.4

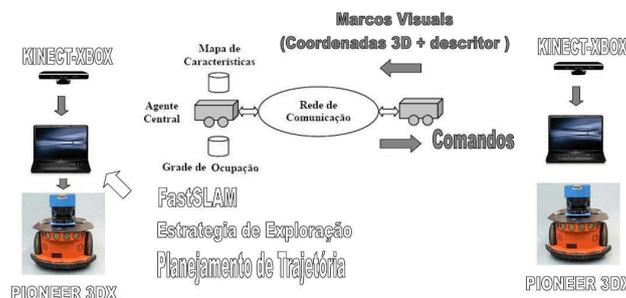


Figura 8.2: Esquema da exploração que utiliza a abordagem FastSLAM com mapa comum.

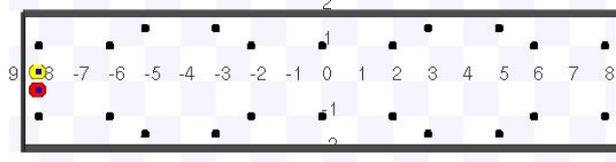


Figura 8.3: Experimento 01 - Ambiente explorado com veículos em suas poses iniciais

a seleção dos destinos, o caminho a ser percorrido pelo veículo foi calculado usando o algoritmo *A-star* [2]. A cada passo, as coordenadas ( $x$  e  $y$ ) do veículo e sua orientação ( $\varphi$ ) (Fig.8.4); a distância  $\tau$  e as orientações  $\phi_1$  e  $\phi_2$  (Fig.4.1) foram amostradas de acordo com a tabela 8.1.

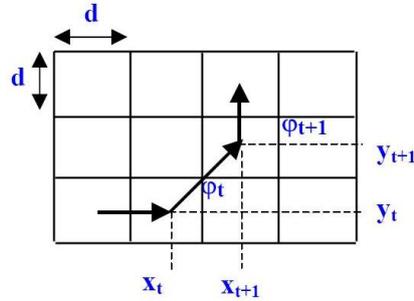


Figura 8.4: Modelo de movimento usado

Tabela 8.1: Amostragem dos modelo de movimento (Fig.8.4) e observação (Fig.4.1).  $N(0, \sigma)$  é uma distribuição normal com média zero e desvio padrão  $\sigma = 0.5$ . Valores em metros e radianos.

Variável	Erro Médio
$x_{t+1} = x_t + \Delta x_t + \bar{d}N(0, \sigma)$	$\bar{d} = \frac{5}{100}d$
$y_{t+1} = y_t + \Delta y_t + \bar{d}N(0, \sigma)$	$\bar{d} = \frac{5}{100}d$
$\varphi_{t+1} = \varphi_t + \Delta \varphi_t + \bar{g}N(0, \sigma)$	$\bar{g} = \frac{3\pi}{180} = 0,05235$
$\tau = \tau + \bar{\tau}N(0, \sigma)$	$\bar{\tau} = 0.1$
$\phi_i = \phi_i + \bar{\phi}_iN(0, \sigma) \quad i = \{1, 2\}$	$\bar{\phi}_i = \frac{\pi}{1800} = 0,00174$

No experimento, os seguintes parâmetros foram adotados:  $d = 0,5$  m,  $E_0 = 0$  e  $P_0 = 0,9$ . Como cada marco tem somente um (distinto) descritor SIFT, fizemos  $E_0 = 0$ .

A figura 8.5 mostra a trajetória percorrida pelos veículos em uma tarefa de exploração.

Os resultados da tabela 8.2 estão relacionados com a figura 8.5 considerando, respectivamente, 2000 (caso “a”) e 4000 (caso “b”) partículas e um alcance do sensor visual de 0,5 a 3,5 metros. O número de partículas

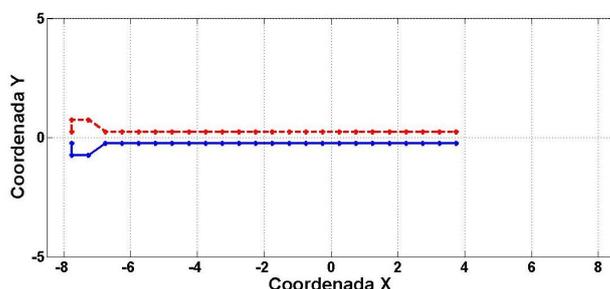


Figura 8.5: Experimento 01 - Trajetória percorrida pelos veículos em uma tarefa de exploração do ambiente descrito pela Fig.8.3: veículo 01 (linha contínua) e veículo 02 (linha tracejada). Número total de passos por veículo: 25.

necessárias para se obter uma boa estimativa cresce exponencialmente com a dimensão do estado a ser estimado [25], ou seja, quando se usa um mapa comum a dois veículos, o número de partículas tem que ser dobrado. Fizemos uma comparação adicional usando 4000 partículas nos casos “a” e “b” (Tab.8.2).

Tabela 8.2: Resultados do experimento 01 (Fig.8.5- 2000 (caso “a”) e 4000 (caso “b”) partículas)

Erro médio (RMS) nas coordenadas  $x, y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{\text{Estimativas}}{\text{Passos}}$
a	1	0,0420	0,0450	0,1335	176,6	21,2/25
a	2	0,0379	0,0374	0,0880	65,7	12,5/25
b	1	0,0331	0,0470	0,0743	493,7	22,9/25
b	2	0,0396	0,0432	0,0729	473,5	23,1/25

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0270	0,0570	0,0221	0,0680	0,0855	0,1815
a	2	0,0229	0,0529	0,0269	0,0478	0,0517	0,1242
b	1	0,0260	0,0405	0,0307	0,0633	0,0537	0,0948
b	2	0,0216	0,0577	0,0201	0,0663	0,0574	0,0883

Incremento Médio

Caso	Veículo	Incremento
b	1	39%
b	2	12%

Nas tabelas 8.2 e 8.3, verifica-se que a acurácia (Eq.8.1) e a quantidade de estimativas de localização obtidas na exploração do caso “b” (mapa comum) aumentou em relação ao caso “a” (mapas dedicados). Note que estamos usando uma condição bastante restritiva de  $P_0$  a fim de garantir uma boa

Tabela 8.3: Resultados do experimento 01 (Fig.8.5 - 4000 (caso “a” e “b”) partículas)

Erro médio (RMS) nas coordenadas  $x,y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{Estimativas}{Passos}$
a	1	0,0325	0,0465	0,1329	351,2	21,3/25
a	2	0,0381	0,0440	0,0975	132,7	13,6/25
b	1	0,0331	0,0470	0,0743	493,7	22,9/25
b	2	0,0396	0,0432	0,0729	473,5	23,1/25

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0250	0,0400	0,0292	0,0637	0,0455	0,2203
a	2	0,0237	0,0525	0,0256	0,0624	0,0664	0,1285
b	1	0,0260	0,0405	0,0307	0,0633	0,0537	0,0948
b	2	0,0216	0,0577	0,0201	0,0663	0,0574	0,0883

Incremento Médio

Caso	Veículo	Incremento
b	1	39%
b	2	21%

precisão.

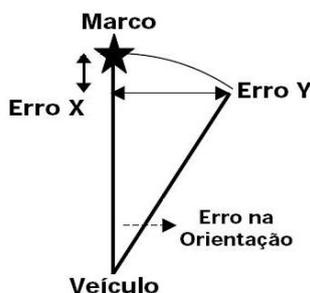


Figura 8.6: Equivalência em metros do erro de  $1^\circ = 0,01745$  rad na orientação do veículo:  $E = \int_{0,5}^{3,5} \left( \frac{A \sin(1) + (A - A \cos(1))}{3} \right) dA = 0,0352$  m, onde  $A$  é o alcance do sensor visual (0,5 a 3,5 metros),  $Erro_Y = A \sin(1)$  e  $Erro_X = A - A \cos(1)$

O incremento percentual médio foi calculado de acordo com a equação 8.1. Esta relação surge do fato de que um erro em  $\alpha$  tem um peso duas vezes maior do que um erro em  $x$  ou  $y$ <sup>2</sup> e da equivalência em metros do erro de

<sup>2</sup>Equações 4.11 e 4.12 mais a combinação do passo 2(c) e do algoritmo FastSLAM Visual (algoritmo 4.1) combinado com as equações 4.6, 4.7 e 4.8. Ver também *ambiguidade de movimento* (seção 2.5.1).

$1^\circ = 0,01745$  rad na orientação do veículo (Fig.8.6).

$$Inc = 100 * \left( 1 - \frac{x_b + y_b + 2 * \frac{\alpha_b}{0,01745} * (0,0352)}{x_a + y_a + 2 * \frac{\alpha_a}{0,01745} * (0,0352)} \right) \quad (8.1)$$

Quando um veículo está explorando um ambiente, a precisão da estimativa de sua pose tende a diminuir, uma vez que depende da precisão estimada dos marcos mapeados. Por sua vez, as coordenadas estimadas dos marcos mapeados dependem da precisão da pose estimada do veículo, que tende a diminuir à medida que o veículo explora o meio ambiente, se não ocorrerem correções através de observações de marcos anteriormente mapeados (*fechamento do loop*).

O uso do FastSLAM com mapa comum é justificado pelo fato de que um veículo pode corrigir sua pose através da reobservação de um marco mais preciso, desde que este marco pode ter sido anteriormente mapeado<sup>3</sup> pelo outro veículo (que, por sua vez, percorreu uma distância menor ou igual) (Fig.1.5).

O SLAM nos permite estimar a pose do veículo com base na estimativa dos erros da hodometria do veículo e do modelo de observação. Esta estimativa da pose do veículo é usada para corrigir o erro da hodometria, considerando que o ambiente é pequeno, já que ela depende da precisão dos marcos já detectados.

Técnicas de SLAM em grande escala devem ser utilizadas para ambientes maiores, fazendo uso de pequenos submapas [32].

## 8.2 Experimento 02

Este experimento (Fig.8.7) é similar ao experimento 01. Os seguintes parâmetros foram adotados:  $d = 0,5$  m,  $E_0 = 0$  e  $P_0 = 0,9$ .

A figura 8.8 mostra a trajetória percorrida pelos veículos na tarefa de exploração.

Os resultados das tabela 8.4 estão relacionados com a figura 8.8 considerando, respectivamente, 2000 (caso “a”) e 4000 (caso “b”) partículas e um alcance do sensor visual de 0,5 a 3,5 metros. Fizemos uma comparação adicional usando 4000 partículas nos casos “a” e “b” (Tab.8.5).

Nas tabelas 8.4 e 8.5, pode-se verificar que a acurácia (Eq.8.1) e a quantidade de estimativas de localização obtidas na exploração do caso “b” (mapa comum) aumentou em relação ao caso “a” (mapas dedicados).

---

<sup>3</sup>Mapear significa inserir ou atualizar o marco no mapa da partícula, tudo de acordo com a pose estimada do veículo.

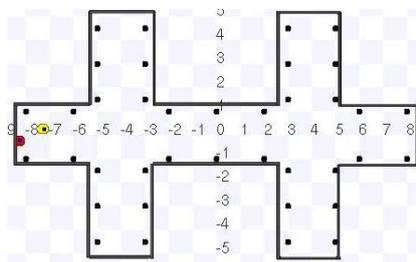


Figura 8.7: Experimento 02 - Ambiente explorado com veículos em suas poses iniciais

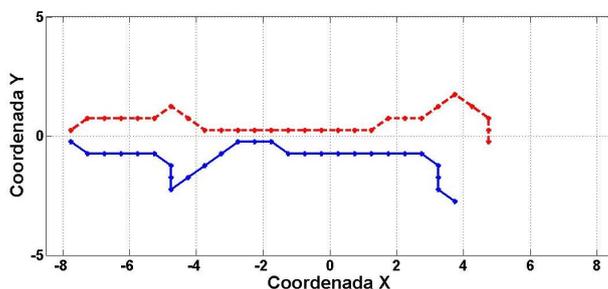


Figura 8.8: Experimento 02 - Trajetória percorrida pelos veículos em uma tarefa de exploração do ambiente descrito pela Fig.8.7: veículo 01 (linha contínua) e veículo 02 (linha tracejada). Número total de passos por veículo: 28.

### 8.3 Experimento 03

Este experimento (Fig.8.9) é similar ao experimento 01. Os seguintes parâmetros foram adotados:  $d = 0,5$  m,  $E_0 = 0$  e  $P_0 = 0,9$ .

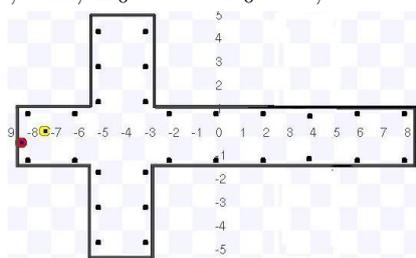


Figura 8.9: Experimento 03 - Ambiente explorado com veículos em suas poses iniciais

A figura 8.10 mostra a trajetória percorrida pelos veículos na tarefa de exploração.

Os resultados da tabela 8.6 estão relacionados com a figura 8.10 considerando, respectivamente, 2000 (caso “a”) e 4000 (caso “b”) partículas e um alcance do sensor visual de 0,5 a 3,5 metros. Fizemos uma comparação

Tabela 8.4: Resultados do experimento 02 (Fig.8.8- 2000 (caso “a”) e 4000 (caso “b”) partículas)

Erro médio (RMS) nas coordenadas  $x, y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{\text{Estimativas}}{\text{Passos}}$
a	1	0,0351	0,0358	0,0788	150	17,1/28
a	2	0,0459	0,0480	0,1094	144	21,9/28
b	1	0,0402	0,0347	0,0759	425,7	23,8/28
b	2	0,0359	0,0508	0,0849	508,7	25,4/28

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0272	0,0430	0,0275	0,0441	0,0450	0,1126
a	2	0,0248	0,0670	0,0262	0,0698	0,0613	0,1574
b	1	0,0243	0,0560	0,0254	0,0440	0,0414	0,1104
b	2	0,0252	0,0466	0,0353	0,0662	0,0563	0,1136

Incremento Médio

Caso	Veículo	Incremento
b	1	2%
b	2	19%

adicional usando 4000 partículas nos casos “a” e “b” (Tab.8.7).

Nas tabelas 8.6 e 8.7, pode-se verificar que a acurácia (Eq.8.1) e a quantidade de estimativas de localização obtidas na exploração do caso “b” (mapa comum) aumentou em relação ao caso “a” (mapas dedicados).

## 8.4 Experimento 04

No quarto experimento, o caminho (Fig.8.11) em um ambiente contendo cadeiras, mesas, computadores, etc, foi percorrido (Fig.8.12) com um *kinect* e, a cada passo, uma imagem e um arquivo contendo implicitamente as distâncias  $[d_x, d_y, d_z]^T$  dos pixels em relação ao sensor foram obtidos e armazenados para, em seguida, serem usados para simular um veículo executando um algoritmo FastSLAM.

As figuras 8.13 e 8.14 mostram a variação no número de atualizações no mapa, considerando, respectivamente, a variação nos limites  $P_0$  (Fig.8.13) e  $E_0$  (Fig.8.14). Na simulação, a distância  $\tau$  e as orientações  $\phi_1$  e  $\phi_2$  (Fig.4.1) para marcos SIFT detectados foram obtidos a partir da distância  $[d_x, d_y, d_z]^T$ . Consideramos o alcance efetivo do *kinect* (0,8 a 3,5 metros).

Podemos concluir que, respectivamente, um  $P_0$  muito alto e um  $E_0$  muito baixo podem inibir atualizações. Por outro lado, um valor de  $P_0$  muito baixo

Tabela 8.5: Resultados do experimento 02 (Fig.8.8-4000 partículas)

Erro médio (RMS) nas coordenadas  $x,y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{Estimativas}{Passos}$
a	1	0,0454	0,0508	0,0849	285,3	17,2/28
a	2	0,0445	0,0441	0,0943	293,8	22,2/28
b	1	0,0402	0,0347	0,0759	425,7	23,8/28
b	2	0,0359	0,0508	0,0849	508,7	25,4/28

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0309	0,0600	0,0320	0,0696	0,0414	0,1283
a	2	0,0267	0,0623	0,0277	0,0605	0,0227	0,1658
b	1	0,0243	0,0560	0,0254	0,0440	0,0414	0,1104
b	2	0,0252	0,0466	0,0353	0,0662	0,0563	0,1136

Incremento Médio

Caso	Veículo	Incremento
b	1	13%
b	2	9%

com um valor de  $E_0$  muito alto podem degradar o mapa e o algoritmo Fast-SLAM. O ideal é um alto valor de  $P_0$  e um valor de  $E_0$  suficientemente baixo. Desta forma, somente marcos próximos da posição prevista pela partícula ( $P_0$  alto) e muitos parecidos ( $E_0$  baixo) podem ser associados.

A fim de definir o valor de  $E_0$ , fizemos uma análise em separado dos descritores SIFT das imagens independentemente do alcance efetivo do *kinect*.

Obtivemos os descritores de todas as 11 imagens (3125 descritores) e calculamos, para cada descritor, a distância de Mahalanobis para os outros descritores. Constatamos que os descritores são distintos, ou seja, todas as distância são diferentes de 0. A distância de Mahalanobis média é igual a 1,2. A tabela 8.8 mostra o percentual de distâncias entre os descritores de todas as imagens que estão abaixo de certos limites.

Fizemos uma análise da correspondência (associação) dos descritores de duas imagens (Fig.8.15). Dos resultados obtidos e consolidados na tabela 8.9, constatamos que os valores  $E_0 = 0,4$  e  $E_0 = 0,5$  proporcionam uma boa combinação no que se refere ao número de correspondências corretas (grande) e ao número de possíveis correspondências (baixo). Para a obtenção dos dados da referida tabela, os cálculos foram feitos considerando que existem características SIFT correspondentes nas duas imagens (Fig.8.15).

Entretanto, no FastSLAM a correspondência (associação) é feita entre um marco observado de uma imagem e os marcos do mapa de características



Tabela 8.6: Resultados do experimento 03 (Fig.8.10- 2000 (caso “a”) e 4000 (caso “b”) partículas)

Erro médio (RMS) nas coordenadas  $x,y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{Estimativas}{Passos}$
a	1	0,0299	0,0335	0,0877	122,5	16,8/27
a	2	0,0421	0,0317	0,0895	139,1	22,4/27
b	1	0,0422	0,0468	0,0675	467,3	24,4/27
b	2	0,0404	0,0560	0,0806	511,1	24,5/27

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0184	0,0414	0,0207	0,0462	0,0457	0,1296
a	2	0,0251	0,0590	0,0221	0,0413	0,0473	0,1317
b	1	0,0290	0,0554	0,0331	0,0606	0,0418	0,0933
b	2	0,0239	0,0568	0,0344	0,0776	0,0495	0,1116

Incremento Médio

Caso	Veículo	Incremento
b	1	14%
b	2	4%

que esperamos 10 segundos para o deslocamento do veículo, o tempo total de processamento do FastSLAM foi de 346 s (21 s/passos). Caso não usássemos a estrutura *kd-tree*, o tempo cresceria linearmente com o número total de marcos. Constatamos (Fig.8.20) que em 40% dos passos, foi obtida uma estimativa de localização. A figura 8.21 mostra o número de marcos processados a cada passo. Na figura 8.22, temos a grade em maior resolução - *mapa a laser*- considerando em cada passo a melhor partícula.

A fim de avaliar o impacto da escolha de  $E_0$  no FastSLAM, a partir das imagens e do arquivo contendo implicitamente as distâncias  $[d_x, d_y, d_z]^T$  dos pixels em relação ao sensor visual, que foram armazenados durante o experi-



Figura 8.12: Experimento 04 - Imagem do laboratório mencionado na Figura 8.11 referente, respectivamente, a pose simulada (1,0,90) do veículo.

Tabela 8.7: Resultados do experimento 03 (Fig.8.10-4000 partículas)

Erro médio (RMS) nas coordenadas  $x,y$  e orientação  $\alpha$ ; número médio de atualizações de marcos e de estimativas de localizações

Caso	Veículo	$x$	$y$	$\alpha$	Atualizações	$\frac{Estimativas}{Passos}$
a	1	0,0492	0,0387	0,0882	269,6	18/27
a	2	0,0470	0,0409	0,1067	268,7	22,5/27
b	1	0,0422	0,0468	0,0675	467,3	24,4/27
b	2	0,0404	0,0560	0,0806	511,1	24,5/27

Valores máximos e mínimos do Erro médio (média  $\pm$  desvio padrão)

Caso	Veículo	$x_{min}$	$x_{max}$	$y_{min}$	$y_{max}$	$\alpha_{min}$	$\alpha_{max}$
a	1	0,0276	0,0707	0,0270	0,0505	0,0659	0,1104
a	2	0,0250	0,0689	0,0211	0,0607	0,0296	0,1838
b	1	0,0290	0,0554	0,0331	0,0606	0,0418	0,0933
b	2	0,0239	0,0568	0,0344	0,0776	0,0495	0,1116

Incremento Médio

Caso	Veículo	Incremento
b	1	19%
b	2	19%

mento, obtivemos a quantidade de estimativas de localização, a quantidade de atualizações de marcos e o tempo de processamento (Tab.8.10). Constatamos que embora  $E_0$  tenha variado 75%, o tempo de processamento variou apenas 41%.

## 8.6 Experimento 06

No sexto experimento, a partir das imagens e dos arquivos contendo as distâncias  $[d_x, d_y, d_z]^T$  dos pixels em relação ao *kinect* do experimento 5 (Fig.5.13), simulamos dois veículos percorrendo os caminhos descritos na figura 8.23 e

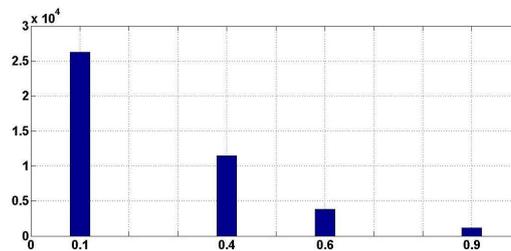


Figura 8.13: Experimento 04 - Número médio de atualizações no mapa em função de  $P_0$  ( $E_0 = 1, 2$  e 1000 partículas).

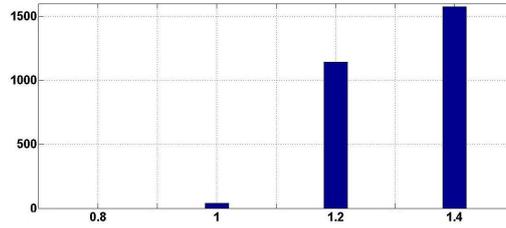


Figura 8.14: Experimento 04 - Número médio de atualizações no mapa em função de  $E_0$  ( $P_0 = 0,9$  e 1000 partículas).

Tabela 8.8: Experimento 04 - análise dos descritores 1

Distância-Limite ( $E_0$ )	Quantidade de distâncias	Percentual(%)
0,1	0	0
0,3	252	0,003
0,5	6362	0,065
0,7	53588	0,550
0,9	298898	3,061
1,1	1559184	15,971
1,2	3319498	34,002

utilizando FastSLAM com mapas dedicados (caso “a”) e mapa comum (caso “b”) e considerando o alcance efetivo do *kinect* e 1000 partículas.

Este experimento considerou duas situações em relação à abordagem FastSLAM com mapa comum (caso “b”). Na primeira (casos “b1”), quando uma partícula (devido a uma observação do veículo 1 ou do veículo 2) não consegue atualizar (associar) um marco previamente observado, ele é inserido no mapa de características da partícula. Na segunda (casos “b2”), uma partícula somente vai inserir um novo marco em seu mapa de características devido a uma observação de um determinado veículo, se os marcos da região não

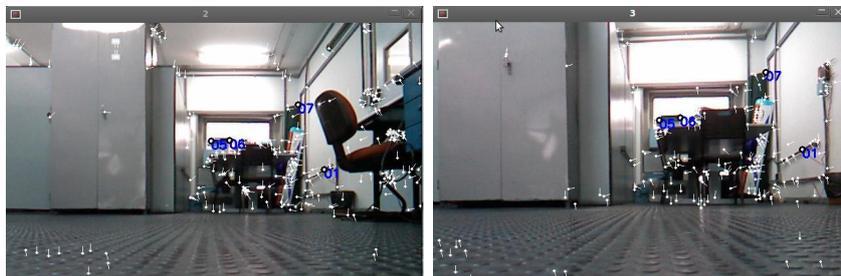


Figura 8.15: Experimento 04 - Correspondência das características SIFT das imagens (setas brancas) colhidas nos passos 2 e 3 (Fig.8.11) considerando  $E_0 = 0,4$ .

Tabela 8.9: Experimento 04 - análise dos descritores 2.

$E_0$	$\frac{Associa\c{o}es\_corretas}{Total\_de\_associa\c{o}es}$
0,4	04/04 = 100%
0,5	13/14 = 93%
0,6	22/25 = 88%
0,7	31/42 = 74%



Figura 8.16: Experimentos 05 e 06 - Imagens referentes as poses (4;3,5;180) e (0;0;0) do veículo (Fig.8.18).

tiverem sido mapeados pelas observações do outro veículo (as observações do veículo 2 redundarão em inserção de marcos no mapa das partículas, apenas nos passos 1 e 2). Em uma tarefa de exploração, o caso “b2” pode ser implementado.

Os resultados (Tabs.8.11 e 8.24) mostram que, no casos “a” (veículo 1 e 2), o número de estimativas de localização obtidos foi reduzido. A explicação para este fato são elencadas a seguir.

O sensor *kinect* estava apontado para direção de movimento do veículo. Nesta configuração, a medida que o veículo desloca-se, a possibilidade de reobservar um marco anteriormente observado localizado na direção frontal ao movimento do veículo reduz (Figs.8.16 e 8.25(a)), pois o campo visual do



Figura 8.17: Experimento 05 - Pioneer 3DX com sensores laser (LMS-200) e *kinect*.

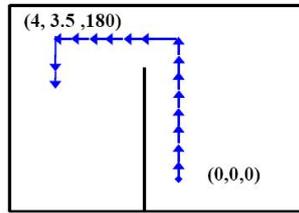


Figura 8.18: Experimento 05 - Caminho (16 passos) percorrido pelo veículo Pioneer 3DX no ambiente da figura 8.16. Cada passo tem 0,5 ou 1,0 metros.

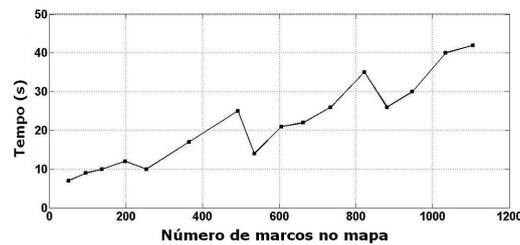


Figura 8.19: Experimento 05 - Tempo de processamento em função do número de marcos no mapa a cada passo ( $P_0 = 0,9$  e  $E_0 = 0,5$ ).

*kinect* é  $58^\circ$  (horizontal) e  $45^\circ$  (vertical). Além disso, em relação a marcos visuais na lateral do veículo (Figs.8.16 e 8.25(b)), devido ao campo visual do *kinect* e a invariância parcial ao ponto de vista 3D do descritor SIFT (Fig.3.9), apenas marcos distantes do veículo tem a possibilidade de serem reobservados após os respectivos mapeamentos em um passo anterior. O alcance efetivo do *kinect* é de 0,8 a 3,5 metros.

No caso “b2” (veículo 1), o número de estimativas de localização continuou reduzido, devido as razões anteriormente elencadas.

No caso “b2” (o veículo 2) o número de estimativas de localização e de atualizações (Tab.8.11 e Fig.8.24) aumentou. Isto implica que ocorreram atualizações de marcos previamente mapeados pelo veículo 1.

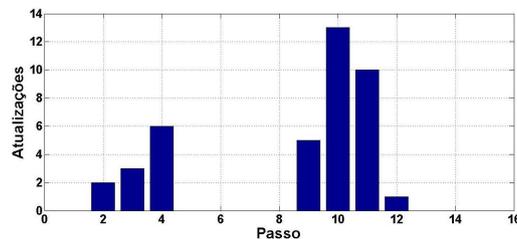


Figura 8.20: Experimento 05 - Atualizações em cada passo ( $P_0 = 0,9$  e  $E_0 = 0,5$ ).

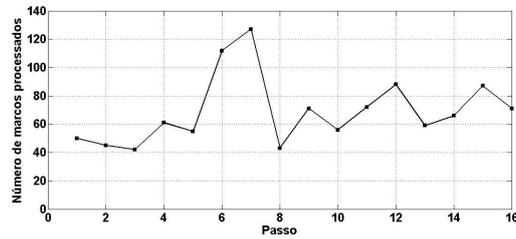


Figura 8.21: Experimento 05 - Número de marcos processados por passo.

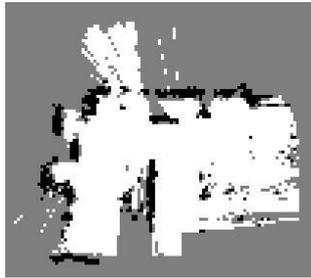


Figura 8.22: Experimento 05 - Grade em maior resolução - *mapa a laser* - gerado pela melhor partícula ( $P_0 = 0,9$  e  $E_0 = 0,5$ ).

O número de estimativas e o número de atualizações cresceram muito devido ao fato do veículo 2 reobservar marcos anteriormente mapeados pelas observações do veículo 1.

No caso “b2”, o veículo 2 percorreu a mesma trajetória do veículo 1, e portanto, esteve em uma condição privilegiada, pois observou sob um mesmo ponto de vista, um marco anteriormente mapeado pelo veículo 1. O problema da invariância parcial do descritor SIFT ao ponto de vista 3D da câmera teve pouca influência (Fig.3.9). Contudo não há uma garantia de que isso ocorra. Inserir um marco quando o mesmo não seja atualizado (caso “b1”) é a melhor escolha a fim de possibilitar que o veículo consiga uma associação de dados

Tabela 8.10: Experimento 05: quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de  $E_0$

$E_0$	$\frac{\text{Estimativas}}{\text{Passos}}$	Atualizações	Tempo (seg)	Tempo (min)
0,4	3/16	3	323	5,38
0,5	7/16	40	346	5,76
0,6	9/16	1135	388	6,46
0,7	10/16	200	458	7,63

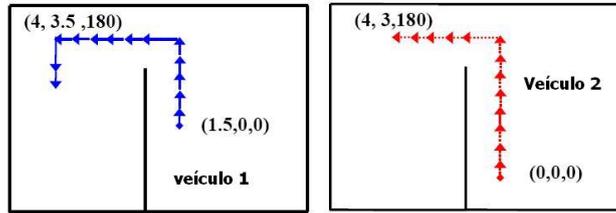


Figura 8.23: Experimento 06 - Caminho (13 passos) percorrido - de forma simulada - pelos veículos 01 e 02 : cada passo tem 0,5 ou 1,0 metros.

Tabela 8.11: Experimento 06 - Quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de  $E_0 = 0,5$ .

Caso	Veículo	$\frac{Estimativas}{Passos}$	Atualizações	Tempo (min)
a	1	05/13	38	4,30
a	2	07/13	31	3,85
b1	1	04/13	38	16,61
b1	1	12/13	388	16,61
b2	1	06/13	41	10,68
b2	2	12/13	760	10,68

em um passo posterior (Fig.8.25(a) e Fig.8.25(b)).

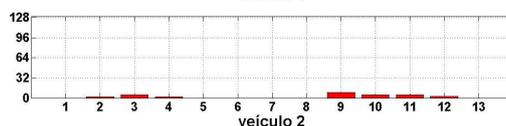
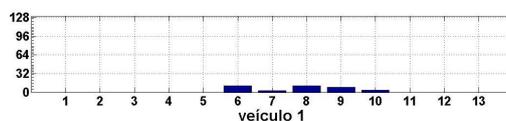
A figura 8.26 mostra os mapas de marcos gerados pela melhor partícula. Nos casos “b1” e “b2”, cada partícula que atualizou um marco, o fez em apenas alguns deles ( $P_0 > 0,9$ ). No caso “b1”, o número de marcos foi praticamente igual a soma do número de marcos do caso “a” ( $N_{b1} \cong N_{a(1)} + N_{a(2)}$ ). Por outro lado, no caso “b2”,  $N_{b2} \cong 0,57 * (N_{a(1)} + N_{a(2)})$ .

No caso “b1”, constatamos que a pesquisa pelos vizinhos mais próximos na árvore *kd-tree* de uma partícula selecionou 2,52 vezes mais marcos em relação ao caso “a”. No caso “b2”, pesquisa dos vizinhos mais próximos selecionou 1,98 vezes mais marcos.

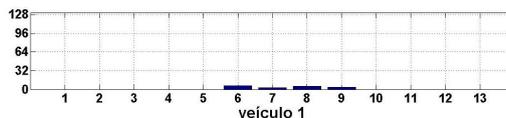
Os valores no tempo de processamento (tab.8.11) foram devidos ao número de marcos selecionados que é função do número de marcos no mapa.

## 8.7 Experimento 07

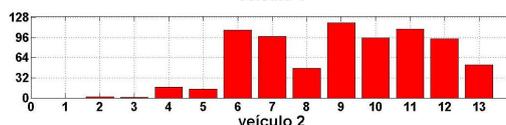
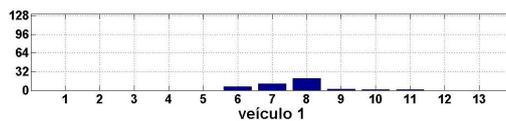
No sexto experimento, a partir de imagens e dos arquivos contendo as distâncias  $[d_x, d_y, d_z]^T$  dos pixels em relação ao *kinect*, simulamos dois veículos percorrendo os caminhos descritos na figura 8.27 e utilizando FastSLAM com mapas dedicados (caso “a”) e mapa comum (caso “b”) e considerando



Caso “a” - Mapas dedicados



Caso “b1” - Mapa comum



Caso “b2” - Mapa comum

Figura 8.24: Experimento 06 - Número médio de atualizações de marcos por passo ( $E_0 = 0,5$  e  $P_0 = 0,9$ ).

o alcance efetivo do *kinect* e 1000 partículas.

O sensor visual estava apontado para direção perpendicular ao movimento do veículo (para uma parede lateral).

Pelos resultados obtidos, de maneira diversa ao ocorrido no experimento 06, ambos os veículos, nos casos “a”, “b1” e “b2”, obtiveram uma quantidade equivalente de estimativas por passo (Tabs.8.12 e 8.28). A quantidade des estimativas também foi boa. Isto ocorreu pela direção em que estava apontado a câmera<sup>4</sup>.

Pela figura 8.29(a), percebemos que quanto maior a distância da parede, maior a probabilidade de que o sensor visual obtenha marcos visuais que

<sup>4</sup>Neste experimento, e no experimento 06, dois sensores visuais (um apontando para cada lado) poderiam ser utilizados.

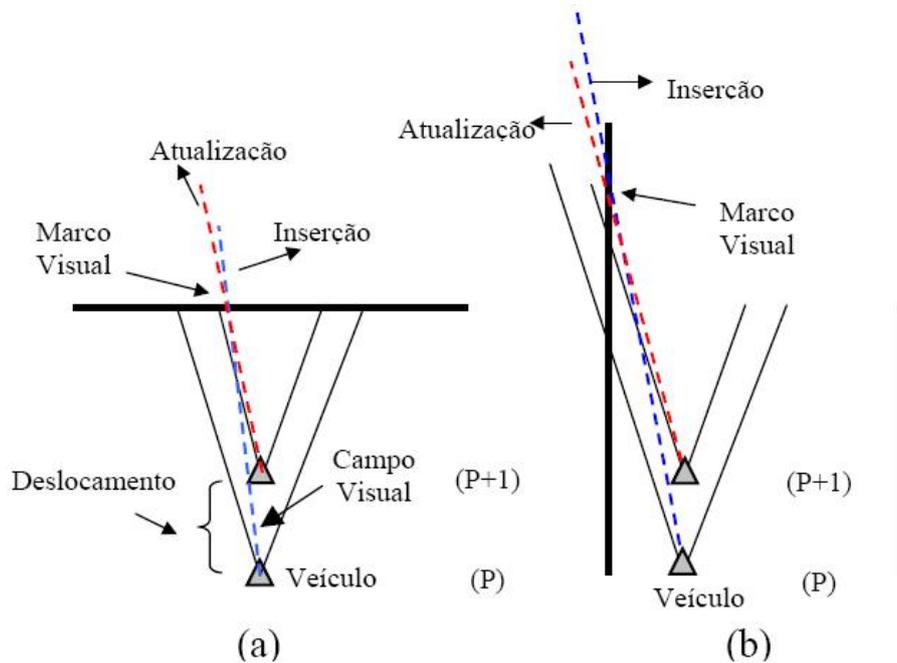


Figura 8.25: Experimento 06: Percepção de marcos visuais.

possibilite, em um passo posterior, a obtenção de uma associação de dados (Fig.8.29(b)).

No caso “b2”, o veículo 2 conseguiu atualizar marcos anteriormente mapeados pelo veículo 1, desde que ele reobservou-os a partir de um ponto de vista 3D similar (Fig.3.9). Contudo não há uma garantia de que isso ocorra. Inserir um marco quando o mesmo não seja atualizado (caso “b1”) é a melhor escolha a fim de possibilitar que o veículo consiga uma associação de dados em um passo posterior (Fig.8.29(b)).

Tabela 8.12: Experimento 07 - Quantidade de estimativas de localização obtidas dividido pelo número de passos; quantidade de atualizações de marcos e tempo de processamento em função de  $E_0 = 0,5$ .

Caso	Veículo	$\frac{Estimativas}{Passos}$	Atualizações
a	1	06/08	95
a	2	06/08	79
b1	1	06/08	92
b1	2	07/08	185
b2	1	06/08	144
b2	2	07/08	76

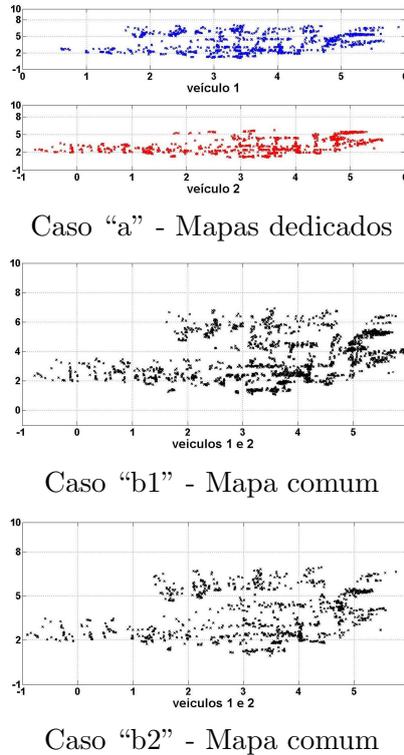


Figura 8.26: Experimento 06 - Mapa 2D dos marcos gerados pela melhor partícula ( $E_0 = 0,5$  e  $P_0 = 0,9$ )

## 8.8 Avaliação da Abordagem

Nos experimentos 01,02 e 03 (seção 8.1,8.2 e 8.3) a associação de dados é conhecida e os descritores totalmente invariantes ao ponto de vista 3D da câmera (não como o descrito na figura 3.9). Nosso objetivo foi mostrar que, nestas condições, a abordagem FastSLAM com mapa comum aumenta a precisão na pose estimada dos veículo. Nestes experimentos, quando um veículo observa um marco no ambiente, ele tem condições de encontrar o marco correspondente no mapa de características da partícula com base na distância entre os descritores. O marco visual no mapa da partícula com distância de Mahalanobis igual a zero corresponde a associação correta.

Os resultados dos experimentos 01,02 e 03 mostram que o uso da abordagem FastSLAM com mapa comum aumentou a acurácia na pose dos veículos, bem como o número de estimativas de localização obtidas pelos veículos ao longo de seus respectivos percursos.

O uso do FastSLAM com mapa comum é justificado pelo fato de que um veículo pode corrigir sua pose através da reobservação de um marco mais

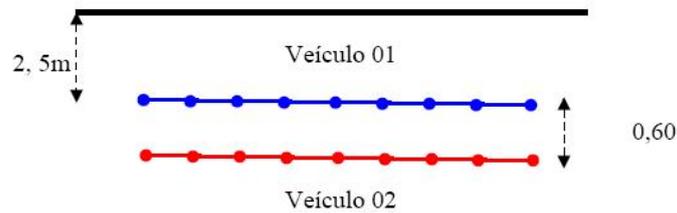


Figura 8.27: Experimento 07 - Caminho (8 passos) percorrido - de forma simulada - pelos veículos 01 e 02 : cada passo tem 0,5 metro.

preciso (*fechamento do loop*), desde que este marco pode ter sido anteriormente mapeado pelo outro veículo (que percorreu uma distância menor ou igual) (Fig.1.5).

A melhoria na acurácia do experimento 01 foi mais expressiva. Isto ocorreu por que a reobservação de marcos anteriormente mapeados (*fechamento do loop*) pelo outro veículo ocorreu em uma frequência maior.

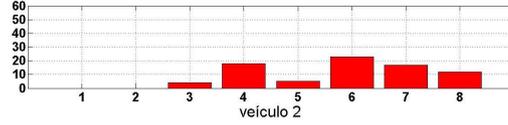
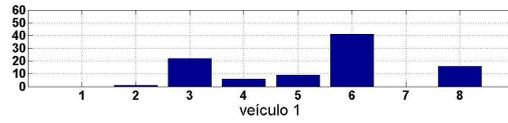
Nos experimentos 02 e 03, os veículos percorreram em alguns passos trajetórias distintas. Nestes passos, um veículo não reobservou marcos anteriormente mapeados pelo outro veículo, apenas reobservou marcos mapeados por ele mesmo. Quando os veículos retornaram ao corredor central, passaram a corrigir as respectivas poses com base em marcos anteriormente mapeados pelo outro veículo.

Caso um dos veículos ficasse bem à frente do outro (Fig.8.30), o veículo que viesse a retaguarda poderia reobservar um marco mapeado pelo outro veículo que percorreu uma distância menor. O *fechamento do loop* ocorreria com um marco mais preciso.

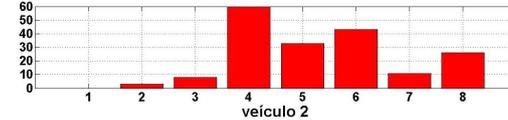
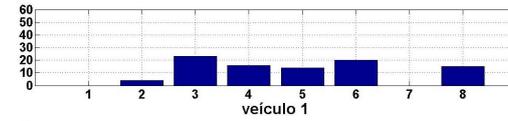
Quando se considera que o descritor SIFT tenha invariância total ao ponto de vista 3D (experimentos 01,02 e 03), para melhoria da acurácia, um veículo tem que passar por uma área onde o outro já tenha mapeado (ou que está mapeando) (corredor central da figura 8.30).

Em um experimento real (experimentos 05, 06 e 07), a associação de dados é desconhecida e os descritores são parcialmente invariantes ao ponto de vista 3D da câmera (Fig.3.9). Ela é baseada na probabilidade de associação do marco visual observado no ambiente e dos marcos visuais no mapa e na distância Euclidiana do descritor do marco observado e dos descritores dos marcos no mapa (Algoritmo 2.1).

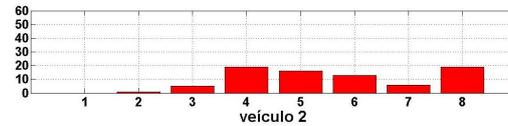
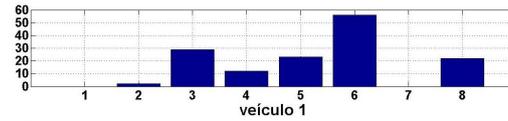
Quando a invariância parcial ao ponto de vista 3D é considerada (experimentos 05, 06 e 07), para melhoria da acurácia, um veículo tem que passar por uma área onde o outro já tenha mapeado (ou que está mapeando) e reobservar marcos mapeados pelo outro veículo de um ponto de vista similar (corredor central da figura 8.30).



Caso “a” - Mapas dedicados



Caso “b1” - Mapa comum



Caso “b2” - Mapa comum

Figura 8.28: Experimento 07 - Número médio de atualizações de marcos por passo ( $E_0 = 0,5$  e  $P_0 = 0,9$ ).

Estamos considerando que existem marcos visuais detectáveis no ambiente e que os veículos, pela sua disposição no ambiente, consigam obter estimativas de localização ao longo de suas trajetórias.

Na figura 8.31(a) o veículo 1 inseriu um marco e no passo posterior conseguiu atualizá-lo pois reobservou-o sob um ponto de vista similar. Na seqüência, o veículo 2 consegue novamente atualizá-lo. Na figura 8.31(b) o veículo 1 inseriu um marco, mas no passo posterior não conseguiu atualizá-lo pois reobservou-o sob um ponto de vista muito diverso. Na seqüência, o veículo 2 consegue atualizá-lo. Se não conseguisse, inseriria no mapa da partícula esse marco, que teria um descritor compatível com o ponto de vista 3D da observação (caso “b2”).

Na abordagem FastSLAM com um mapa comum aos veículos, todo processamento fica concentrado no agente central (outro veículo - Fig. 6.1 - ou

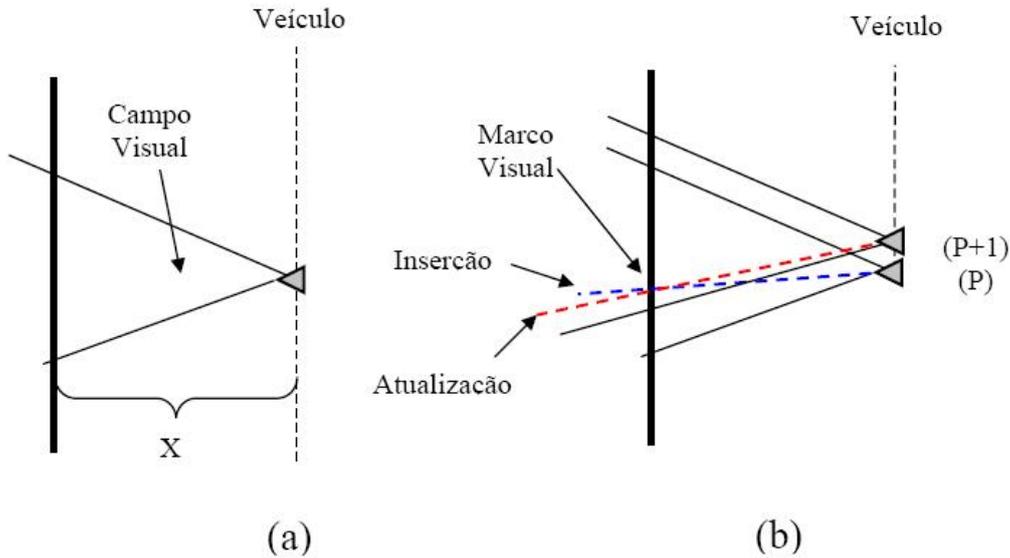


Figura 8.29: Experimento 07: Percepção de marcos visuais.

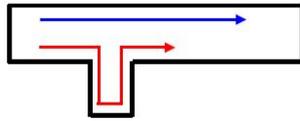


Figura 8.30: Avaliação da abordagem: trajetórias dos veículos.

uma estação gerenciadora). Ela possui um custo de  $2M_b \log N_b$  ( $M_b = 2M_a$  e  $N_b > N_a$ ). Embora haja um aumento do custo computacional, a ordem de complexidade da abordagem permanece  $O(M \log N_b)$  e os marcos estão armazenados em uma estrutura *kd-tree*. O aumento no custo computacional e no número duplicado de partículas pode ser compensado pela paralelização e pelo aumento do poder computacional do agente central (equivalente ao número de veículos utilizados).

O processamento dos marcos observados pelos veículos pode ser paralelizado, uma vez que uma partícula representa uma estimativa das poses dos veículos com seu respectivo conjunto de marcos estimados: enquanto o processo 1 computa a observação do veículo 1 nas  $M/2$  primeiras partículas, o processo 2 computa a observação do veículo 2 nas  $M/2$  partículas restantes.

O número de marcos processados na abordagem com mapa comum ( $N_b$ ) cresce somente com a área comum percorrida pelos veículos. Tanto na abor-

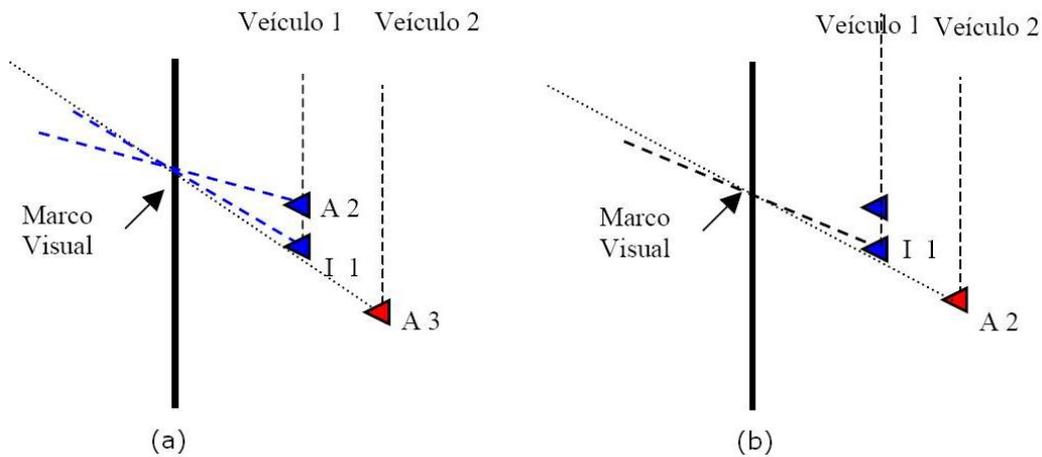


Figura 8.31: Avaliação da abordagem: Percepção de marcos visuais.

dagem com mapa comum, quanto na com mapas dedicados, quando um veículo explora sozinho uma área (Fig.8.30), ele terá que mapear marcos observados e não atualizados, tanto na ida, quanto na volta. Isto ocorre devido a invariância parcial do descritor SIFT.

Em relação à complexidade na extração de marcos com o algoritmo SIFT, consideramos que ela é feita individualmente em cada veículo. Embora tenhamos utilizado o SIFT, o SURF é mais rápido e possui uma eficiência maior (seção 3.6).

No que se refere à necessidade de comunicação para a transmissão dos descritores para o agente central, cujo custo não foi analisado, temos que a referida comunicação também é exigida para transmissão das atualizações da grade de ocupação.

Na tabela 8.13 temos um comparativo das abordagens de exploração integrada.

## 8.9 Conclusão

Neste capítulo, foram descritos os experimentos e apresentado uma avaliação da abordagem. Nos capítulos 9 e 9.2, apresentamos, respectivamente, nossas conclusões e sugestões de trabalhos futuros.

Tabela 8.13: Comparação das abordagens de Exploração Integrada.

Item	Caso “a” (Mapas dedicados)	Caso “b” (Mapa comum)
Complexidade	$M_a \text{Log} N_a$	$M_b \text{Log} N_b$ ( $N_b \geq N_a$ ) ( $M_b = 2M_a$ ) $N_b$ cresce com o aumento do tamanho do caminho comum aos veículos
Comunicação (transmissão da grade)	sim	sim
Comunicação (transmissão de descritores)	não	sim
<i>Fechamento do Loop</i>	próprios marcos mapeados ao longo do caminho	próprios marcos mapeados ao longo do caminho e marcos mapeados pelo outro veículo em qualquer local

Neste trabalho, foi investigada a exploração integrada de um ambiente com múltiplos veículos usando uma estratégia de exploração baseada em grade de ocupação e uma estratégia SLAM para localização dos veículos. A exploração usa conceitos de custo e utilidade de células-fronteira. Além disso, o método de SLAM utilizado é baseado no algoritmo FastSLAM com marcos extraídos do ambiente a partir de sensores visuais com a técnica SIFT (*Scale-Invariant Feature Transform*) e em um mapa de características comum aos veículos. Ambas as atividades - localização dos veículos e exploração do ambiente - são coordenadas por um agente central. Os resultados mostram que quando dois veículos comunicam-se com um agente central construindo um mapa de características dos marcos comum aos veículos, a tarefa de exploração torna-se mais eficiente do que a realizada com mapas dedicados, pois a precisão nas posição e orientação dos veículos são incrementadas.

Este aumento é devido ao uso de um mapa de características comum aos veículos utilizados.

A melhoria na precisão proporcionada pela abordagem FastSLAM com um mapa comum depende da trajetória percorrida pelo veículo. Um veículo tem que passar por uma região onde outro já tenha mapeado. Além disso, ele tem que reobservar marcos mapeados pelo outro veículo sobre um ponto de vista similar.

O aumento no custo computacional e no número duplicado de partículas pode ser compensado pela paralelização e pelo aumento do poder computacional do agente central (equivalente ao número de veículos utilizados)

## 9.1 Contribuições

As principais contribuições do presente trabalho foram (a) mostrar, através de resultados experimentais obtidos em ambientes simulados, que quando dois veículos comunicam-se com um agente central construindo um mapa de características dos marcos comum aos veículos, a tarefa de exploração torna-se mais eficiente do que a realizada com mapas dedicados, pois a precisão nas posição e orientação dos veículos são incrementadas e (b) apresentar e avaliar a implementação da abordagem em um ambiente real usando um veículo robótico autônomo e sensores (visual estereoscópico e a laser). Estas contribuições foram vislumbradas a partir do trabalho apresentado em [39] no que se refere a abordagem FastSLAM com mapa comum e a estratégia de associação de dados

Todas as outras contribuições abaixo discriminadas são secundárias e servem de suporte para as contribuições principais:

- Adaptação da técnica FastSLAM 2D para marcos 3D (FastSLAM 3D) (capítulo 4).
- Modificação da abordagem para geração da grade de ocupação com sensor laser (seção 7.1).
- Integrações de códigos:
  - Técnica FastSLAM 3D (mapa comum e mapas dedicados).
  - Extração de características SIFT.
  - Obtenção de coordenadas 3D de marcos visuais SIFT.
  - Reamostragem estratificada.
  - Estratégia de exploração coordenada (seção 7.4).
  - Planejamento da trajetória com o algoritmo *A-estrela*.
  - *Player/Stage* (simulação).
  - *Player/Pionner 3DX* (experimento real).
- Estabelecimento de uma metodologia para comparação das abordagens FastSLAM com mapa comum e mapas dedicados (Eq.8.1).

Dentre essas, destacamos a apresentada no capítulo 4, que descreve a adaptação do algoritmo FastSLAM para marcos 3D obtidos com o sensor *kinect*.

As implementações foram feitas em um compilador “gcc linux”. Algumas bibliotecas foram utilizadas: *boost/ublas*[56], *OpenKinect*[54], *A-estrela*[59], *SIFT*[55] e *kd-tree*[57], *player/stage* [58].

## 9.2 Perspectivas Futuras

Apresentamos neste capítulo sugestões de trabalhos futuros baseados em resultados do presente trabalho:

- Atualização da odometria do veículo com as estimativas obtidas pelo FastSLAM.
- Geração da grade de ocupação e do mapa ambiente (2D e 3D) com o uso do *kinect* [52] em substituição ao sensor laser LMS-200.
- Paralelização da abordagem FastSLAM com mapa comum.
- Pesquisa do uso da técnica de reconhecimento de objetos proporcionada pelo algoritmo SIFT [21] na geração de um algoritmo FastSLAM em que os marcos passam a ser objetos encontrados no ambiente [44].

## 10.1 Filtro de Kalman

O filtro de Kalman é um conjunto de equações que fornecem um eficiente meio computacional recursivo para estimar o estado de um processo que minimiza a média do erro quadrático [27]. O filtro é poderoso em muitos aspectos: ele suporta a estimação dos estados passados, presentes, e até mesmo futuros, e pode fazê-lo mesmo quando o modelo do sistema utilizado na estimação é preciso.

### 10.1.1 Filtro de Kalman Discreto

Em 1960, R.E. Kalman publicou seu famoso artigo descrevendo uma solução recursiva para o problema da filtragem linear de dados discretos. Desde então, o filtro de Kalman tem sido objeto de extensa pesquisa e aplicação, particularmente na área da navegação.

O filtro de Kalman trata do problema geral de tentar estimar o estado  $\psi \in R^n$  de um processo de tempo discreto que é governado por uma equação de diferenças linear estocástica

$$\psi_k = A\psi_{k-1} + Bu_{k-1} + w_{k-1} \quad (10.1)$$

com a medição  $z \in R^n$  que é

$$z_k = G\psi_k + v_k \quad (10.2)$$

As variáveis aleatórias  $w_k$  e  $v_k$  representam o ruído do processo e da medição, respectivamente. Assume-se que os mesmos são independentes um do outro, brancos, e com distribuição normal

$$p(w) \sim N(0, Q) \quad (10.3)$$

$$p(v) \sim N(0, R) \quad (10.4)$$

A matriz  $A$  ( $n \times n$ ) na equação de diferenças 10.1 relaciona o estado no passo anterior  $k - 1$  com o estado no passo  $k$ , na ausência ausência de uma função de condução ou de ruído no processo. A matriz  $B$  ( $n \times 1$ ) relaciona a entrada de controle opcional  $u \in R^l$  ao estado  $\psi$ . A matriz  $G$  ( $m \times n$ ) na equação de medição 10.2 relaciona o estado à medição  $z_k$ .

Iniciaremos com uma visão geral, cobrindo a operação em alto nível de um filtro de Kalman discreto para depois reduzirmos o foco para equações específicas e o uso delas nesta versão do filtro.

O filtro de Kalman estima um processo usando uma forma de controle com realimentação: o filtro estima o estado do processo em algum instante e então obtém retorno na forma de medições ruidosas. As equações do filtro de Kalman recaem em dois grupos: *equações de atualização devido ao tempo* e *equações de atualização devido a medições*. As *equações de atualização devido ao tempo* são responsáveis por projetar para frente (no tempo) o corrente estado e a estimativa da covariância do erro para se obter uma estimativa *a priori* para o próximo passo no tempo. As *equações de atualização devido a medições* são responsáveis pela realimentação - isto é, por incorporar uma nova medição na estimativa *a priori* de forma a se obter uma estimativa *a posteriori* melhorada.

As *equações de atualização devido ao tempo* podem também ser pensadas como *equações preditoras*, enquanto as *equações de atualização devido a medições* podem ser pensadas como *equações corretoras*. De fato, o algoritmo de estimação final assemelha-se ao *algoritmo preditor-corretor* para solução de problemas numérico, como mostrado na figura 10.1.

**Predição.** As equações específicas para as atualizações devido ao tempo são as abaixo discriminadas

1. Novo estado *a priori*<sup>1</sup>

$$\hat{\psi}_k^- = A\hat{\psi}_{k-1} + Bu_{k-1} \quad (10.5)$$

---

<sup>1</sup> $\hat{\psi}_k^-$  representa o estado *a posteriori* e  $\hat{\psi}_k^-$  o estado *a priori*.

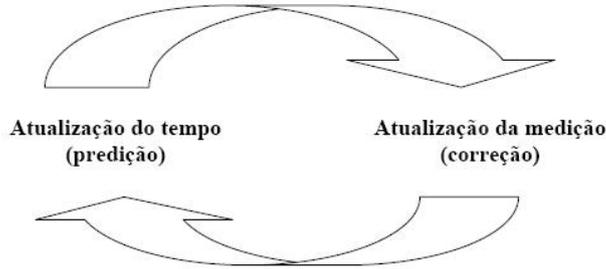


Figura 10.1: Ciclo do filtro de Kalman discreto. A *atualização devido ao tempo* projeta a estimativa do estado corrente para frente no tempo. A *atualização devido a medição* ajusta a estimativa projetada pela medição real naquele instante de tempo.

2. Nova covariância do erro *a priori*<sup>2</sup>

$$\Sigma_k^- = A\Sigma_{k-1}A^T + Q \quad (10.6)$$

Note que as equações de atualização do tempo projetam as estimativas do estado e da covariância do erro de estimação anterior para frente do passo no tempo  $k - 1$  para o passo  $k$ .

**Correção.** As equações específicas para as atualizações devido às medições são:

1. Ganho de Kalman

$$K_k = \Sigma_k^- G^T (G \Sigma_k^- G^T + R)^{-1} \quad (10.7)$$

2. Novo estado *a posteriori* devido à medição  $z_k$

$$\hat{\psi}_k = \hat{\psi}_k^- + K_k(z_k - G\hat{\psi}_k^-) \quad (10.8)$$

3. Nova covariância do erro *a posteriori*

$$\Sigma_k = (I - K_k G) \Sigma_k^- \quad (10.9)$$

O primeiro passo durante a atualização devido à medição é calcular o ganho de Kalman  $K_k$  (Eq. 10.7) e o próximo é obter  $z_k$  para então gerar uma estimativa de estado *a posteriori* ( $\hat{\psi}_k$ ) incorporando à medição esta estimativa na equação 10.8. O passo final é a obtenção da estimativa da covariância do erro *a posteriori* ( $\Sigma_k$ ) via 10.9.

Após cada par de atualização devido ao tempo e à medição, o processo é repetido com a estimativa *a posteriori* usada para projetar ou prever a nova estimativa *a priori*.

<sup>2</sup> $\Sigma_k^-$  representa a covariância do erro *a priori*.

## 10.1.2 Filtro de Kalman estendido

Como descrito na seção 10.1.1, o filtro de Kalman trata do problema geral de tentar estimar o estado  $\psi \in R^n$  de um processo controlado ou simplesmente submetido a uma perturbação externa, de tempo discreto, que é governado por uma equação de diferenças estocástica linear. Mas o que acontece se o processo a ser estimado e (ou) a relação da medição para o processo não seja linear? Algumas das mais interessantes e bem sucedidas aplicações do filtro de Kalman têm ocorrido em tais situações. O filtro de Kalman que lineariza sobre a atual média e covariância é conhecido como um filtro de Kalman estendido ou EKF.

Com algo semelhante a uma série de Taylor, podemos linearizar a estimação entorno da atual estimativa usando derivadas parciais das funções de medição e do processo para computar estimativas mesmo em face das relações não lineares. Vamos assumir que o processo tenha um vetor de estado  $\psi \in R^n$ , mas que é agora governado por uma equação de diferença estocástica não linear

$$\psi_k = h(\psi_{k-1}, u_{k-1}, w_{k-1}) \quad (10.10)$$

com a medição  $z \in R^n$  que é

$$z_k = g(\psi_k, v_k) \quad (10.11)$$

onde as variáveis aleatórias  $w_k$  e  $v_k$  representam, respectivamente, o ruído do processo e da medição. Neste caso, a função não linear  $h$  na equação 10.10 relaciona o estado em um passo prévio  $k - 1$  ao estado no passo atual  $k$ . Ele inclui como parâmetros alguma função de condução  $u_{k-1}$  e o ruído no processo de média zero  $w_k$ . A função não linear  $g$  na equação 10.11 relaciona o estado  $\psi_k$  e a medição  $z_k$ .

O conjunto completo de equações EKF são mostradas abaixo. Note que colocamos subscrito  $k$  nos Jacobianos,  $A$ ,  $W$ ,  $G$  e  $V$ , para reforçar a noção de que eles são diferentes e são computados a cada passo.

**Predição.** As equações devido à atualização do tempo do EKF são:

1. Novo estado *a priori*

$$\hat{\psi}_k^- = h(\hat{\psi}_{k-1}, u_{k-1}, 0) \quad (10.12)$$

2. Nova covariância do erro *a priori*

$$\Sigma_k^- = A_k \Sigma_{k-1} A_k^T + W_k Q_{k-1} W_k^T \quad (10.13)$$

Como com o filtro de Kalman básico, as equações de atualização do tempo projetam as estimativas de covariância do erro e o estado para frente do passo  $k - 1$  para o passo  $k$ .  $A_k$  e  $W_k$  são Jacobianos no passo  $k$ <sup>3 4</sup>, e  $Q_k$  é a covariância do ruído do processo (Eq. 10.3) no passo  $k$ .

**Correção.** As equações de atualização da medição do EKF são:

1. Ganho de Kalman

$$K_k = \Sigma_k^- G_k^T (G_k \Sigma_k^- G_k^T + V_k R_k V_k^T)^{-1} \quad (10.14)$$

2. Novo estado *a posteriori* devido à medição  $z_k$

$$\hat{\psi}_k = \hat{\psi}_k^- + K_k (z_k - g(\hat{\psi}_k^-, 0)) \quad (10.15)$$

3. Nova covariância do erro *a posteriori*

$$\Sigma_k = (I - K_k G_k) \Sigma_k^- \quad (10.16)$$

De forma similar ao filtro de Kalman discreto, as equações devido à atualização da medição corrigem as estimativas do estado e da covariância do erro com a medição  $z_k$ , onde  $G$  e  $V$  são Jacobianos das medições no passo  $k$ <sup>5 6</sup>, e  $R_k$  é a covariância do ruído de medição (Eq. 10.4) no passo  $k$ .

---

<sup>3</sup> $A$  é a matriz Jacobiana de derivadas parciais de  $h$  com relação a  $\psi$ .

<sup>4</sup> $W$  é a matriz Jacobiana de derivadas parciais de  $h$  com relação a  $w$ .

<sup>5</sup> $G$  é a matriz Jacobiana de derivadas parciais de  $g$  com relação a  $\psi$ .

<sup>6</sup> $V$  é a matriz Jacobiana de derivadas parciais de  $g$  com relação a  $v$ .

## 10.2 Tabelas

Tabela 10.1: Experimentos da tabela 8.2

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas Passos</i>
1	a-v 1	0,0489	0,0962	0,0708	218	23/25
2	a-v 1	0,0541	0,055	0,1601	201	23/25
3	a-v 1	0,0369	0,0347	0,1516	136	18/25
4	a-v 1	0,0517	0,0465	0,0747	211	22/25
5	a-v 1	0,0214	0,0417	0,1089	184	23/25
6	a-v 1	0,0275	0,0301	0,1298	164	23/25
7	a-v 1	0,0444	0,0249	0,1434	146	21/25
8	a-v 1	0,0373	0,0357	0,1434	171	21/25
9	a-v 1	0,0274	0,0675	0,239	149	17/25
10	a-v 1	0,0709	0,0186	0,1138	186	21/25
1	a-v 2	0,0434	0,0432	0,083	66	13/25
2	a-v 2	0,0271	0,0333	0,0393	86	13/25
3	a-v 2	0,0243	0,0238	0,0557	91	13/25
4	a-v 2	0,0373	0,0357	0,1333	36	11/25
5	a-v 2	0,0305	0,0413	0,0447	64	14/25
6	a-v 2	0,0328	0,0448	0,1314	44	12/25
7	a-v 2	0,0251	0,0281	0,0974	66	11/25
8	a-v 2	0,0322	0,035	0,0706	81	12/25
9	a-v 2	0,071	0,0289	0,1349	51	12/25
10	a-v 2	0,0559	0,0599	0,0899	72	14/25
1	b-v 1	0,0378	0,023	0,0686	526	23/25
2	b-v 1	0,0328	0,0513	0,1	465	23/25
3	b-v 1	0,0355	0,0525	0,0856	446	23/25
4	b-v 1	0,043	0,0397	0,0533	518	23/25
5	b-v 1	0,0295	0,0581	0,0731	476	23/25
6	b-v 1	0,0313	0,0433	0,0891	501	22/25
7	b-v 1	0,0445	0,0836	0,0728	523	23/25
8	b-v 1	0,032	0,0356	0,0448	558	23/25
9	b-v 1	0,0256	0,0368	0,0513	514	23/25
10	b-v 1	0,0211	0,0466	0,1045	410	23/25
1	b-v 2	0,0275	0,041	0,0697	563	24/25
2	b-v 2	0,0744	0,0258	0,0848	549	24/25
3	b-v 2	0,0448	0,0341	0,1067	490	23/25
4	b-v 2	0,0544	0,0216	0,0671	560	23/25
5	b-v 2	0,0222	0,0528	0,0843	451	23/25
6	b-v 2	0,0287	0,0237	0,0718	383	24/25
7	b-v 2	0,0232	0,0389	0,0706	505	24/25
8	b-v 2	0,0308	0,0506	0,0594	524	24/25
9	b-v 2	0,0297	0,1016	0,0539	434	22/25
10	b-v 2	0,0612	0,0424	0,0608	276	20/25

Tabela 10.2: Experimentos da tabela 8.3

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas Passos</i>
1	a-v 1	0,0278	0,0766	0,0378	417	23/25
2	a-v 1	0,0251	0,0659	0,0818	404	23/25
3	a-v 1	0,0313	0,0528	0,1822	301	20/25
4	a-v 1	0,0376	0,0355	0,253	251	18/25
5	a-v 1	0,03	0,0542	0,071	396	23/25
6	a-v 1	0,0421	0,0477	0,2572	247	20/25
7	a-v 1	0,0275	0,0184	0,1331	275	22/25
8	a-v 1	0,0423	0,0413	0,0497	487	23/25
9	a-v 1	0,02166	0,0277	0,2155	256	18/25
10	a-v 1	0,04023	0,0451	0,0482	478	23/25
1	a-v 2	0,0241	0,0576	0,1322	127	14/25
2	a-v 2	0,0211	0,0294	0,0589	145	14/25
3	a-v 2	0,0255	0,0347	0,0808	179	14/25
4	a-v 2	0,0455	0,0317	0,0651	140	13/25
5	a-v 2	0,0419	0,0237	0,1434	142	13/25
6	a-v 2	0,0675	0,0457	0,1041	108	13/25
7	a-v 2	0,0332	0,0203	0,1001	119	14/25
8	a-v 2	0,0523	0,0661	0,0794	127	14/25
9	a-v 2	0,0398	0,0641	0,1373	95	13/25
10	a-v 2	0,0307	0,0674	0,0737	145	14/25
1	b-v 1	0,0378	0,023	0,0686	526	23/25
2	b-v 1	0,0328	0,0513	0,1	465	23/25
3	b-v 1	0,0355	0,0525	0,0856	446	23/25
4	b-v 1	0,043	0,0397	0,0533	518	23/25
5	b-v 1	0,0295	0,0581	0,0731	476	23/25
6	b-v 1	0,0313	0,0433	0,0891	501	22/25
7	b-v 1	0,0445	0,0836	0,0728	523	23/25
8	b-v 1	0,032	0,0356	0,0448	558	23/25
9	b-v 1	0,0256	0,0368	0,0513	514	23/25
10	b-v 1	0,0211	0,0466	0,1045	410	23/25
1	b-v 2	0,0275	0,041	0,0697	563	24/25
2	b-v 2	0,0744	0,0258	0,0848	549	24/25
3	b-v 2	0,0448	0,0341	0,1067	490	23/25
4	b-v 2	0,0544	0,0216	0,0671	560	23/25
5	b-v 2	0,0222	0,0528	0,0843	451	23/25
6	b-v 2	0,0287	0,0237	0,0718	383	24/25
7	b-v 2	0,0232	0,0389	0,0706	505	24/25
8	b-v 2	0,0308	0,0506	0,0594	524	24/25
9	b-v 2	0,0297	0,1016	0,0539	434	22/25
10	b-v 2	0,0612	0,0424	0,0608	276	20/25

Tabela 10.3: Experimentos da tabela 8.4

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas Passos</i>
1	a-v 1	0,0302	0,0484	0,0838	158	18/28
2	a-v 1	0,0239	0,019	0,0577	157	17/28
3	a-v 1	0,0406	0,0361	0,1049	114	15/28
4	a-v 1	0,0342	0,0304	0,0356	185	19/28
5	a-v 1	0,0427	0,0413	0,0724	162	17/28
6	a-v 1	0,0351	0,0332	0,1234	123	15/28
7	a-v 1	0,0359	0,0346	0,0444	151	20/28
8	a-v 1	0,0357	0,0446	0,135	115	14/28
9	a-v 1	0,0493	0,0313	0,0826	145	16/28
10	a-v 1	0,0239	0,0397	0,049	191	20/28
1	a-v 2	0,0386	0,0246	0,1373	154	22/28
2	a-v 2	0,0457	0,0452	0,13	132	23/28
3	a-v 2	0,0205	0,051	0,173	93	20/28
4	a-v 2	0,0502	0,0372	0,0968	136	22/28
5	a-v 2	0,0936	0,0892	0,0804	166	22/28
6	a-v 2	0,046	0,0285	0,0942	134	23/28
7	a-v 2	0,067	0,0564	0,077	175	23/28
8	a-v 2	0,0376	0,0577	0,193	77	16/28
9	a-v 2	0,0321	0,0714	0,0733	171	24/28
10	a-v 2	0,028	0,0193	0,039	202	24/28
1	b-v 1	0,0674	0,0268	0,058	452	26/28
2	b-v 1	0,0229	0,0322	0,0555	386	21/28
3	b-v 1	0,0321	0,0436	0,0645	456	24/28
4	b-v 1	0,0288	0,0522	0,0829	405	25/28
5	b-v 1	0,0363	0,0461	0,0473	528	24/28
6	b-v 1	0,0562	0,032	0,0842	449	26/28
7	b-v 1	0,0242	0,0265	0,1038	327	22/28
8	b-v 1	0,0576	0,0251	0,0444	511	25/28
9	b-v 1	0,048	0,0295	0,159	259	20/28
10	b-v 1	0,0288	0,0333	0,0599	484	25/28
1	b-v 2	0,0183	0,0608	0,0664	502	25/28
2	b-v 2	0,053	0,0364	0,0908	415	24/28
3	b-v 2	0,0374	0,043	0,1098	523	26/28
4	b-v 2	0,0316	0,0366	0,1072	471	26/28
5	b-v 2	0,0247	0,0402	0,043	672	26/28
6	b-v 2	0,0323	0,0391	0,0683	516	26/28
7	b-v 2	0,0322	0,0585	0,1376	416	24/28
8	b-v 2	0,0398	0,0797	0,0891	469	25/28
9	b-v 2	0,0395	0,0704	0,0522	582	26/28
10	b-v 2	0,0507	0,0436	0,0854	521	26/28

Tabela 10.4: Experimentos da tabela 8.5

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas</i> <i>Passos</i>
1	a-v 1	0,0662	0,0903	0,0997	277	17/28
2	a-v 1	0,0232	0,0585	0,1775	154	12/28
3	a-v 1	0,0704	0,0298	0,0434	315	18/28
4	a-v 1	0,0535	0,0422	0,0962	304	17/28
5	a-v 1	0,035	0,0483	0,0518	276	18/28
6	a-v 1	0,0468	0,0635	0,0933	286	17/28
7	a-v 1	0,0419	0,0417	0,0732	310	20/28
8	a-v 1	0,0349	0,0433	0,0456	337	19/28
9	a-v 1	0,0381	0,0638	0,0424	353	18/28
10	a-v 1	0,0446	0,027	0,1259	241	16/28
1	a-v 2	0,0385	0,0414	0,0182	183	21/28
2	a-v 2	0,0419	0,0175	0,0372	368	25/28
3	a-v 2	0,0491	0,0422	0,0954	240	22/28
4	a-v 2	0,0503	0,0637	0,2087	196	21/28
5	a-v 2	0,0341	0,0456	0,1914	214	19/28
6	a-v 2	0,0456	0,0224	0,0946	257	22/28
7	a-v 2	0,0662	0,0565	0,0341	441	25/28
8	a-v 2	0,0774	0,07	0,023	468	25/28
9	a-v 2	0,0192	0,0427	0,1671	181	22/28
10	a-v 2	0,0233	0,0392	0,0733	390	20/28
1	b-v 1	0,0674	0,0268	0,058	452	26/28
2	b-v 1	0,0229	0,0322	0,0555	386	21/28
3	b-v 1	0,0321	0,0436	0,0645	456	24/28
4	b-v 1	0,0288	0,0522	0,0829	405	25/28
5	b-v 1	0,0363	0,0461	0,0473	528	24/28
6	b-v 1	0,0562	0,032	0,0842	449	26/28
7	b-v 1	0,0242	0,0265	0,1038	327	22/28
8	b-v 1	0,0576	0,0251	0,0444	511	25/28
9	b-v 1	0,048	0,0295	0,159	259	20/28
10	b-v 1	0,0288	0,0333	0,0599	484	25/28
1	b-v 2	0,0183	0,0608	0,0664	502	25/28
2	b-v 2	0,053	0,0364	0,0908	415	24/28
3	b-v 2	0,0374	0,043	0,1098	523	26/28
4	b-v 2	0,0316	0,0366	0,1072	471	26/28
5	b-v 2	0,0247	0,0402	0,043	672	26/28
6	b-v 2	0,0323	0,0391	0,0683	516	26/28
7	b-v 2	0,0322	0,0585	0,1376	416	24/28
8	b-v 2	0,0398	0,0797	0,0891	469	25/28
9	b-v 2	0,0395	0,0704	0,0522	582	26/28
10	b-v 2	0,0507	0,0436	0,0854	521	26/28

Tabela 10.5: Experimentos da tabela 8.6

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas</i> <i>Passos</i>
1	a-v 1	0,0258	0,0309	0,0663	139	15/27
2	a-v 1	0,0201	0,0284	0,0522	144	20/27
3	a-v 1	0,0222	0,0263	0,0775	105	18/27
4	a-v 1	0,0235	0,0483	0,0579	118	16/27
5	a-v 1	0,0313	0,0421	0,0862	125	15/27
6	a-v 1	0,045	0,021	0,1881	89	13/27
7	a-v 1	0,0225	0,0164	0,0508	122	19/27
8	a-v 1	0,0172	0,0586	0,123	108	17/27
9	a-v 1	0,0487	0,0308	0,1023	134	16/27
10	a-v 1	0,0434	0,0325	0,0729	141	17/27
1	a-v 2	0,0234	0,0222	0,0409	149	24/27
2	a-v 2	0,0435	0,0182	0,1138	135	20/27
3	a-v 2	0,03	0,0305	0,1102	115	21/27
4	a-v 2	0,0454	0,0287	0,0223	190	24/27
5	a-v 2	0,0782	0,0255	0,1109	153	23/27
6	a-v 2	0,0235	0,0379	0,1322	122	23/27
7	a-v 2	0,0322	0,0333	0,1326	115	21/27
8	a-v 2	0,0566	0,0484	0,1243	104	21/27
9	a-v 2	0,0513	0,0447	0,0657	150	24/27
10	a-v 2	0,0369	0,028	0,0425	158	24/27
1	b-v 1	0,0434	0,0343	0,058	433	25/27
2	b-v 1	0,0407	0,0383	0,0834	436	24/27
3	b-v 1	0,0544	0,0566	0,1256	308	22/27
4	b-v 1	0,0738	0,0654	0,0819	509	25/27
5	b-v 1	0,0402	0,0291	0,0407	572	25/27
6	b-v 1	0,0287	0,0613	0,082	469	24/27
7	b-v 1	0,0379	0,0363	0,0492	575	25/27
8	b-v 1	0,0304	0,0649	0,0481	467	25/27
9	b-v 1	0,0338	0,0432	0,0533	449	24/27
10	b-v 1	0,0395	0,0394	0,0535	455	25/27
1	b-v 2	0,0456	0,0579	0,0547	626	25/27
2	b-v 2	0,075	0,0788	0,0992	384	24/27
3	b-v 2	0,0485	0,0916	0,1469	336	23/27
4	b-v 2	0,0179	0,0667	0,1108	418	25/27
5	b-v 2	0,0294	0,0214	0,0514	629	25/27
6	b-v 2	0,0357	0,06117	0,0608	582	25/27
7	b-v 2	0,0293	0,0467	0,0558	601	25/27
8	b-v 2	0,0261	0,0519	0,0796	513	24/27
9	b-v 2	0,0443	0,0591	0,0873	471	24/27
10	b-v 2	0,0522	0,025	0,0597	551	25/27

Tabela 10.6: Experimentos da tabela 8.7

Nr	Caso	$x$	$y$	$\alpha$	Atualizações	<i>Estimativas Passos</i>
1	a-v 1	0,0263	0,0656	0,1061	231	17/27
2	a-v 1	0,0226	0,0369	0,0712	303	20/27
3	a-v 1	0,0547	0,052	0,1028	319	18/27
4	a-v 1	0,073	0,0304	0,0997	227	19/27
5	a-v 1	0,0421	0,0328	0,1125	290	18/27
6	a-v 1	0,0343	0,0243	0,072	292	18/27
7	a-v 1	0,0756	0,0369	0,07	225	18/27
8	a-v 1	0,079	0,0365	0,0737	271	17/27
9	a-v 1	0,0562	0,0364	0,0545	307	18/27
10	a-v 1	0,0283	0,0361	0,1195	231	17/27
1	a-v 2	0,0358	0,0582	0,1955	170	22/27
2	a-v 2	0,0532	0,0504	0,0826	310	24/27
3	a-v 2	0,097	0,0377	0,074	253	23/27
4	a-v 2	0,0195	0,0385	0,082	305	22/27
5	a-v 2	0,0376	0,0216	0,1085	193	24/27
6	a-v 2	0,0538	0,0272	0,0534	336	23/27
7	a-v 2	0,0617	0,0331	0,045	360	24/27
8	a-v 2	0,0323	0,0267	0,041	371	24/27
9	a-v 2	0,0283	0,0874	0,2864	105	16/27
10	a-v 2	0,0509	0,0289	0,0993	284	23/27
1	b-v 1	0,0434	0,0343	0,058	433	25/27
2	b-v 1	0,0407	0,0383	0,0834	436	24/27
3	b-v 1	0,0544	0,0566	0,1256	308	22/27
4	b-v 1	0,0738	0,0654	0,0819	509	25/27
5	b-v 1	0,0402	0,0291	0,0407	572	25/27
6	b-v 1	0,0287	0,0613	0,082	469	24/27
7	b-v 1	0,0379	0,0363	0,0492	575	25/27
8	b-v 1	0,0304	0,0649	0,0481	467	25/27
9	b-v 1	0,0338	0,0432	0,0533	449	24/27
10	b-v 1	0,0395	0,0394	0,0535	455	25/27
1	b-v 2	0,0456	0,0579	0,0547	626	25/27
2	b-v 2	0,075	0,0788	0,0992	384	24/27
3	b-v 2	0,0485	0,0916	0,1469	336	23/27
4	b-v 2	0,0179	0,0667	0,1108	418	25/27
5	b-v 2	0,0294	0,0214	0,0514	629	25/27
6	b-v 2	0,0357	0,06117	0,0608	582	25/27
7	b-v 2	0,0293	0,0467	0,0558	601	25/27
8	b-v 2	0,0261	0,0519	0,0796	513	24/27
9	b-v 2	0,0443	0,0591	0,0873	471	24/27
10	b-v 2	0,0522	0,025	0,0597	551	25/27

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] R. Howard (1960). Dynamic Programming and Markov Process. *MIT Press and Wiley*.
- [2] Hart, P. E., Nilsson, N. J., Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics SSC4*, Vol. 4, Issue 2, pp. 100107
- [3] Koenderick, J.J. (1984). The structure of images. *Biological Cybernetics*, 50, 363396
- [4] Koenderick, J.J. (1994). Scale-space theory:A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2), 224270
- [5] K. Konolige (1997). Small vision system: Hardware and implementation. *Proceedings of the International Symposium on Robotics Research*, pp. 111-116
- [6] Brian Yamauchi(1997). A Frontier-Based Approach for Autonomous Exploration. *Published in the Proceedings of IEEE CIRA97*
- [7] R. I. hartley (1998). Theory and practice of projective rectification. *International Journal of Computer Vision*, pp. 115-1277
- [8] Brian Yamauchi(1998). A Frontier-Based Exploration Using Multiple Robots. *Proceedings of the Second International Conference on Autonomous Agents*
- [9] Stowers, J. and Hayes, M. and Bainbridge-Smith, A. (1998). VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. *Proceedings of*

*the 1998 IEEE International Conference on Robotics & Automation.*  
pp. 1572 -1577

- [10] Jun S. Liu and Rong Chen (1998). Sequential Monte Carlo methods for dynamics systems. *J. american Statistic Assoc*, 93(443):1032-1044
- [11] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte and M. Csorba (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics And Automation*, Vol. 17, Issue 3, pp. 229-241.
- [12] Pedro Alberto Barbetta (2002) *Estatística Aplicada às Ciências Sociais* Editora UFSC, 5ª edição
- [13] I. M. Rekleits (2002). A particle filter tutorial for mobile robot localization. *TR-CIM-04-02 of Centre for Intelligent Machines, McGill University*, Montreal, Quebec, Canada.
- [14] Brown, M. and Lowe, D. G. (2002) Invariant features from interest point groups. *In British Machine Vision Conference*. pp. 656-665.
- [15] Mikolajczyk, K (2002) Detection of local features invariant to affine transformation. *Ph.D thesis, Institut National Polytechnique de Grenoble, France*.
- [16] J.W. Fenwick, P.M. Newman, J.J. Leonard (2002). Cooperative concurrent mapping and localization. *IEEE Int. Conf. on Robotics and Automation, ICRA*, pp. 1810-1817
- [17] A. A. Makarenko, S. B. Williams, F. Bourgault, H. F. Durrant-Whyte (2002). An Experiment of Integrated Exploration. *Proceedings of the 2002 IEEE-RSJ. Intl Conference on Intelligent Robots and Systems*, Vol. 1, pp. 534-539.
- [18] M. Montemerlo and S. Thrun (2003). Simultaneous localization and mapping with unknown data association. *Proceedings of the IEEE, International Conference on Robotics and Automation*, Vol. 2, pp. 1985-1991.
- [19] B. Stewart, J. Ko, D. Fox, K. Konolige (2003). A hierarchical Bayesian approach to mobile robot map structure estimation. *Proceedings of the Conference on Uncertainty in AI, UAI*.

- [20] Brian P. Gerkey, Richard T. Vaughan and Andrew Howard (2003). The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. *In Proceedings of the International Conference on Advanced Robotics (ICAR 2003)*. pp. 317-323
- [21] David Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 2, pp. 91-110
- [22] Ke, Y., and Sukthankar, R. (2004) PCA-SIFT: A More Distinctive Representation for Local Image Descriptors *Computer Vision and Pattern Recognition*
- [23] Mikolajczyk, K.; Schmid, C. (2005) A performance evaluation of local descriptors *IEEE Transactions on Pattern Analysis and Machine Intelligence*
- [24] W. Burgard, M. Moors, C. Stachniss and F. Schneider (2005). Coordinated Multi-Robot Exploration, *IEEE Transactions on Robotics*, Vol. 21, Issue 3, pp. 376-386.
- [25] S. Thrun, W. Burgard and D. Fox (2005). Probabilistic robotics. *Massachusetts Institute of Technology Press*, Cambridge, MA, USA, 2005.
- [26] Bay, H., Tuytelaars, T., Gool, L.V. (2006) SURF: Speeded Up Robust Features *Proceedings of the ninth European Conference on Computer Vision*
- [27] G. Welch and G. Bishop (2006). An Introduction to the Kalman Filters. *TR 95-041. Department of Computer Science (University of North Carolina at Chapel Hill)*
- [28] A. Howard (2006). Multi-robot simultaneous localization and mapping using particle filters. *International Journal of Robotics Research*, Vol. 25, Issue 12, pp. 1243-1256.
- [29] A. Gil, O. Reinoso, O. Martínez-Mozos, C. Stachniss, W. Burgard (2006). Improving Data Association in Vision-based SLAM. *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* pp. 2076-2081
- [30] Thomas H. Cormen, Charles E. Leiserson; Ronald L. Rivest (2006). Introduction to Algorithms. *MIT Press and McGraw-Hill. Chapter 10* Chapter 10

- [31] Kristopher R. Beevers (2007). Mapping with limited sensing. *Doctorate thesis of Rensselaer Polytechnic Institute*, TROY, NY, USA.
- [32] Pedro Pinés, Juan D. Tardós (2008). Large Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision. *IEEE Transactions on Robotics*. Vol. 24, NO. 5
- [33] Gonzalez and Woods (2008) Digital Image Processing Prentice Hall
- [34] Gary Bradski and Adrian Kaehler (2008). Published by O’Reilly Media, Inc.
- [35] Matthew Toews, William M. Wells (2009) SIFTRank: Ordinal Descriptors for Invariant Feature Correspondence IEEE International Conference on Computer Vision and Pattern Recognition
- [36] Vitor Campanholo Guizilini (2009) Localização e Mapeamento Simultâneos com Auxílio Visual OmniDirecional. *Dissertação de Mestrado da Escola Politécnica da Universidade de São Paulo*
- [37] Antônio Henrique Pinto Selvatici (2009). Construção de Mapas de Objetos para Navegação de Robôs. *Tese apresentada na Escola Politécnica da Universidade de São Paulo para obtenção do título de Doutor em Engenharia Elétrica*.
- [38] C. Fulgenzi, G. Ippoliti, S. Longhi (2009). Experimental validation of FastSLAM algorithm integrated with a linear features based map. *Elsevier: Mechatronics*, Vol. 19, Issue 5, pp. 609-616
- [39] A. Gil, O. Reinoso, M. Ballesta, M. Julia (2009). Multi-robot visual SLAM using Rao-blackwellized particle filter. *ACM: Robotics and Autonomous Systems*, Vol. 58, Issue 1, pp. 68-80.
- [40] Adão de Melo Neto, C. Pinheiro, Paulo F. F. Rosa, F. S. Vidal, R. M. Bernardo, M. M. Correa (2009). Operação Autônoma de Dirigíveis Aéreos Não Tripulados. *VIII Simpósio Brasileiro de Transporte Aéreo*, v. 8. pp. 469-483
- [41] Vilar F. da Camara Neto, Mario F. M. Campos (2009) Localização e Mapeamento Simultâneos em Robótica Cooperativa com Planejamento: uma revisão *IX Simpósio Brasileiro de Automação Inteligente (SBAI)*

- [42] Adão de Melo Neto, C. Pinheiro, Paulo F. F. Rosa, F. S. Vidal, R. M. Bernardo, M. M. Correa (2010). Autonomous Operation of a Fleet Of Unmanned Airship. *Journal of the Brazilian Air Transportation Research Society*, v. 6, pp. 39-53
- [43] Adão de Melo Neto, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2010). Mapeamento de Ambiente Interno Semi-estruturado com Múltiplos Veículos utilizando Sensor Visual Embarcado. *VI Simpósio Brasileiro de Engenharia Inercial*, v. 1. pp. 1-6
- [44] J. Okamoto Jr e V. C. Guizilini (2010) On-line SLAM Using Clustered Landmarks with Omnidirectional Vision *ABCM: Jornal da Associação Brasileira de Engenharia e Ciências Mecânicas* Vol. XXXII, pp 468-476
- [45] Adão de Melo Neto, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Exploração Integrada de um Ambiente Interno com Técnica de SLAM. *X Simpósio Brasileiro de Automação Inteligente (SBAI 2011)*, São João Del Rey - Brasil, v. X. p. 1274-1279
- [46] Adão de Melo Neto, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Integrated Exploration of an Indoor Environment with SLAM Technique. *15th Portuguese Conference on Artificial Intelligence (EPIA 2011)*. Lisboa - Portugal, 10 a 13 de Outubro de 2011, v. 1. p. 425-439
- [47] Adão de Melo Neto, Thiago Eustáquio Alves de Oliveira, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Exploration with FastSLAM Technique. *9th IEEE International Conference on Control & Automation (ICCA 11)*, Santiago - Chile, 19 a 21 de Dezembro de 2011
- [48] Adão de Melo Neto, Thiago Eustáquio Alves de Oliveira, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Environment Exploration with Multiple Vehicles and FastSLAM Technique. *37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011)*, Melbourne - Austrália, 7 a 10 de Novembro de 2011
- [49] Adão de Melo Neto, Thiago Eustáquio Alves de Oliveira, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Environment Exploration with Visual FastSLAM Technique: Experiments and Results. *3rd Int. Congress on Ultra Modern Telecommunications and Control Systems Workshops (ICUMT)*, Budapeste - Hungria v. 1. pp. 1-7

- [50] Adão de Melo Neto, Thiago Eustáquio Alves de Oliveira, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2011). Exploração de Ambiente Interno com Múltiplos Veículos e Técnica FastSLAM. *XII Defense Operational Applications Symposium (XII SIGE)*, São José dos Campos - Brasil v. 1. pp. 1-7
- [51] Adão de Melo Neto, Thiago Eustáquio Alves de Oliveira, Paulo Fernando Ferreira Rosa, Paulo César Pellanda (2012). Eficiência da Técnica FastSLAM Visual com um Mapa de Características Comum aos Veículos na Exploração Integrada de um Ambiente Interno. *Revista da Sociedade Brasileira de Automática*, Em processo de publicação (segunda versão submetida)
- [52] Kouros Khoshelham, Sander Oude Elberink (2012). Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *www.mdpi.com/journal/sensors*. sensors ISSN 1424-8220
- [53] *Pont Grey Stereo* (catálogo). <http://www.ptgrey.com>
- [54] *OpenKinect Project*. [http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)
- [55] *AI-Shack*. <http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/>
- [56] *Boost/Ublas*. <http://www.boost.org/>
- [57] *Kd-Tree*. <http://code.google.com/p/kdtree/downloads/list>
- [58] *Player/Stage*. <http://playerstage.sourceforge.net/>
- [59] *STL A star*. <http://www.heyes-jones.com/astar.html>

## ÍNDICE REMISSIVO

- Keypoint* SIFT, 57
- Matching*, 68
- Openkinect*, 93
  
- Algoritmo A estrela, 110
- Ambiguidade de medição, 45
- Ambiguidade de movimento, 46
- Amostragem por importância, 38
- Associação de dados (1), 14
- Associação de dados (2), 21
  
- Características do sensor *Kinect*, 93
- Caso (a), 115
- Caso (b), 115
  
- Descritor de um ponto-chave, 63
- Descritor SIFT, 19
  
- EKF-SLAM, 22
- Erro de atualização, 73
- Erro na associação de dados, 73
- Estrutura *K-d tree*, 74
- Exploração coordenada, 107
  
- FastSLAM, 23
- FastSLAM (2), 30
- FastSLAM com mapa comum aos veículos (1), 23
- FastSLAM com mapa comum aos veículos (2), 95
  
- Fatorização do problema do SLAM, 34
- Filtro de Bayes, 33
- Filtro de partículas, 30
  
- Grade de Ocupação, 100
  
- Invariância parcial do descritor SIFT (1), 20
- Invariância parcial do descritor SIFT (2), 67
  
- Modelo de medição, 31
- Modelo de movimento, 32
  
- Partícula no FastSLAM, 44
- Peso da partícula, 42
- Ponto-chave SIFT, 57
- Pose dos veículos, 14
  
- Reamostragem (1), 38
- Reamostragem (2), 52
- Reconhecimento de Objetos, 56
  
- Sensor *Kinect*, 91
- Sensoriamento com sensor laser, 103
- SLAM, 21
  
- Teorema de Bayes, 35