



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Localização de Robôs Móveis Autônomos Utilizando Fusão Sensorial de Odometria e Visão Monocular

Guilherme Leal Santos

Orientador: Prof. Dr. Pablo Javier Alsina

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, maio de 2010

Divisão de Serviços Técnicos

Catálogo da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Santos, Guilherme Leal.

Localização de Robôs Móveis Autônomos Utilizando Fusão Sensorial de Odometria e Visão Monocular / Guilherme Leal Santos - Natal, RN, 2010
72 p.

Orientador: Pablo Javier Alsina

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

1. Redação técnica - Tese. 2. L^AT_EX- Tese. I. Melo, Sicrano Matosinho de. II. Amaral, Beltrano Catandura do. III. Título.

RN/UF/BCZM

CDU 004.932(043.2)

Localização de Robôs Móveis Autônomos Utilizando Fusão Sensorial de Odometria e Visão Monocular

Guilherme Leal Santos

Dissertação de Mestrado aprovada em 7 de maio de 2010 pela banca examinadora composta pelos seguintes membros:

Dr. Pablo Javier Alsina (orientador) DCA/UFRN

Dr. Anfranserai Morais Dias (examinador externo) UEFS

Dr. Adelardo Adelino Dantas de Medeiros (examinador interno) . DCA/UFRN

*Ao maior presente que Deus me deu,
João Guilherme.*

Agradecimentos

Agradeço primeiramente a Deus, por ter me iluminado e dado a inteligência e força necessária para terminar este trabalho.

Aos meus pais, pelo grande incentivo, apoio e carinho. Também por sempre terem me mostrado que o caminho a ser seguido é o caminho dos estudos, e por terem me proporcionado um estudo de qualidade.

Aos meus tios Erasmo e Maria de Jesus (*in memoriam*), que deram uma grande ajuda para a realização deste trabalho.

À minha família pelo apoio durante esta jornada.

Agradeço ao meu orientador e amigo, Pablo Alsina, pelos conselhos, ensinamentos e compreensão.

Agradeço também ao professor Adelardo Medeiros e ao amigo André Macêdo, pela grande ajuda e apoio na elaboração deste trabalho. Enfim, a todo o pessoal do Laboratório de Robótica.

À CAPES, pelo apoio financeiro.

Resumo

O DESENVOLVIMENTO e aperfeiçoamento de técnicas que façam simultaneamente o mapeamento e a localização (*Simultaneous Localization and Mapping* - SLAM) de um robô móvel autônomo e a criação de mapas locais 3-D, a partir de uma sequência de imagens, é bastante estudada no meio científico.

Neste trabalho é apresentada uma técnica de SLAM visual monocular baseada no filtro de Kalman estendido, que utiliza características encontradas em uma sequência de imagens através do descritor SURF (*Speeded Up Robust Features*) e determina quais características podem ser utilizadas como marcas através de uma técnica de inicialização atrasada baseada em retas 3-D. Para isso, tem-se disponível apenas as coordenadas das características detectadas na imagem e os parâmetros intrínsecos e extrínsecos da câmera. É possível determinar a posição das marcas somente com a disponibilidade da informação de profundidade.

Os experimentos realizados mostraram que durante o percurso, o robô móvel detecta a presença de características nas imagens e, através de uma técnica proposta para inicialização atrasada de marcas, adiciona novas marcas ao vetor de estados do filtro de Kalman estendido (FKE) após estimar a profundidade das características. Com a posição estimada das marcas, foi possível estimar a posição atualizada do robô a cada passo; obtendo resultados satisfatórios que comprovam a eficiência do sistema de SLAM visual monocular proposto neste trabalho.

Palavras-chave: Robótica, Visão Computacional, Inicialização atrasada, SLAM visual.

Abstract

THE development and refinement of techniques that make simultaneous localization and mapping (SLAM) for an autonomous mobile robot and the building of local 3-D maps from a sequence of images, is widely studied in scientific circles.

This work presents a monocular visual SLAM technique based on extended Kalman filter, which uses features found in a sequence of images using the SURF descriptor (*Speeded Up Robust Features*) and determines which features can be used as marks by a technique based on delayed initialization from 3-D straight lines. For this, only the coordinates of the features found in the image and the intrinsic and extrinsic camera parameters are available. It is possible to determine the position of the marks only on the availability of information of depth.

Tests have shown that during the route, the mobile robot detects the presence of characteristics in the images and through a proposed technique for delayed initialization of marks, adds new marks to the state vector of the extended Kalman filter (EKF), after estimating the depth of features. With the estimated position of the marks, it was possible to estimate the updated position of the robot at each step, obtaining good results that demonstrate the effectiveness of monocular visual SLAM system proposed in this paper.

Keywords: Robotics, Computational Vision, Delayed Initialization, Visual SLAM.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
Lista de Algoritmos	vii
Nomenclatura	ix
1 Introdução	1
1.1 Motivação	2
1.2 Justificativa	3
1.3 Objetivos	3
1.3.1 Objetivos Específicos	4
1.4 Estrutura do Documento	4
2 Fundamentação Teórica	5
2.1 Calibração de câmera	5
2.2 Descritores de Características	7
2.2.1 <i>Scale Invariant Features</i> - SIFT	8
2.2.2 <i>Speeded Up Robust Features</i> - SURF	9
2.2.3 SIFT x SURF	13
2.3 <i>SLAM</i> - Localização e Mapeamento Simultâneos	15
2.4 Considerações	16
3 Técnicas de SLAM visual	17
3.1 Sistemas de visão	17
3.1.1 Sistemas de Visão Monocular	18
3.1.2 Sistemas de Visão Estéreo	18
3.2 Extração de pontos de interesse	18
3.3 Inicialização das marcas	19
3.3.1 Inicialização direta	19
3.3.2 Inicialização atrasada	19
3.4 Técnicas de SLAM	20
3.4.1 SLAM com Filtro de Kalman Estendido - FKE SLAM	20
3.4.2 SLAM com Filtro de Partículas - FP SLAM	20

3.4.3	Outras técnicas	21
4	Técnica Proposta	23
4.1	Caracterização do problema	25
4.2	Solução proposta	25
4.2.1	Técnica de inicialização atrasada utilizando retas 3-D	27
4.2.2	Localização do robô	29
4.3	Resultados	33
5	Conclusão e Trabalhos Futuros	43
5.1	Trabalhos Futuros	43
	Referências bibliográficas	45
A	Filtro de Kalman	49
A.1	Filtro de Kalman Discreto - FKD	50
A.2	Filtro de Kalman Estendido - FKE	51

Lista de Figuras

2.1	Sistema de coordenadas do mundo.	6
2.2	Sistema de coordenadas do robô.	6
2.3	Relação entre um ponto no mundo e um ponto na imagem: (X_i, Y_i) são as coordenadas de imagem, (X_c, Y_c, Z_c) são as coordenadas de câmera e (X_m, Y_m, Z_m) são as coordenadas de mundo.	7
2.4	Um descritor para o ponto de interesse é criado para primeiramente calcular a magnitude e orientação do gradiente, como mostrado na figura esquerda. Essas amostras são acumuladas dentro de histogramas de orientação resumindo o conteúdo em regiões 4x4, como mostrado na figura direita. Este exemplo mostra um vetor descritor 2x2 calculado a partir de um conjunto de amostras 8x8. Figura retirada de [Lowe 1999].	9
2.5	Referências detectadas pela técnica SIFT.	10
2.6	Para construir o descritor, um grid quadrático orientado com 4x4 regiões quadradas são calculadas sobre o ponto de interesse (imagem esquerda). As subdivisões 2x2 de cada quadrado corresponde ao campo atual do descritor. Estas são as somas $\sum dx$, $\sum dx $, $\sum dy$ e $\sum dy $, calculadas relativamente à orientação do grid (imagem direita). Figura retirada de [Bay et al. 2006]	12
2.7	Referências detectadas pela técnica SURF.	14
2.8	Ao invés de reduzir iterativamente o tamanho da imagem (esquerda), como no SIFT, o uso de imagens integrais pelo SURF permite o aumento de escala do filtro a um valor constante (direita) [Bay et al. 2006]	15
2.9	Modelo gráfico para os problemas de SLAM [Thrun et al. 2006].	16
4.1	Robô móvel utilizado para a experiência.	23
4.2	Problema presente durante o percurso realizado pelo robô.	24
4.3	Ambiente de teste da técnica de inicialização atrasada.	26
4.4	Probabilidade das partículas após 9 iterações, calculada utilizando 100 partículas que estimam a profundidade de uma marca em até 10 metros.	26
4.5	Movimentação do robô com a visualização de uma mesma marca e a intersecção das retas para a estimação da posição desta marca.	27
4.6	Retas traçadas para uma marca vista da Imagem 1 até a Imagem 6. O ponto verde representa $^M(X, Y, Z)$ calculado, o ponto vermelho representa $^M(X, Y, Z)$ real e os pontos em azul representam os pontos iniciais PI	29
4.7	Reflexo da luz nos objetos prejudicou o <i>matching</i> de características.	33
4.8	Trajetória realizada pelo robô móvel.	34

4.9	Zoom no final da trajetória mostrada na Figura 4.8. Comparação entre a localização por odometria e por SLAM visual. O erro entre a pose final real do robô e a pose dada pela odometria foi de $0,3m$. O erro de localização dado pelo SLAM visual foi de $0,09m$	35
4.10	Comportamento do vetor de estados e a detecção de pontos de interesse durante a movimentação do robô.	37
4.11	Comportamento da correlação de características para cada imagem.	38
4.12	Comparação entre as trajetórias calculadas por odometria e SLAM visual.	39
4.13	Zoom no final das trajetórias mostradas na Figura 4.12.	40
4.14	Comparação entre o SLAM visual com taxa de $0,05$ (em azul) e SLAM visual com taxa de $0,15$ (em vermelho).	41
A.1	O problema da filtragem	49

Lista de Tabelas

2.1	Diferenças entre SURF e SIFT	13
-----	--	----

Lista de Algoritmos

1	FILTRO DE PARTÍCULAS	21
2	FILTRO DE KALMAN	51
3	FILTRO DE KALMAN ESTENDIDO	52

Nomenclatura

A menos que referência contrária seja fornecida, as siglas abaixo possuem os seguintes significados:

FK	Filtro de Kalman.
FKE	Filtro de Kalman Estendido.
UKF	<i>Unscented Kalman Filter.</i>
FP	Filtro de Partículas.
GPS	<i>Global Positioning System.</i>
SLAM	<i>Simultaneous Localization and Mapping.</i>
SIFT	<i>Scale Invariant Features.</i>
SURF	<i>Speeded Up Robust Features.</i>
VANT	Veículo Aéreo Não-Tripulado.
AUV	<i>Autonomous Underwater Vehicle.</i>
$I(\mathbf{u}, \mathbf{v})$	Pixel na imagem.
$R(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$	Ponto nas coordenadas do robô.
$C(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$	Ponto nas coordenadas de câmera.
$M(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$	Ponto nas coordenadas do mundo.
$I\mathbf{P}_{\text{marca}}$	Marca em pixels na imagem.
$C\mathbf{P}_{\text{marca}}$	Marca nas coordenadas de câmera.
$R\mathbf{P}_{\text{marca}}$	Marca nas coordenadas do robô.
$M\mathbf{P}_{\text{marca}}$	Marca nas coordenadas de mundo.
\mathbf{P}_N	Ponto pertencente à reta N .
\mathbf{PINI}_N	Posição do centro ótico da câmera em coordenadas do mundo.
δ_N	$PFIM_N - PINI_N$.
t_N	Escalar.
$(\mathbf{CX}, \mathbf{CY})$	Centro da imagem.
\mathbf{F}	Foco da câmera.
$M\mathbf{R}_R$	Matriz de rotação entre robô e mundo.
$M\mathbf{P}_R$	Posição do robô mapeada nas coordenadas de mundo.
$C\mathbf{R}_R$	Matriz de rotação entre robô e câmera.
$C\mathbf{P}_R$	Posição do robô mapeada nas coordenadas de câmera.

Capítulo 1

Introdução

A ORIGEM da palavra *robô*, que em tcheco significa “trabalho”, é atribuído ao escritor tcheco, Karel Capeck, em sua peça de teatro de 1923, *Rossum’s Universal Robots*. O termo *Robótica* foi introduzido pela primeira vez na história de ficção científica de Isaac Asimov, “*Eu, Robô*”, em 1941. O termo surgiu como uma maneira de representar a ciência dedicada ao estudo dos robôs. Robótica é a ciência da percepção e manipulação do mundo físico através de dispositivos controlados por computador. O estudo de robótica é relacionado ao desejo de sintetizar alguns aspectos das funções humanas a partir do uso de mecanismos, sensores, atuadores e computadores. Os sistemas robóticos atuam no mundo físico, e recebem informações do ambiente através de *sensores*. Exemplos de sistemas robóticos eficientes incluem plataformas móveis para exploração planetária, braços robóticos industriais em linhas de montagem, etc.

Para realizar o desvio de obstáculos efetivamente, um robô móvel deve ser equipado com sensores que fornecem informações sobre o ambiente. Alguns desses sensores, como sensores infravermelhos, sonares e telêmetros *laser*; podem apresentar alto custo. Para contornar esse empecilho, o sistema de visão de um robô pode ser utilizado como a principal fonte de informação sensorial.

Os sensores possuem limitações que surgem de uma série de fatores. O alcance e a resolução de um sensor são sujeitos à limitações físicas. Por exemplo, câmeras não podem ver através de paredes, e a resolução espacial de uma imagem de câmera é limitada. Sensores também estão sujeitos a ruídos, que influem nos resultados das medidas e limitam a informação que pode ser extraída.

Por fornecer informações detalhadas sobre o ambiente, os sistemas de visão começaram a ser desenvolvidos por pesquisadores de todo o mundo. Um sistema de visão robusto deve ser capaz de detectar objetos e fornecer uma representação adequada do mundo para processos de nível mais alto. O sistema de visão deve ser também altamente eficiente, para responder rapidamente a uma mudança de ambiente.

A vantagem dos sistemas de visão é que, além de baratos, são passivos. Essa passividade acontece porque a atividade dos sensores não influenciam o estado do ambiente, ao contrário do que acontece com a emissão de luz infravermelha, *laser* ou pulsos de ultra-som. Outra vantagem é a quantidade de informações que uma única imagem pode fornecer, que é bem maior que as informações fornecidas pelos demais sensores.

Um sistema de visão consegue operar em praticamente todo o espectro visível de radiações eletromagnéticas e com velocidade de processamento aceitável para aplicações em tempo real. Porém, não é um sistema simples porque não pode trabalhar sob condições muito variadas de iluminação, realce, etc. Outro problema é a inflexibilidade, porque um sistema de visão só apresenta bom desempenho geralmente nas condições em que foi projetado.

O desenvolvimento e aperfeiçoamento de técnicas que façam simultaneamente o mapeamento e a localização de um robô móvel autônomo e a criação de mapas locais 3-D, a partir de uma sequência de imagens, é bastante estudada no meio científico. A principal razão é a acessibilidade, porque o uso de um sistema de visão é uma alternativa barata em relação ao uso de outros tipos de sensores, como sonar, sensor infravermelho, telêmetro laser (preços mais elevados). A captura de imagens pode ser realizada por uma simples webcam. Além do baixo custo, a quantidade de informações úteis fornecidas por uma sequência de imagens é grande.

O SLAM visual pode ser dividido em 2 tipos: o estéreo, que utiliza 2 ou mais câmeras e o monocular, que utiliza uma única câmera. O SLAM visual monocular foi apresentado por [Davison 2003], onde o sistema proposto faz uma reconstrução de ambientes internos. Em [Salvi et al. 2008] o SLAM visual foi implementado em um AUV (*Autonomous Underwater Vehicle*) equipado com um sistema de câmeras estéreo; onde o sistema proposto fornece como saída uma aquisição 3-D em larga escala do ambiente navegado. Em [Zhang et al. 2008] foi proposto um sistema de SLAM visual monocular que detecta características no ambiente utilizando um extrator SURF (*Speeded Up Robust Feature*), e utiliza um FKE (Filtro de Kalman Estendido) para calcular os estados da câmera e das marcas visuais encontradas. Em [Artieda et al. 2009] foi apresentado um sistema que utiliza a técnica de SLAM visual 3-D para VANTs (Veículos Aéreos Não-Tripulados) onde são detectadas características em um ambiente parcialmente estruturado e depois são calculadas as distâncias até o VANT.

Os trabalhos descritos no parágrafo anterior são alguns dos poucos trabalhos existentes na comunidade científica onde um sistema de SLAM visual monocular, que utiliza inicialização atrasada de marcas visuais e o algoritmo FKE SLAM (SLAM com filtro de Kalman estendido) é utilizado na navegação de um robô móvel e na geração de um mapa 3-D do ambiente.

1.1 Motivação

A localização e mapeamento simultâneo é o problema em que o robô é capaz de construir um mapa de um ambiente desconhecido e simultaneamente localizar-se no mesmo. Esta habilidade torna o robô verdadeiramente autônomo. Enquanto realiza o SLAM, o robô observa o seu ambiente de navegação e detecta a posição de alguns pontos de interesse. Alguns destes pontos de interesse são utilizados como *marcas* para o processo de SLAM.

A estimação da posição destas marcas constitui a parte de mapeamento do processo de SLAM. A medida que o robô se move, novas marcas são observadas e então é realizada uma correlação com as marcas já encontradas anteriormente e a diferença entre o valor

esperado e o valor medido é utilizada para estimar a posição do robô. Esta parte é a etapa de localização do processo de SLAM.

Na literatura, o telêmetro *laser* e o sensor sonar têm sido utilizados para a percepção do ambiente e assim realizar o SLAM. Entretanto, a situação mudou rapidamente durante a última década pois muitas pesquisas foram direcionadas para a utilização de sensores visuais para o SLAM. Estes sensores proveem uma rica quantidade de informações sobre o ambiente, possibilitando a detecção de pontos de interesse. Por isso, é muito interessante a utilização de câmeras como sensor principal de um sistema de SLAM para um robô móvel autônomo.

1.2 Justificativa

As soluções dos problemas que envolvem a localização e mapeamento simultâneos são de grande importância porque um robô só poderá ser considerado inteligente se for capaz de se localizar e orientar de forma autônoma. Isso é fundamental para a operação em ambientes extremos como o fundo do mar e a superfície de outros planetas, como Marte. Nestas condições, sistemas de posicionamento, como o GPS, são ineficazes e inaplicáveis.

Entretanto, atualmente existe um grande incentivo à utilização do SLAM para novas aplicações científicas e comerciais, como por exemplo: robôs de limpeza de baixo custo ao nível de consumidor e guias turísticos interativos que operam em museus. A localização e mapeamento simultâneos tem sido um desafio para a comunidade robótica durante várias décadas e atualmente os problemas relacionados tem despertado interesse de especialistas de outras áreas como visão computacional e inteligência artificial e cada vez passa a ser utilizada em aplicações do mundo real.

A principal contribuição desta dissertação é fornecer uma técnica de SLAM visual monocular que permita a navegação de um robô móvel autônomo em um ambiente interno, fazendo um mapeamento 3-D do local navegado. Para isso, o sistema deverá ser robusto às variações de iluminação dos ambientes de navegação do robô e deverá corrigir a posição do robô fornecida pelo sistema de odometria de rodas.

Para realizar o SLAM é necessário primeiro que características do ambiente sejam encontradas a partir de uma sequência de imagens, possibilitando uma inicialização atrasada de marcas. Para isso, foi escolhido o detector e descritor SURF por causa de sua invariância à rotação e aos efeitos de escala, e por sua rapidez na extração de características, fazendo com que o sistema de visão apresentasse resultados aceitáveis de processamento.

1.3 Objetivos

O objetivo desta dissertação é o desenvolvimento de um sistema de SLAM visual monocular que gere um mapa local 3-D da região onde ocorrerá a navegação de um robô móvel autônomo e que seja capaz de determinar a sua localização, a partir da detecção de marcas no ambiente. Será avaliado também o desempenho do sistema de visão, levando

em consideração o tempo de resposta, a robustez a ruídos, a variação de iluminação no ambiente de navegação, e a precisão dos dados fornecidos como resposta.

1.3.1 Objetivos Específicos

Considerando o que foi exposto até aqui, pode-se definir alguns objetivos específicos:

- Desenvolver um método de inicialização atrasada de marcas visuais, para que estas sejam utilizadas como entrada no vetor de estados de um filtro de Kalman estendido, a fim de determinar a posição de um robô em relação ao ambiente de navegação com maior precisão;
- Desenvolver um sistema de SLAM visual monocular, que utilize uma fusão sensorial para permitir a localização de um robô móvel autônomo em um ambiente interno desconhecido, e realizar simultaneamente um mapeamento do ambiente de navegação.

1.4 Estrutura do Documento

Este documento mostra técnicas e fundamentos teóricos que foram utilizados para a criação de sistema de SLAM visual monocular que gera um mapa local 3-D do ambiente de navegação de um robô móvel autônomo e permite a sua localização no ambiente, a partir da detecção de pontos de interesse presentes na cena. Os próximos capítulos deste trabalho estão listados a seguir:

- O capítulo 2 mostra a fundamentação teórica, onde é feito um detalhamento sobre sistemas de Visão Robótica, filtro de Kalman e localização e mapeamento simultâneo (SLAM);
- O capítulo 3 faz uma abordagem sobre as diferentes técnicas de SLAM visual existentes na literatura;
- O capítulo 4 mostra a técnica de SLAM visual proposta, fazendo uma caracterização do problema e expondo a solução proposta, juntamente com os resultados obtidos;
- O capítulo 5 aborda as conclusões e os trabalhos futuros.

Capítulo 2

Fundamentação Teórica

O SISTEMA de visão proposto neste trabalho utiliza visão monocular para a detecção e rastreamento de pontos de interesse no ambiente de navegação do robô móvel autônomo. Nas seções a seguir, serão expostas as técnicas utilizadas no processamento das imagens (calibração de câmera, filtragem, detecção e descrição de características) e as técnicas utilizadas para a localização e mapeamento simultâneos do robô, descrevendo as principais características e objetivos de SLAM.

2.1 Calibração de câmera

Para a reconstrução 3-D ou cálculo da posição de objetos no espaço, é necessário definir relações entre coordenadas de pontos 3-D com as coordenadas 2-D de imagens dos mesmos. O referencial da câmera pode ser localizado em relação a algum outro referencial bem conhecido (referencial de mundo), considerando que foi assumido um referencial. As coordenadas das imagens de pontos no quadro de câmera podem ser obtidas a partir das coordenadas de *pixels* (únicas coordenadas disponíveis a partir da imagem), pelo menos x e y .

No padrão adotado tem-se que $(^M X, ^M Y)$ é um sistema de coordenadas do mundo; $(^R X, ^R Y)$ é um sistema de coordenadas do robô; $(^C X, ^C Y, ^C Z)$ é o sistema de coordenadas da câmera; $(^M X_R, ^M Y_R)$ é a localização do robô mapeada em coordenadas do mundo e a variável $^M \theta_R$ representa o ângulo de rotação do sistema de coordenadas do robô.

A principal idéia por trás da *calibração* é determinar as equações de projeção, relacionando as coordenadas conhecidas de um conjunto de pontos 3-D e suas projeções, e resolver para os parâmetros da câmera. Esses parâmetros podem ser subdivididos em parâmetros *intrínsecos* e *extrínsecos* [Trucco & Verri 1998].

Os parâmetros intrínsecos são os parâmetros necessários para associar as coordenadas de *pixel* de um ponto na imagem com as respectivas coordenadas no referencial de câmera. São constituídos por:

- f : distância focal;
- (O_x, O_y) : localização do centro da imagem, coordenadas de *pixel*;
- (S_x, S_y) : tamanho efetivo horizontal e vertical do *pixel*;

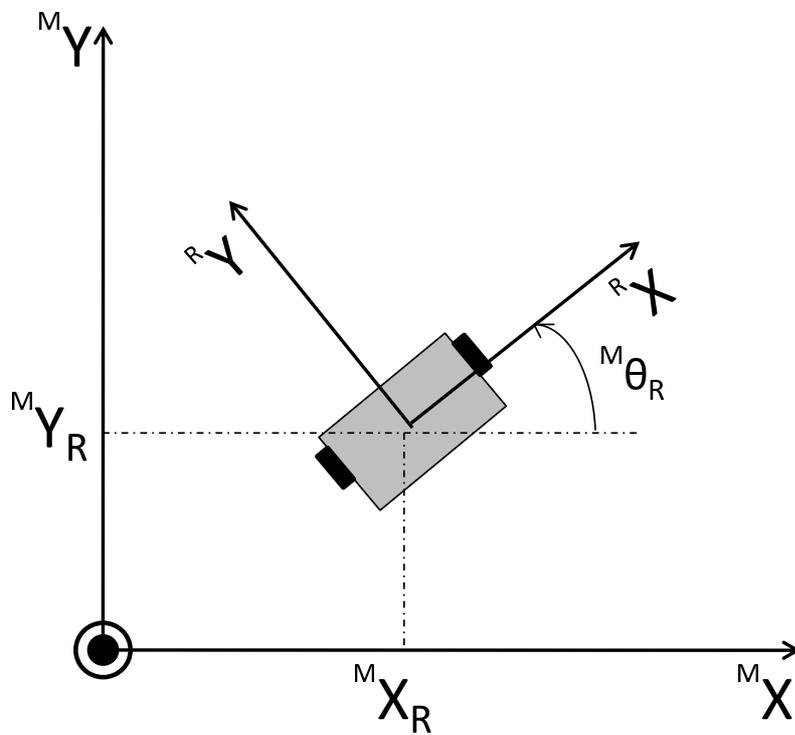
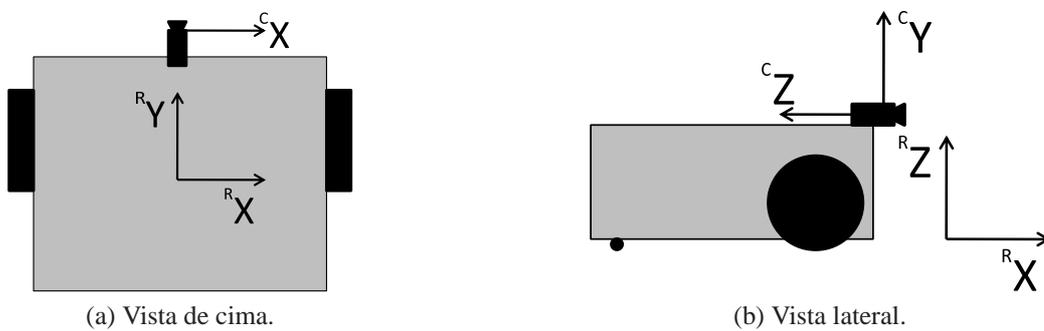


Figura 2.1: Sistema de coordenadas do mundo.



(a) Vista de cima.

(b) Vista lateral.

Figura 2.2: Sistema de coordenadas do robô.

- (K_1, K_2) : coeficientes de distorção, se forem requeridos;

Os parâmetros extrínsecos são os parâmetros que definem a localização e orientação do quadro de câmera com relação a um referencial de mundo conhecido. São constituídos por:

- T : vetor de translação;
- R : matriz de rotação;

A Figura 2.3 mostra como ocorre a projeção de um ponto no mundo (X_m, Y_m, Z_m) em uma imagem, obtendo um ponto (X_i, Y_i) .



Figura 2.3: Relação entre um ponto no mundo e um ponto na imagem: (X_i, Y_i) são as coordenadas de imagem, (X_c, Y_c, Z_c) são as coordenadas de câmera e (X_m, Y_m, Z_m) são as coordenadas de mundo.

Neste trabalho, a técnica utilizada para a calibração foi implementada com base na biblioteca de visão computacional *OpenCV*, em linguagem C++. Tal técnica foi adotada porque apresenta resultados satisfatórios e é muito utilizada na literatura.

2.2 Descritores de Características

As características de uma imagem são partes locais detectáveis, que possuem algum significado. Devem ser detectáveis, pois deve haver um algoritmo de detecção, caso contrário a característica não irá servir e devem possuir significado porque devem ser associadas a elementos da cena que sejam de interesse, através do processo de formação da imagem. Por exemplo: contornos (apresentam alta variação de intensidade) e regiões com nível de intensidade uniforme.

Durante o processo de aquisição de imagem, qualquer elemento que não interessa aos propósitos do processamento principal é considerado como um *ruído*. Para algoritmos de extração de características (como bordas ou detecção de linhas), o ruído pode ser uma variação dos valores dos *pixels* introduzidos pelo sistema de aquisição de imagem.

Existem vários problemas em visão computacional onde a correspondência de imagens é fundamental, como o reconhecimento de objetos, construção de mapas 3-D de um ambiente, localização de robôs através de um SLAM visual. Os descritores locais podem ser utilizados para solucionar o problema de correspondência de imagens, pois são vetores de características de uma imagem ou de uma determinada região da imagem. Várias técnicas podem ser utilizadas para se descrever regiões locais em uma imagem.

A procura por correspondências de pontos em imagens discretas pode ser dividida em três etapas principais. Primeiro, os pontos de interesse (características) são selecionados em pontos distintos da imagem, como cantos e bordas. A propriedade mais valiosa de um

detector de característica é sua *repetibilidade* [Bay et al. 2006]. A repetibilidade expressa a confiança de um detector para encontrar o mesmo ponto de interesse físico dentro de diferentes condições de visão.

Em seguida, a vizinhança de cada característica encontrada é representada por um vetor. Este vetor de descrição tem que ser distintivo e ao mesmo tempo robusto a ruídos, e a variações na detecção da imagem e deformações geométricas e fotométricas. Finalmente, o vetor de descrição é correlacionado entre diferentes imagens. Essa correlação é baseada na distância entre os vetores, podendo ser utilizada a distância Euclidiana ou a distância de Mahalanobis. A dimensão do descritor tem um impacto direto no tempo computacional, e descritores com dimensões menores são mais adequados para uma rápida correlação entre pontos de interesse. Entretanto, um vetor de descrição de menor dimensão é menos distintivo que um vetor de maior dimensão.

O descritor mais simples é um vetor com as intensidades dos *pixels* da imagem. A medida de correlação cruzada pode ser utilizada para calcular a similaridade entre duas regiões [Trucco & Verri 1998]. Porém, a complexidade computacional desta comparação é alta devido ao vetor ser de grande dimensão. Existem vários métodos na literatura para se identificar e rastrear características detectadas em um ambiente. Os métodos mais citados na literatura são: o método SIFT (*Scale Invariant Features*) e o SURF (*Speeded Up Robust Features*), que serão detalhados a seguir.

2.2.1 *Scale Invariant Features* - SIFT

Desenvolvido por David G. Lowe (1999), SIFT é uma técnica de processamento de imagens que permite a detecção e extração de descritores locais, invariáveis a mudanças de iluminação, ruído de imagem, rotação, escala e pequenas mudanças de perspectiva. Estes descritores podem ser utilizados para fazer a correspondência de diferentes visões de um objeto ou cena. O algoritmo SIFT foi criado inicialmente para o reconhecimento de objetos [Lowe 1999].

Um ponto calculado com a técnica SIFT pode ser corretamente encontrado com grande probabilidade dentre um grande número de pontos de diversas imagens, porque os descritores obtidos possuem grande distinção. As características SIFT são localizadas no máximo e mínimo de funções diferenças de gaussianas (DoG) aplicadas em um espaço escalar. Em seguida, os descritores são calculados baseados em histogramas de orientação em regiões 4x4 ao redor do ponto de interesse, resultando em um vetor com 64 elementos. A Figura 2.4 mostra como ocorre o processo de criação do vetor descritor.

Os descritores SIFT podem ser obtidos através das seguintes etapas [Belo 2006]:

1. **Detecção de extremos:** é utilizada a diferença de filtros gaussianos de modo a identificar pontos de interesse invariáveis à escala e rotação;
2. **Localização de características:** para cada detecção de extremo, um modelo detalhado é ajustado para determinar-se a localização e escala. Os pontos de interesse são então selecionados com base em medidas de estabilidade;
3. **Definição de orientação:** é definida a orientação de cada característica através dos gradientes locais da imagem. Toda operação a partir de então será feita com relação

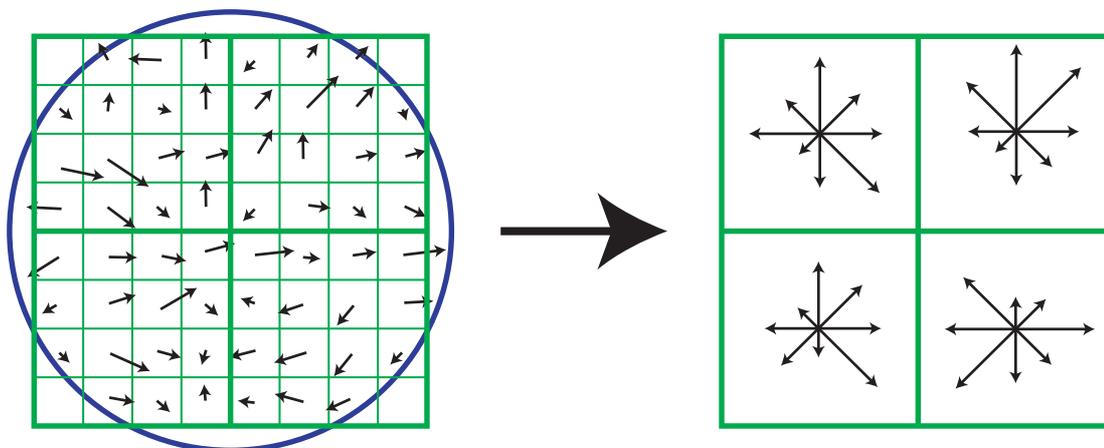


Figura 2.4: Um descritor para o ponto de interesse é criado para primeiramente calcular a magnitude e orientação do gradiente, como mostrado na figura esquerda. Essas amostras são acumuladas dentro de histogramas de orientação resumindo o conteúdo em regiões 4x4, como mostrado na figura direita. Este exemplo mostra um vetor descritor 2x2 calculado a partir de um conjunto de amostras 8x8. Figura retirada de [Lowe 1999].

a dados da imagem transformados em relação à orientação, escala e localização de cada característica. Assim, obtém-se invariância às transformações;

4. **Descritores das características:** os descritores são gerados ao medir-se gradientes locais em uma região vizinha a cada característica. Estas medidas são transformadas para uma representação que permite níveis significativos de distorção e mudança na iluminação.

A Figura 2.5 mostra 2 imagens consecutivas no ambiente de navegação do robô móvel, sendo que os pontos azuis representam as características detectadas e as retas interligam as características correlacionadas nas 2 imagens consecutivas. Para maior detalhamento matemático sobre a técnica SIFT, consultar [Lowe 1999].

2.2.2 *Speeded Up Robust Features - SURF*

O SURF é um descritor invariante à rotação e aos efeitos de escala apresentado por [Bay et al. 2006]. Nesta técnica, o processo de detecção de referências baseia-se na matriz Hessiana. A técnica SURF surgiu como um desafio de criar um detector e descritor de características que, em comparação ao estado da arte, fosse rápido de calcular sem sacrificar o desempenho. Para isso, foi simplificado o processo de detecção mantendo a precisão, e reduzido o tamanho do vetor de descrição de forma a mantê-lo suficientemente distinto.

O descritor SURF padrão tem uma dimensão de 64 e a versão estendida (e-SURF) tem a dimensão de 128 [Ballesta et al. 2007]. O maior avanço deste método em relação aos anteriores é a velocidade de detecção, permitindo a sua utilização em diversas aplicações em tempo real.

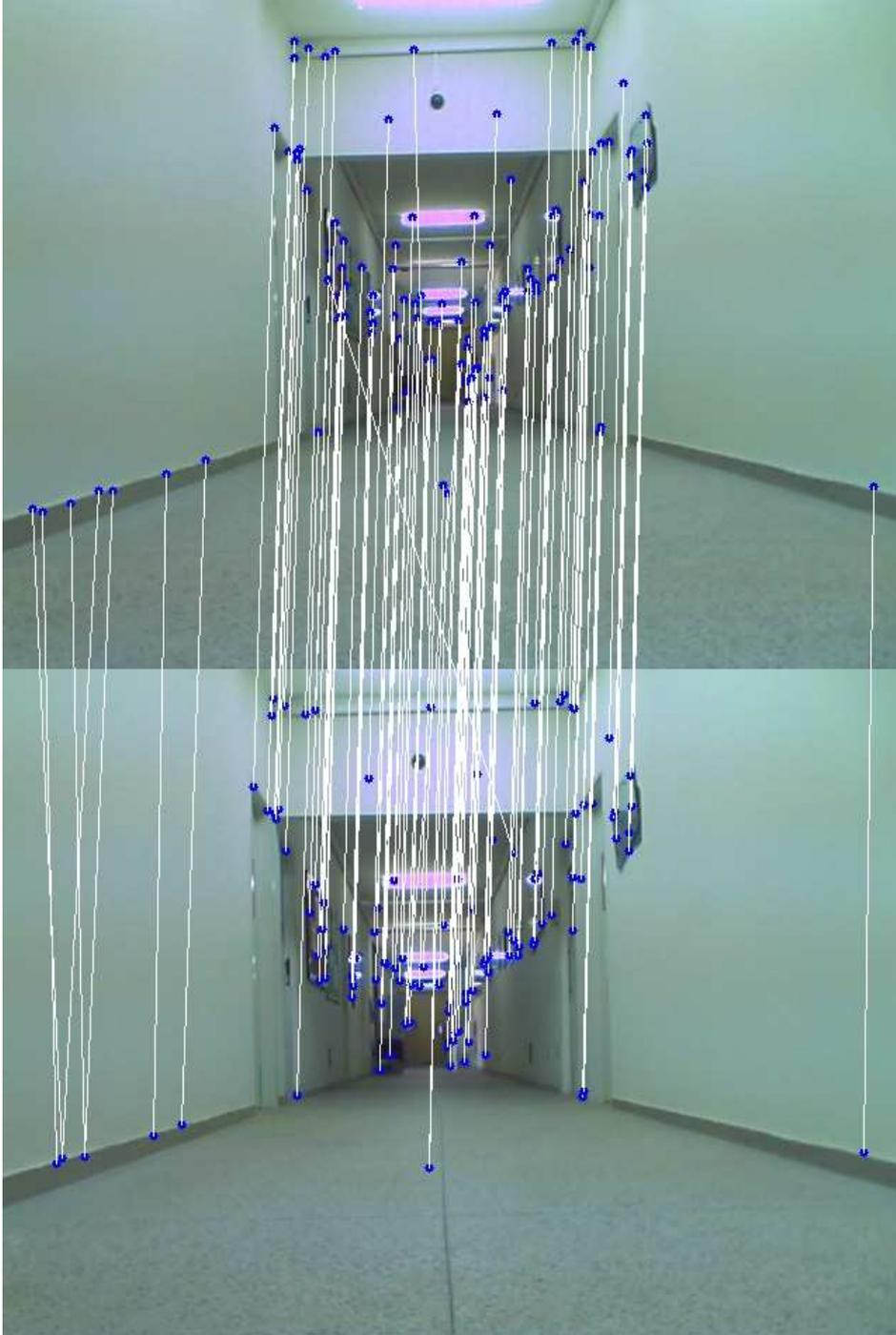


Figura 2.5: Referências detectadas pela técnica SIFT.

A técnica SURF pode ser resumida em quatro etapas:

1. **Detecção dos pontos de interesse:** é utilizada uma matriz Hessiana básica de aproximação, porque esta possui um bom desempenho quanto à precisão. São detectadas regiões com a aparência de borões onde o determinante é máximo. Dado um ponto $p = (x, y)$ em uma imagem I , a matriz Hessiana $H(p, \sigma)$ em p na escala σ é definida como

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix},$$

onde $L_{xx}(p, \sigma)$ é a convolução da derivada de segunda ordem da Gaussiana $\frac{d^2}{dx^2}g(\sigma)$ com a imagem I no ponto p , e similarmente para $L_{xy}(p, \sigma)$ e $L_{yy}(p, \sigma)$.

2. **Representação dos pontos de interesse no espaço escalar:** o espaço escalar é dividido em oitavas, onde cada uma representa uma série de mapas de resposta filtrados obtidos através da convolução da imagem inicial com um filtro com crescimento de tamanho.
3. **Localização dos pontos de interesse:** para a localização na imagem e através de escalas, o máximo do determinante da matriz Hessiana é interpolado nos espaços da imagem e da escala. Isto é importante porque a diferença em escala entre as primeiras camadas de cada oitava é relativamente grande.
4. **Descrição e correlação de pontos de interesse:** o vetor de descrição representa a distribuição de intensidade dentro da vizinhança dos pontos de interesse, similarmente à informação de gradiente extraído pelo SIFT [Lowe 1999]. Primeiramente, é identificada uma orientação reproduzida baseada em informações de uma região circular, ao redor do ponto de interesse. Em seguida, é construída uma região quadrada alinhada à orientação selecionada e é extraído o descritor SURF desta. A Figura 2.6 mostra o processo de criação do vetor de descrição.

Localização dos pontos de interesse

Os pontos de interesse precisam ser encontrados em diferentes escalas, porque as buscas de correspondências muitas vezes requerem a sua comparação em imagens onde os pontos de interesse são vistos em diferentes escalas. Os espaços de escala são geralmente implementados como uma imagem em pirâmide. As imagens são repetidamente suavizadas com uma Gaussiana e sub-amostradas, a fim de alcançar um nível mais alto da pirâmide.

Devido ao uso de filtros de bloco 9x9 e imagens integrais, não há a necessidade de aplicar iterativamente o mesmo filtro em uma saída de uma camada já filtrada, mas podem-se aplicar filtros de bloco de qualquer tamanho, exatamente na mesma velocidade, diretamente na imagem original. Portanto, o espaço de escala é analisado aumentando-se a escala do tamanho do filtro ao invés de reduzir iterativamente o tamanho da imagem (Ver Figura 2.8 na Seção 2.2.3). A saída do filtro 9x9 é considerada como a escala da camada inicial, para a qual é referida como $s =$ escala 1:2. As seguintes camadas são obtidas filtrando a imagem com máscaras de tamanho maior.

O espaço de escala é dividido em oitavas. Uma oitava representa uma série de mapas de resposta do filtro obtida por convolução com a imagem de entrada com um filtro de tamanho crescente. Portanto, uma oitava engloba um fator de escala de 2 e cada uma é dividida em um número constante de níveis de escala.

Descrição dos pontos de interesse

O descritor mostra a distribuição da intensidade dentro da vizinhança do ponto de interesse. Para isso, utiliza-se as respostas da wavelet de Haar na direção X e Y ao invés do gradiente, são exploradas em imagens integrais e utilizadas apenas 64 dimensões. Isso reduz o tempo de computação e correspondência, e provou, simultaneamente, aumentar a robustez.

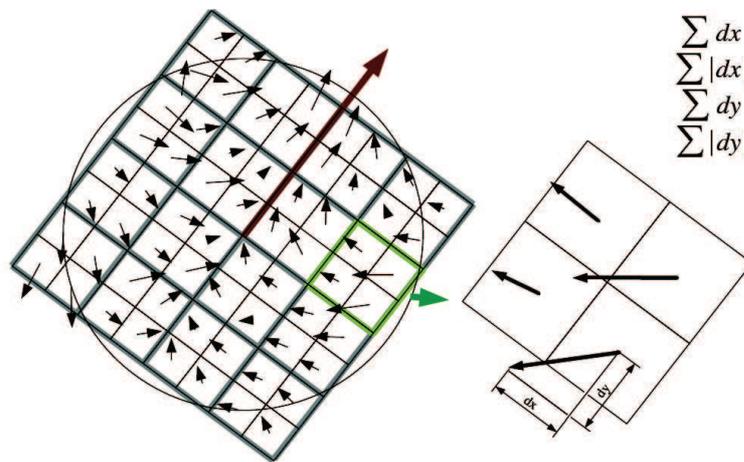


Figura 2.6: Para construir o descritor, um grid quadrático orientado com 4x4 regiões quadradas são calculadas sobre o ponto de interesse (imagem esquerda). As subdivisões 2x2 de cada quadrado corresponde ao campo atual do descritor. Estas são as somas $\sum dx$, $\sum |dx|$, $\sum dy$ e $\sum |dy|$, calculadas relativamente à orientação do grid (imagem direita). Figura retirada de [Bay et al. 2006]

A fim de ser invariante à rotação de imagem, é possível obter-se uma orientação para os pontos de interesse. Para isso, calcula-se as respostas da wavelet de Haar nas direções X e Y dentro de uma região circular em torno do ponto de interesse. Uma vez que as respostas wavelet são calculadas e ponderadas com uma gaussiana centrada no ponto de interesse, as respostas são representadas como pontos em um espaço com uma resposta ao longo da abscissa x e da ordenada y . A posição dominante da orientação é estimada pelo cálculo da soma de todas as respostas dentro de uma janela deslizante de orientação.

Para a extração do descritor, o primeiro passo consiste na construção de uma região em torno da região do ponto de interesse. A região é dividida regularmente em pequenos quadrados de 4x4, onde cada quadrado se subdivide em regiões de 2x2 pixels, preservando assim a informação espacial. Por razões de simplicidade, a resposta wavelet na abscissa

foi chamada de dx e na ordenada foi chamada de dy (ver Figura 2.6). Para aumentar a robustez para deformações geométricas e erros de localização, as respostas dx e dy são ponderadas com uma Gaussiana centrada no ponto de interesse. Em seguida, os pulsos dx e dy são somados ao longo de cada sub-região e formam um primeiro conjunto de entradas.

A Figura 2.7 mostra a técnica sendo utilizada no ambiente de navegação do robô móvel, sendo que os pontos de diferentes cores representam as características detectadas e as retas interligam as características correlacionadas nas 2 imagens consecutivas. Para maior detalhamento matemático sobre a técnica SURF, consultar [Bay et al. 2006]. A biblioteca *OpenSURF* utilizada neste trabalho está detalhada em [Evans 2009].

2.2.3 SIFT x SURF

Existe uma dificuldade em comparar os diferentes tipos de extração de referências, levando em consideração as diferentes abordagens e as condições distintas de teste. [Mozos et al. 2007] analisaram recentemente a qualidade de vários detectores de pontos de interesse utilizados no SLAM visual.

Os detectores foram avaliados de acordo com a sua repetibilidade e robustez sob variações no ponto de observação e de escala, tanto para objetos planos como para cenários 3-D usualmente utilizados no SLAM visual.

A medição foi feita baseando-se no “coeficiente de sobrevivência”, S_i , que indica a percentagem de referências detectadas na primeira imagem e que se mantêm localizadas até a i -ésima imagem [Mozos et al. 2007]. Esse coeficiente é muito importante para as aplicações de visão que utilizam características, porque tal percentagem não pode ter um valor muito baixo (ou seja, baixo número de correspondências) e nem muito alto. Portanto, tem influência direta no desempenho do sistema de visão. A Figura 2.8 a seguir mostra a diferença da representação do espaço escalar no SIFT e no SURF:

Um resumo destes resultados pode ser consultado na tabela a seguir:

Técnicas	Vantagens	Desvantagens
SIFT	Invariante à translação, rotação e efeitos de escala; Pouco sensível a variações na luminosidade;	Elevada complexidade computacional na extração de características
SURF	Mais robusto que o SIFT a variações de escala; Extração de características mais rápida que no SIFT	Normalmente seleciona poucos pontos de interesse

Tabela 2.1: Diferenças entre SURF e SIFT

Em um estudo recente, [Gil et al. 2009] fez um estudo comparativo do cálculo de descritores de pontos de interesse para SLAM visual. Os resultados obtidos com critérios diferentes mostraram que SURF é o melhor descritor. Portanto, o descritor SURF seria o mais adequado em situações em que a rotação da câmera não é restringida.

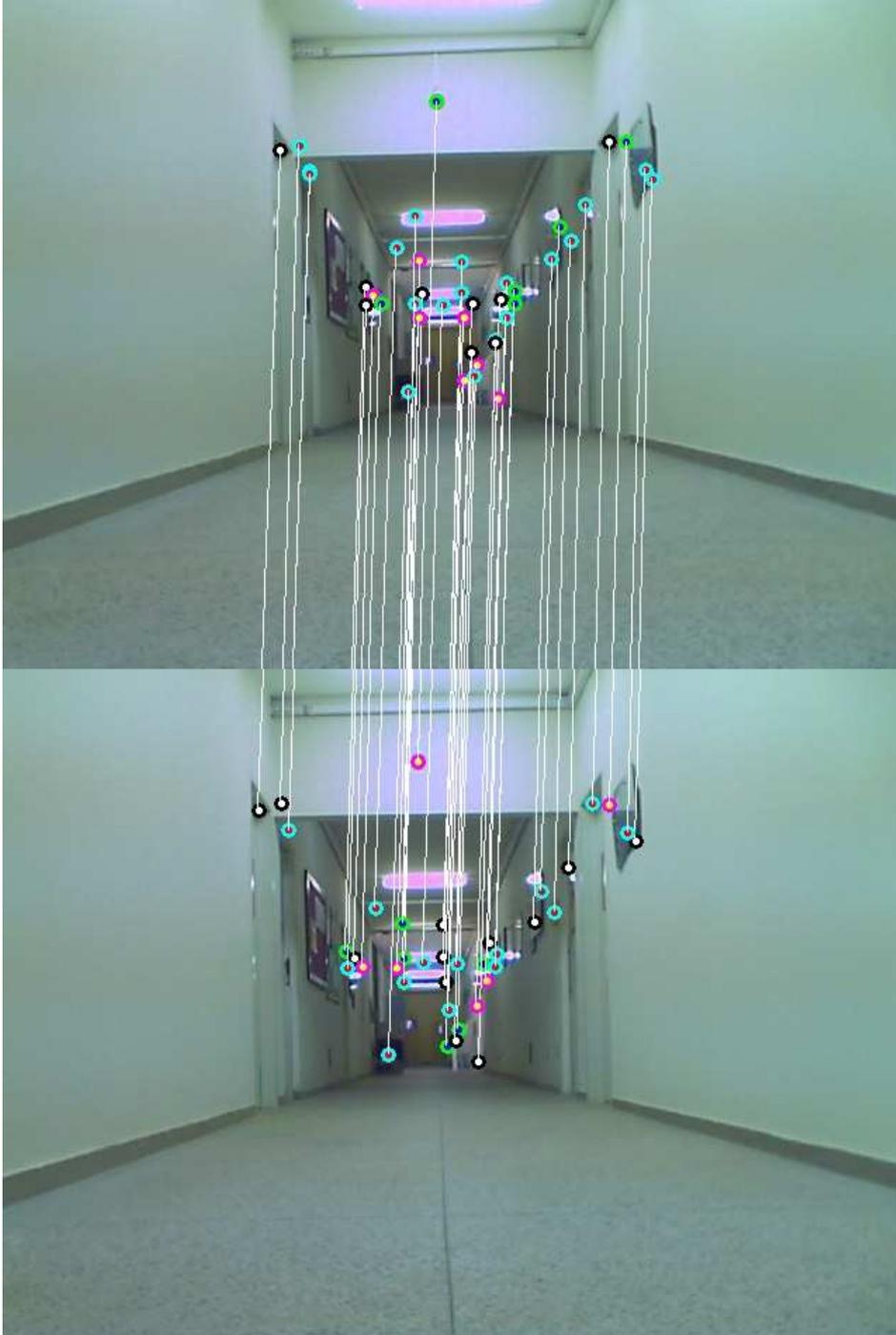


Figura 2.7: Referências detectadas pela técnica SURF.

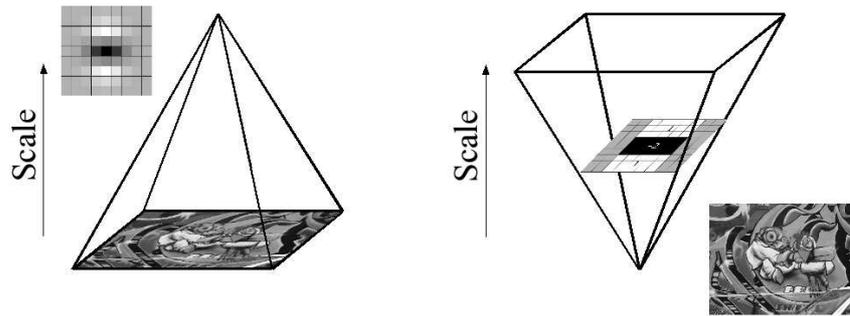


Figura 2.8: Ao invés de reduzir iterativamente o tamanho da imagem (esquerda), como no SIFT, o uso de imagens integrais pelo SURF permite o aumento de escala do filtro a um valor constante (direita) [Bay et al. 2006]

2.3 SLAM - Localização e Mapeamento Simultâneos

O problema de localização e mapeamento simultâneos (SLAM - *Simultaneous Localization and Mapping*) surge quando um robô móvel não tem acesso a um mapa do ambiente de navegação, nem conhece a sua própria localização. As primeiras implementações de SLAM foram feitas por Moutarlier e Chatila (1989). O termo SLAM foi originalmente desenvolvido por Hugh Durrant-Whyte e John J. Leonard (1991) baseado no trabalho anterior de Smith, Self e Cheeseman (1990). Durrant-Whyte originalmente designou-o por SMAL, mas foi mais tarde alterado para criar maior impacto.

O SLAM consiste em múltiplos processos: extração de referências espaciais, associação dos dados, estimação do estado (posição) e atualização do estado e das referências. O objetivo é utilizar o ambiente de navegação para determinar a posição do robô móvel. O método a ser utilizado deve extrair referências espaciais do ambiente (marcas ou *landmarks*) e reobservá-las à medida que o robô se movimenta. Estas referências são características que são facilmente reobserváveis e distinguíveis perante as restantes [Lucas 2009].

Um *estado* pode ser definido como um conjunto de todos os aspectos do robô (como pose, velocidade, configuração dos atuadores robóticos) e também aspectos de localização de marcas presentes no ambiente de navegação. De uma perspectiva probabilística, existem duas formas principais do problema de SLAM [Thrun et al. 2006]:

- **SLAM em tempo real** (*online SLAM*): é a estimação da posição posterior, tendo disponível somente a posição corrente x_t do robô no mapa m , com as medições $z_{1:t}$ e ações de controle $u_{1:t}$ (Figura 2.9a).

$$p(x_t, m | z_{1:t}, u_{1:t})$$

- **SLAM completo** (*full SLAM*): é a estimação da posição posterior, tendo disponível todas as posições anteriores do robô (Figura 2.9b).

$$p(x_{1:t}, m | z_{1:t}, u_{1:t})$$

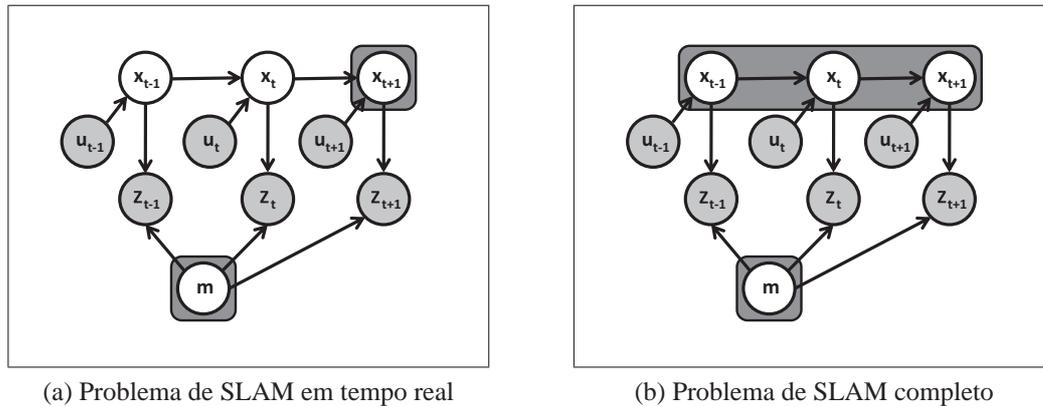


Figura 2.9: Modelo gráfico para os problemas de SLAM [Thrun et al. 2006].

Existem dois problemas principais que podem comprometer a eficácia de um algoritmo de SLAM:

- **Instabilidade numérica:** a complexidade computacional do algoritmo aumenta a medida que o número de marcas adicionadas ao vetor de estados cresce. Para resolver esse problema, são utilizadas técnicas que restringem a quantidade (e a qualidade, no caso de marcas visuais) de marcas a serem adicionadas ao vetor de estados.
- **Problema de *closed-loop*:** dificuldade em correlacionar marcas vistas anteriormente quando o robô passa por um local já navegado anteriormente. Este problema é tratado com a utilização de técnicas de sub-mapeamento, reduzindo a quantidade de marcas a serem correlacionadas.

2.4 Considerações

Para problemas de navegação de robôs móveis autônomos, onde a visão é utilizada como a principal fonte sensorial e pontos de interesse na imagem são utilizados para realizar a localização do robô no ambiente, é importante que o sistema de visão tenha uma capacidade de detecção e descrição de características em tempo real. Assim, neste trabalho, optou-se pela utilização da técnica SURF, devido ao que foi apresentado na Seção 2.2.3.

A utilização do algoritmo de filtro de Kalman estendido para solucionar o problema de SLAM foi a alternativa adotada, porque as técnicas de localização absolutas, como a utilização de marcas visuais, são imprecisas devido às medidas ruidosas dos sensores utilizados. Portanto, é interessante a utilização de um filtro para obter as informações sensoriais com a maior precisão possível [Santana 2007]. No Capítulo 3 serão expostas as principais características e funcionalidades de SLAM visual, mostrando as principais técnicas utilizadas e qual a técnica proposta neste trabalho. No Capítulo 4 serão mostrados os resultados experimentais obtidos.

Capítulo 3

Técnicas de SLAM visual

A VISÃO é uma rica fonte de informação de nosso ambiente. Por isso que os algoritmos de SLAM começaram a ser usados com informações visuais [Artieda et al. 2009]. A informação fornecida por sistemas de visão consiste em uma vasta quantidade de dados a serem processados para que os algoritmos de SLAM sejam providos de informações úteis. Portanto, os algoritmos de processamento de imagens tem grande importância nos resultados e no sucesso da resolução do problema de SLAM visual.

Os telêmetros *lasers* são sensores ativos de grande precisão, mas a sua característica de medição ponto-a-ponto limita a sua capacidade de reconhecer e seguir objetos. Os sistemas de localização baseados em sonar são rápidos, baratos e tem uma capacidade de reconhecer e seguir objetos semelhantes à dos sistemas que se baseiam na visão, mas tendem a produzir medições menos precisas e mais suscetíveis ao ruído. Os sensores inerciais, assim como os odômetros, são úteis porém o acúmulo de erros de medição geram grande imprecisão nas estimativas de posição posteriores.

Entretanto, os sistemas de localização baseados em visão são baratos, pesam pouco e permitem medições de grande alcance e com grande resolução [Chen et al. 2007]. A visão é muito utilizada nas técnicas de percepção acopladas ao SLAM porque existem técnicas eficazes de visão computacional que consistem em extrair e comparar características visuais. Uma das limitações para o uso de um sistema de visão é o elevado esforço computacional associado, tornando difícil a criação de sistemas para funcionamento em tempo real.

A maioria das técnicas de SLAM visual existentes na literatura podem ser divididas em quatro etapas: percepção do ambiente utilizando um sensor visual, extração de pontos de interesse, seleção de pontos de interesse e a inicialização como marcas a serem utilizadas pelo SLAM, implementação de um algoritmo de estimação. Nas seções a seguir, serão descritas as principais características destas partes.

3.1 Sistemas de visão

Quanto à etapa de percepção do ambiente, o SLAM visual pode ser dividido em sistemas que utilizam somente uma câmera (monocular) ou com duas ou mais câmeras (estéreo).

3.1.1 Sistemas de Visão Monocular

Os sistemas de visão monocular utilizam somente uma câmera para capturar as imagens do ambiente, não fornecendo diretamente a informação de profundidade. O SLAM visual monocular foi apresentado por [Davison 2003], onde o sistema proposto faz uma reconstrução de ambientes internos através do mapeamento de um conjunto de características utilizando modelagem de movimento.

Em [Artieda et al. 2009] foi apresentado um sistema que utiliza a técnica de SLAM visual 3-D para VANTs (Veículos Aéreos Não-Tripulados) onde são detectadas características em um ambiente parcialmente estruturado e depois são calculadas as distâncias até o VANT. O SLAM visual monocular também obteve bons resultados nos trabalhos de [Sim et al. 2005, Mouragnon et al. 2006, Lemaire & Lacroix 2007a, Zhang et al. 2008].

3.1.2 Sistemas de Visão Estéreo

O SLAM visual estereo utiliza duas ou mais câmeras para capturar as imagens do ambiente, e através da técnica de triangulação de pontos é possível obter a informação de profundidade. Entretanto, o problema da correlação de características é mais complicado que no caso do SLAM visual monocular, porque há um aumento na complexidade computacional, dado que as características tem que ser correlacionadas entre as duas imagens do par estereo e entre as aquisições consecutivas no tempo [Lemaire et al. 2007b].

[Davison & Murray 2002] apresentou o primeiro exemplo de um sistema geral para localização autônoma utilizando visão ativa. Tal sistema foi possível devido à utilização de visão estereo de alto desempenho, abordando questões como a seleção de medidas baseadas em incertezas, mapa automático de manutenção e direção à objetivo. Foram apresentados vários experimentos em tempo real em um ambiente complexo.

Em [Salvi et al. 2008] o SLAM visual foi implementado em um AUV (*Autonomous Underwater Vehicle*) equipado com um sistema de câmeras estereo; onde o sistema proposto fornece como saída uma aquisição 3-D em larga escala do ambiente navegado. [Lemaire et al. 2007b] implementou um sistema de SLAM visual estereo para robôs terrestres e aéreos.

3.2 Extração de pontos de interesse

Em um sistema de SLAM visual, as características tem que ser extraídas do ambiente para serem utilizadas como marcas. Estas características devem ser estáveis e observadas de diferentes pontos de visão e ângulos. [Mouragnon et al. 2006] utilizou o detector de cantos de Harris em um sistema de SLAM onde o objetivo é encontrar a posição e orientação em um sistema de referência global da câmera em tempos t , assim como a posição 3-D das características vistas ao longo da cena. [Davison et al. 2007] utilizou o detector de Shi e Tomasi.

[Salvi et al. 2008] criaram um sistema de SLAM visual onde as características são extraídas com um algoritmo misto de SIFT e SURF sincronizados, para assim obter imagens pré-processadas com um equilíbrio entre quantidade de marcas detectadas, robustez e

complexidade computacional. Em [Zhang et al. 2008] foi proposto um sistema de SLAM visual monocular que detecta características no ambiente utilizando um extrator SURF (*Speeded Up Robust Feature*), e utiliza um FKE (Filtro de Kalman Estendido) para calcular os estados da câmera e das marcas visuais encontradas. [Sünderhalf et al. 2007] também utilizou o detector SURF para obter os pontos de interesse.

3.3 Inicialização das marcas

Após a etapa de extração dos pontos de interesse na imagem, parte desses pontos serão determinados como marcas que serão utilizadas pelo sistema de SLAM visual. Existem duas técnicas principais para realizar tal inicialização, que serão mostradas a seguir.

3.3.1 Inicialização direta

Para conseguir informações como direção e profundidade de características logo na inicialização do sistema de SLAM, uma alternativa é utilizar visão estéreo. [Lemaire et al. 2007b] apresentaram um sistema SLAM visual que utiliza visão estéreo para robôs terrestres e aéreos. Uma outra forma de fazer uma inicialização direta de marcas é usar um alvo artificial de aparência conhecida para inicializar o sistema de SLAM.

[Davison et al. 2007] utilizaram um retângulo sólido impresso em um papel como alvo para inicialização. [Civera et al. 2008] mostrou um sistema de SLAM visual 3-D totalmente automático que utiliza somente uma câmera hand-held sem sensores adicionais. Os experimentos mostraram a robustez do sistema em ambientes internos e externos com um alcance alto para profundidade da cena, movimento variado e também um giro 360° em tempo real. A inicialização direta de marcas é realizada utilizando uma técnica de inversão de profundidade parametrizada.

3.3.2 Inicialização atrasada

Quando utiliza-se um sistema de SLAM visual monocular, não é possível determinar a profundidade das características com a primeira imagem adquirida. Para isso, a câmera deve mover-se para diferentes pontos de visão e utilizar a correlação de características entre diferentes quadros, para assim utilizar alguma técnica para a inicialização das marcas para o sistema de SLAM. Os trabalhos de [Kim & Sukkarieh 2003, Bryson & Sukkarieh 2005] utilizaram inicialização atrasada, onde características que são visualizadas em diferentes imagens consecutivas são tornadas marcas visuais e suas posições no mundo são estimadas.

No trabalho de [Zhang et al. 2008] é utilizado um filtro de partículas para inicializar as marcas. Partículas, denominadas de marcas virtuais, são distribuídas com a mesma prioridade em uma linha reta 3-D que passa pela característica encontrada onde cada uma destas possui uma matriz de covariância. Esta matriz de covariância define uma área elíptica para pesquisa na imagem. Nos passos seguintes, a prioridade das partículas é alterada calculando-se a verossimilhança de cada partícula. Finalmente, a marca virtual com maior probabilidade é transformada em uma marca real.

3.4 Técnicas de SLAM

Existem diferentes técnicas que foram criadas para solucionar o problema de SLAM. Nesta subseção, serão apresentadas as técnicas de filtro de Kalman estendido e filtro de partículas.

3.4.1 SLAM com Filtro de Kalman Estendido - FKE SLAM

O FKE estima a posição do robô através da observação de características presentes no ambiente. As características a serem utilizadas pelo FKE neste trabalho são detectadas e rastreadas, utilizando a técnica SURF citada na Seção 2.2.2.

O FKE utilizado para solucionar o problema de SLAM está sujeito a muitos problemas quando usado para navegação de robô móveis como: complexidade computacional, associação dos dados e não-linearidades. Sob condições ideais para o FKE SLAM, a covariância da estimativa da localização do robô e as posições individuais das marcas convergiriam para zero. Contudo, a complexidade computacional da etapa de correção cresce quadraticamente com o número de marcas [Sim et al. 2005].

A técnica de FKE SLAM é extremamente sensível à associação incorreta das marcas com as novas observações. O problema da associação dos dados é muito difícil para o SLAM, no caso em que as referências são re-observadas a partir de pontos de observação distintos, e a linearização do modelo não-linear do movimento pode potencializar resultados com soluções inconsistentes [Thrun et al. 2006].

A utilização do algoritmo de SLAM baseado no filtro de Kalman estendido é uma das alternativas mais utilizadas para solucionar o problema de SLAM. Algumas implementações de FKE para SLAM visual podem ser encontradas em [Davison et al. 2007, Lemaire & Lacroix 2007a, Lemaire et al. 2007b, Zhang et al. 2008, Salvi et al. 2008, Artieda et al. 2009].

3.4.2 SLAM com Filtro de Partículas - FP SLAM

Neste tipo de filtro, as amostras de uma distribuição posterior são chamadas *partículas* e são representadas por

$$\chi_t := x_t^{[1]}, x_t^{[2]}, \dots, x_t^{[M]} \quad (3.1)$$

onde cada partícula $x_t^{[\psi]}$ (com $1 \leq \psi \leq M$) é uma possibilidade de como estaria o estado no tempo t . M representa o número de partículas do conjunto χ_t . A principal idéia do filtro de partículas é aproximar a crença $bel(x_t)$ através de um conjunto de partículas χ_t [Thrun et al. 2006].

O Algoritmo 1 mostra o funcionamento de um filtro de partículas. Primeiramente, é construído um conjunto temporário de partículas $\bar{\chi}_t$ que representa a crença $\bar{bel}(x_t)$. Isto é feito processando cada partícula $x_{t-1}^{[\psi]}$ no conjunto de entrada χ_t . A função *distribuir* (linha 3) é responsável por distribuir as partículas x_{t-1} , com um sinal de controle u_t , gerando

Algoritmo 1 FILTRO DE PARTÍCULAS**Entrada:** χ_{t-1}, u_t, z_t

```

1:  $\bar{\chi}_t = \chi_t = 0$ ;
2: for  $m = 1$  to  $M$  do
3:   distribuir( $x_t^{[\psi]} \sim p(x_t | u_t, x_{t-1})$ );
4:    $w_t^{[\psi]} = p(z_t | x_t^{[\psi]})$ ;
5:    $\bar{\chi}_t = \bar{\chi}_t + \langle x_t^{[\psi]}, w_t^{[\psi]} \rangle$ ;
6: end for
7: for  $i = 1$  to  $M$  do
8:   redistribuir( $x_t^{[i]}, w_t^{[i]}$ );
9:   adicionar( $x_t^{[i]}, \chi_t$ );
10: end for

```

Retorno: χ_t ;

um estado hipotético x_t . O conjunto obtido após M iterações é a representação da crença $\bar{bel}(x_t)$ pelo filtro. A linha 4 calcula o *fator de importância* $w_t^{[\psi]}$ para cada partícula $x_t^{[\psi]}$. Este é usado para incorporar a medida z_t dentro do conjunto de partículas.

A parte mais importante do algoritmo do filtro de Partículas ocorre entre as linhas 7 e 10, pois é a etapa de *redistribuição* das partículas. Esta etapa é responsável por redistribuir o conjunto temporário de partículas $\bar{\chi}_t$ de acordo com o fator de importância $w_t^{[\psi]}$, forçando assim as partículas para a crença posterior $bel(x_t)$.

O filtro de partículas foi aplicado com sucesso para a resolver o problema de SLAM. O algoritmo FastSLAM, introduzido por [Montemerlo et al. 2002], utilizou um filtro de partículas para estimar a pose do robô e um filtro de Kalman estendido para estimar a localização das marcas. Este algoritmo é baseado no problema de que a posição do robô é conhecida, e a localização das marcas é independente.

Para um estado do robô no tempo t , é calculado para cada partícula o novo estado do robô usando o estado no tempo $t-1$ e a entrada de controle u_t . Isto cria um conjunto temporário de partículas, cada uma recebe um peso diferente, que representa o estado do robô no instante t . Para cada partícula, cada marca é atualizada utilizando um filtro de Kalman estendido.

Os trabalhos de [Nieto et al. 2003, Montemerlo et al. 2002, Hähnel et al. 2003, Sim et al. 2005, Elinas et al. 2006] utilizaram um sistema de SLAM visual baseado no filtro de partículas. [Zhang et al. 2008] utilizou um filtro de partículas para a inicialização atrasada de marcas visuais.

3.4.3 Outras técnicas

[Julier & Uhlmann 1997] propõem a utilização do UKF - *Unscented Kalman Filter* - que realiza uma linearização estocástica através do uso de um processo de regressão linear estatística utilizando pesos. Ao invés de aproximar uma função g através de uma

expansão de série de Taylor, como no filtro de Kalman estendido, o UKF extrai deterministicamente os pontos chamados *sigma* de Gaussianas e passa estes através de g [Thrun et al. 2006]. No trabalho de [Sünderhalf et al. 2007] foi apresentado um sistema de SLAM visual monocular que utiliza a técnica de UKF com uma parametrização de profundidade invertida para controle de dirigíveis autônomos.

O *FastSLAM* é uma abordagem ao SLAM que é utilizada para solucionar o problema da evolução exponencial da complexidade computacional quanto ao aumento de referências espaciais, na utilização do filtro de partículas [Bauer et al. n.d.]. Através da utilização de estruturas de dados eficientes, o *FastSLAM* necessita de um tempo de atualização do mapa de $O(M \log N)$, onde M é o número de partículas e N é o número de referências espaciais.

No trabalho de [Barfoot 2005], o *FastSLAM* foi utilizado juntamente com um sistema de câmeras estéreo e extração de referências por SIFT para gerar estimativas de posicionamento a $3Hz$, o que gerou um erro de 4% na localização do robô após a distância total ser percorrida.

Capítulo 4

Técnica Proposta

UM dos principais desafios na navegação de um robô móvel autônomo é localizar e mapear simultaneamente o ambiente no qual o robô navega. Neste trabalho é apresentada uma técnica de SLAM visual monocular baseada no filtro de Kalman estendido, que utiliza características encontradas em uma sequência de imagens através do descritor SURF (apresentado na Seção 2.2.2) e determina quais características podem ser utilizadas como marcas através de uma técnica de inicialização atrasada baseada em retas 3-D. Conforme novas marcas vão surgindo no ambiente, estas são adicionadas ao vetor de estados do filtro de Kalman estendido. Segundo [Gil et al. 2009], o detector e descritor SURF é a melhor técnica a ser utilizada no SLAM visual, porque tem custo computacional reduzido, facilitando a extração de características em tempo real.



Figura 4.1: Robô móvel utilizado para a experiência.

Os testes foram realizados com o robô móvel *Karel* (Figura 4.1) que possui dois encoders acoplados aos eixos dos motores e uma câmera dedicada para o sistema de localização. O ambiente de navegação é interno, e sujeito a variações de iluminação. O sistema de visão proposto está sujeito também a falhas sensoriais da câmera, ou seja, em alguns momentos as imagens podem apresentar ruídos causando problemas na correlação

dos pontos de interesse e assim fornecendo dados ruidosos para o filtro (Figura 4.2). A posição da câmera em relação ao sistema de referências do robô é conhecida. Os dados obtidos com o FKE são utilizados para atualizar a pose do robô e são comparados aos dados fornecidos pela odometria do robô móvel, obtida com os dados fornecidos pelos encoders óticos. Como a odometria determina a localização do robô com base no acúmulo das informações obtidas nos instantes anteriores, um erro em um determinado instante afeta as medições dos instantes seguintes.

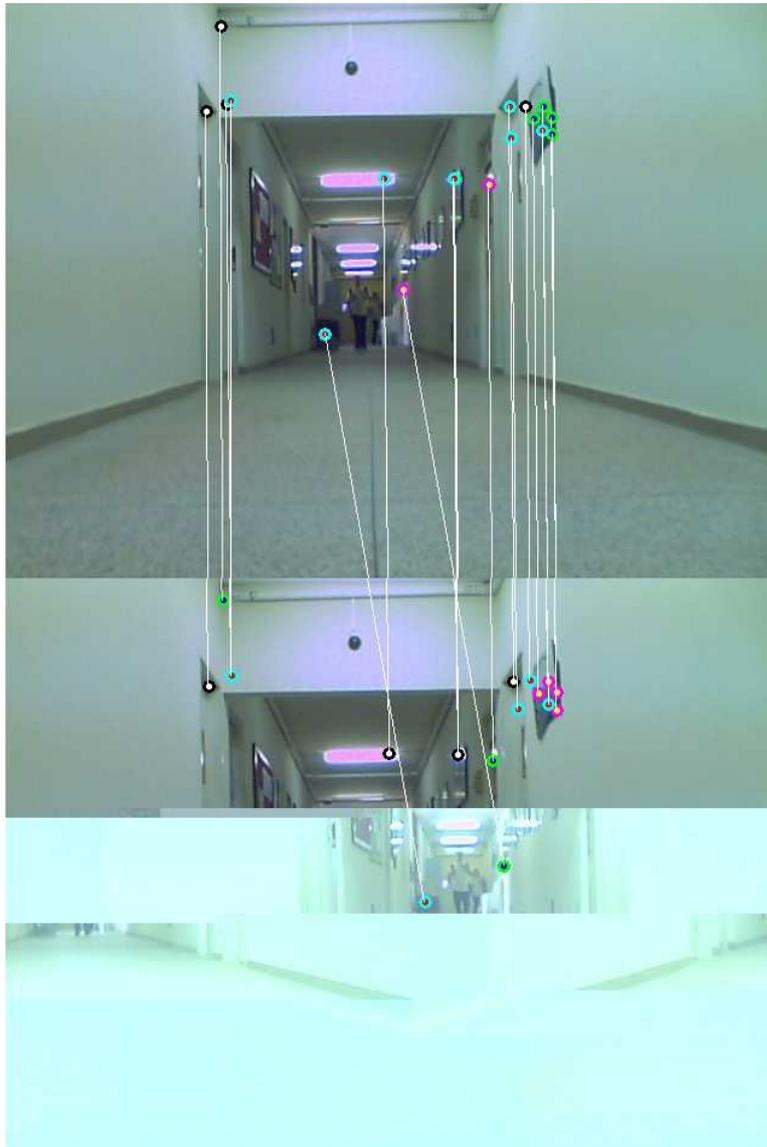


Figura 4.2: Problema presente durante o percurso realizado pelo robô.

4.1 Caracterização do problema

Para iniciar o mapeamento e localização, é necessário primeiro determinar a posição das marcas presentes no ambiente. Para isso, tem-se disponível apenas as coordenadas $I(u, v)$ das características detectadas na imagem e os parâmetros intrínsecos e extrínsecos da câmera. É possível determinar a posição das marcas somente com a disponibilidade da informação da profundidade destas.

Com a utilização de um sistema de visão monocular, não é possível obter a profundidade de forma direta, como em um sistema de visão estéreo. Existe uma técnica denominada de *inicialização atrasada de marcas*, que utiliza informações sobre o movimento da câmera e de várias medidas da mesma marca visual realizadas de diferentes pontos de vista, para estimar a profundidade de uma característica. Entretanto, não é adequado demorar muitos quadros de imagem para poder inicializar uma marca, porque isso pode comprometer o sistema de localização do robô.

O ambiente para testar esta técnica (Figura 4.3) foi utilizado da seguinte forma. O robô foi colocado para se movimentar em linha reta, sempre visualizando os padrões de tabuleiro de xadrez. Dessa forma, foi testado a técnica de inicialização atrasada com partículas e a técnica proposta. Após testes exaustivos, concluiu-se que além de maior velocidade na inicialização das marcas visuais, a técnica proposta apresentou maior eficácia, porque o erro entre a posição real e a estimada das marcas foi menor do que com a utilização de partículas.

Nas técnicas propostas por [Davison et al. 2007, Zhang et al. 2008], um filtro de partículas é utilizado para calcular a profundidade da marca. É estabelecido um conjunto de partículas linearmente espaçado ao longo de uma linha 3-D entre o centro ótico da câmera e o ponto $M(X, Y, Z)$ no mundo que corresponde à característica $I(u, v)$ da imagem, com uma distância máxima estabelecida em metros. Todas as partículas são inicializadas com a mesma probabilidade. A probabilidade para cada uma dessas partículas é atualizada no próximo passo, quando é feita uma estimativa de onde tal característica $I(u, v)$ apareceu na imagem anterior.

Se a probabilidade das partículas formar uma distribuição gaussiana aproximada, a partícula com maior probabilidade é tornada uma marca. Obtendo-se uma marca no ambiente, é verificado se esta é uma marca nova ou uma marca já conhecida. Se for uma marca nova, é adicionada ao vetor de estados do FKE. Senão, é atualizada a pose do robô e das marcas. A Figura 4.4 mostra os resultados obtidos para uma determinada marca em um raio de $10m$, onde a distância entre $M(X, Y, Z)$ real e o $M(X, Y, Z)$ calculado de uma marca foi de $0,52m$ (D.M. na legenda da figura), com um erro de $5,2\%$ (E.M. na legenda da figura).

4.2 Solução proposta

A inicialização atrasada utilizando partículas fornece resultados satisfatórios para o SLAM visual, porém pode apresentar os seguintes problemas:

- O tempo computacional é da ordem de n (número de partículas), se o número de



Figura 4.3: Ambiente de teste da técnica de inicialização atrasada.

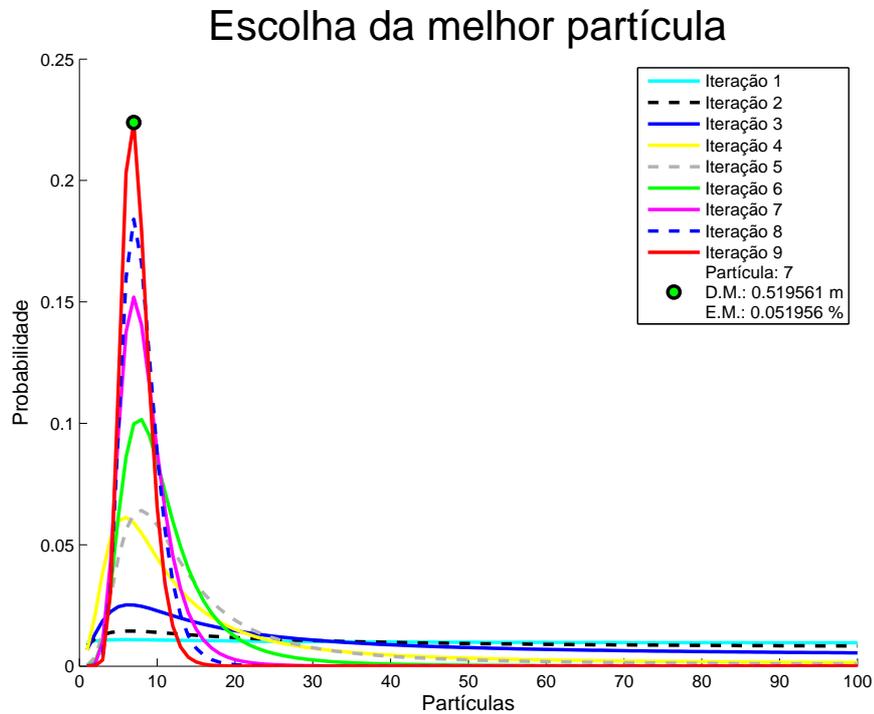


Figura 4.4: Probabilidade das partículas após 9 iterações, calculada utilizando 100 partículas que estimam a profundidade de uma marca em até 10 metros.

- partículas aumentar, o esforço computacional também cresce;
- Se existir a necessidade de identificar marcas a grandes distâncias (50 metros, por exemplo), existem dificuldades porque estas marcas irão demorar para serem adicionadas ao vetor de estados ou até mesmo não irão ser detectadas por possuírem uma baixa paralaxe;
 - Nesta técnica há a necessidade de inicialização prévia de parâmetros, como o número de partículas e a distância máxima de alcance de marcas. Ou seja, para um funcionamento correto da técnica de SLAM visual, deve-se informar parâmetros de acordo com o ambiente de navegação do robô.

4.2.1 Técnica de inicialização atrasada utilizando retas 3-D

O método proposto neste trabalho calcula a posição das marcas com inicialização atrasada, porém com uma maior rapidez (em relação à inicialização atrasada com partículas) devido a estimação da profundidade ser feita de forma analítica. No passo inicial, é calculada a equação paramétrica de uma reta 3-D R_1 que passa pela marca e pelo centro óptico da câmera. Nos passos seguintes, são obtidos os novos parâmetros das retas 3-D R_N que passam pela mesma marca, e a partir do método dos mínimos quadrados encontra-se um ponto $M(X, Y, Z)$ que representa a menor distância entre estas retas (Ver Figura 4.5). O ponto a ser inicializado como marca é o ponto médio entre os pontos encontrados. A seguir, o modelo matemático da técnica será mostrado nas equações 4.1 e 4.3.

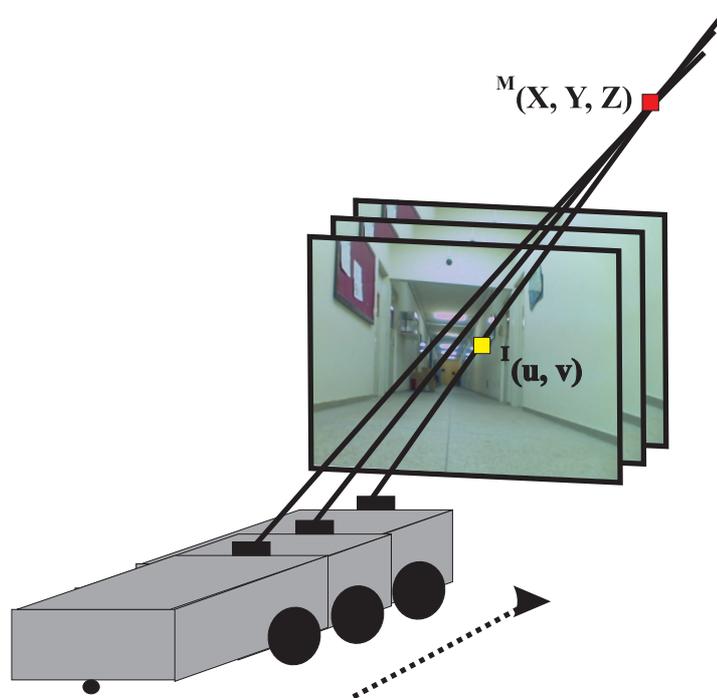


Figura 4.5: Movimentação do robô com a visualização de uma mesma marca e a intersecção das retas para a estimação da posição desta marca.

Sendo P_1, P_2, \dots, P_N pontos pertencentes às retas:

$$\text{Equações Paramétricas das Retas} \begin{cases} R_1 \implies P_1 = PI_1 + \delta_1 * t_1 \\ R_2 \implies P_2 = PI_2 + \delta_2 * t_2 \\ \vdots \\ R_N \implies P_N = PI_N + \delta_N * t_N \end{cases} \quad (4.1)$$

onde:

- $1 \leq K \leq N$;
- $P_K = [P_{Kx} \ P_{Ky} \ P_{Kz}]^T$: é um ponto pertencente à reta K ;
- $PI_K = [PI_{Kx} \ PI_{Ky} \ PI_{Kz}]^T$: é a posição do centro ótico $C(0,0,0)$ da câmera mapeado em coordenadas do mundo, obtido a partir de dados da odometria do robô;
- $\delta_K = [\delta_{Kx} \ \delta_{Ky} \ \delta_{Kz}]^T = PF_K - PI_K$.

Como tem-se a informação de $I(u, v)$, centro da imagem (CX, CY) e foco da câmera F , foi possível estimar um ponto final PF_N pertencente a reta R_N . Para isto, assumiu-se que a profundidade de tal ponto final fosse grande o bastante para que a marca a ser estimada estivesse compreendida entre o PI_N e PF_N .

Como a marca é obtida a partir da intersecção das retas, tem-se que:

$$P_1 = P_2 = P_3 = \dots = P_N \implies P_2 - P_1 = P_3 - P_1 = \dots = P_N - P_1 \quad (4.2)$$

Assim:

$$\begin{bmatrix} \delta_{1x} & -\delta_{2x} & 0 & \dots & 0 \\ \delta_{1y} & -\delta_{2y} & 0 & \dots & 0 \\ \delta_{1z} & -\delta_{2z} & 0 & \dots & 0 \\ \delta_{1x} & 0 & -\delta_{3x} & \dots & 0 \\ \delta_{1y} & 0 & -\delta_{3y} & \dots & 0 \\ \delta_{1z} & 0 & -\delta_{3z} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \delta_{1x} & 0 & 0 & \dots & -\delta_{Nx} \\ \delta_{1y} & 0 & 0 & \dots & -\delta_{Ny} \\ \delta_{1z} & 0 & 0 & \dots & -\delta_{Nz} \end{bmatrix} * \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_N \end{bmatrix} = \begin{bmatrix} PI_{2x} - PI_{1x} \\ PI_{2y} - PI_{1y} \\ PI_{2z} - PI_{1z} \\ PI_{3x} - PI_{1x} \\ PI_{3y} - PI_{1y} \\ PI_{3z} - PI_{1z} \\ \vdots \\ PI_{Nx} - PI_{1x} \\ PI_{Ny} - PI_{1y} \\ PI_{Nz} - PI_{1z} \end{bmatrix} \quad (4.3)$$

De posse dos valores de t_1, t_2, \dots, t_N , substituiu-se os valores nas equações paramétricas das retas, para obter os valores de P_1, P_2, \dots, P_N . A partir da média entre esses pontos, determina-se a localização da marca desejada.

A técnica de inicialização proposta neste trabalho resolve estes problemas existentes na inicialização atrasada com partículas, porque não há a necessidade de uma inicialização prévia de parâmetros como número de partículas e distância máxima de alcance. A

Figura 4.6 mostra como as retas são representadas nas coordenadas de mundo, e mostra que a marca para qual foi estimada a profundidade pelo método das partículas, a distância entre $M(X, Y, Z)$ real e o $M(X, Y, Z)$ calculado foi de 0,11 m. O erro máximo obtido na estimação de profundidade foi de 0,383m e o erro mínimo foi de 0,002m, considerando 56 marcas em um padrão situado a 4 metros de distância do robô.

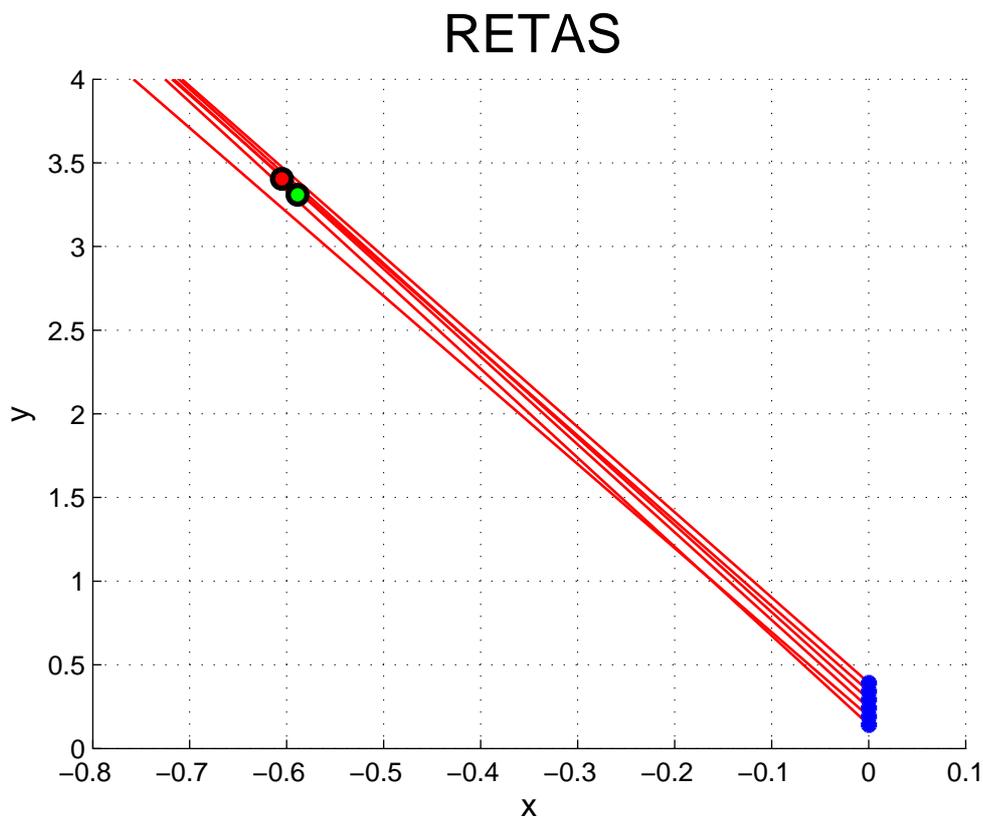


Figura 4.6: Retas traçadas para uma marca vista da Imagem 1 até a Imagem 6. O ponto verde representa $M(X, Y, Z)$ calculado, o ponto vermelho representa $M(X, Y, Z)$ real e os pontos em azul representam os pontos iniciais PI .

4.2.2 Localização do robô

Como mencionado na Seção 2.3, o problema de localização e mapeamento simultâneos surge quando um robô móvel não tem acesso a um mapa do ambiente de navegação, nem conhece a sua própria localização. Para solucionar tal problema, foi desenvolvido um sistema capaz de localizá-lo em um ambiente a partir de marcas visuais que são determinadas a partir da técnica SURF de extração de pontos de interesse na imagem e da

inicialização atrasada de marcas com retas 3-D. A identificação de marcas, juntamente com o modelo de odometria do robô, foram incorporados em um filtro de Kalman estendido a fim de obter sua pose corrigida.

Como a fonte sensorial utilizada retorna um valor ${}^I P_{marca}$, deve-se fazer um mapeamento de cada marca em coordenadas de mundo para coordenadas de imagem. Para isso, devem-se seguir os seguintes passos:

- **Mapear uma marca ${}^M P_{marca}$ para ${}^R P_{marca}$:**

$${}^R P_{marca} = ({}^M R_R)^T * ({}^M P_{marca} - {}^M P_R) \quad (4.4)$$

- **Mapear uma marca ${}^R P_{marca}$ para ${}^C P_{marca}$:**

$${}^C P_{marca} = {}^C R_R * {}^R P_{marca} + {}^C P_R \quad (4.5)$$

- **Mapear uma marca ${}^C P_{marca}$ para ${}^I P_{marca} = {}^I(u, v)$:**

$$u = CX + F * ({}^C X_{marca} / {}^C Z_{marca}) \quad (4.6)$$

$$v = CY + F * ({}^C Y_{marca} / {}^C Z_{marca}) \quad (4.7)$$

onde:

- ${}^M R_R = \begin{bmatrix} \cos({}^M \theta_R) & -\sin({}^M \theta_R) & 0 \\ \sin({}^M \theta_R) & \cos({}^M \theta_R) & 0 \\ 0 & 0 & 1 \end{bmatrix}$: matriz de rotação entre robô e mundo;
- ${}^M P_R = [{}^M X_R \quad {}^M Y_R \quad {}^M Z_R]^T$: posição do robô mapeada nas coordenadas de mundo;
- ${}^C R_R = \begin{bmatrix} 0.999576 & -0.015093 & 0.024887 \\ 0.025759 & 0.060598 & -0.997830 \\ 0.013552 & 0.998048 & 0.060962 \end{bmatrix}$: matriz de rotação entre robô e câmera;
- ${}^C P_R = [-0.006117 \quad 0.304493 \quad -0.160543]$: posição do robô mapeada nas coordenadas de câmera;
- ${}^C P_{marca} = [{}^C X_{marca} \quad {}^C Y_{marca} \quad {}^C Z_{marca}]^T$;

Fase de predição

A odometria é um método clássico utilizado para calcular a pose de um robô. Neste trabalho foram utilizados *encoders* óticos que mediram as rotações das rodas do robô, e assim foi possível calcular a sua pose através de um modelo cinemático. [Thrun et al. 2006] propõem que as informações de odometria não funcionem como medidas sensoriais, mas que elas sejam incorporadas ao modelo do robô. Para maiores detalhes sobre o modelo de odometria do movimento do robô, consultar [Santana 2007].

Fase de atualização

O robô móvel navega em um ambiente onde a posição das marcas no mundo é conhecida e a cada passo o descritor SURF atualiza ${}^I P_{marca}$. Partindo-se das transformações que mapeiam um ponto no mundo para coordenadas de imagem, tem-se que:

$$\left\{ \begin{array}{l} {}^C X_{marca} = ({}^M X_{marca} - {}^M X_R) * ({}^C R_R(1,1) * \cos({}^M \theta_R) - {}^C R_R(1,2) * \sin({}^M \theta_R)) \\ \quad + ({}^M Y_{marca} - {}^M Y_R) * ({}^C R_R(1,1) * \sin({}^M \theta_R) + {}^C R_R(1,2) * \cos({}^M \theta_R)) \\ \quad + {}^C R_R(1,3) * {}^M Z_{marca} + {}^C P_R(1) \\ \\ {}^C Y_{marca} = ({}^M X_{marca} - {}^M X_R) * ({}^C R_R(2,1) * \cos({}^M \theta_R) - {}^C R_R(2,2) * \sin({}^M \theta_R)) \\ \quad + ({}^M Y_{marca} - {}^M Y_R) * ({}^C R_R(2,1) * \sin({}^M \theta_R) + {}^C R_R(2,2) * \cos({}^M \theta_R)) \\ \quad + {}^C R_R(2,3) * {}^M Z_{marca} + {}^C P_R(2) \\ \\ {}^C Z_{marca} = ({}^M X_{marca} - {}^M X_R) * ({}^C R_R(3,1) * \cos({}^M \theta_R) - {}^C R_R(3,2) * \sin({}^M \theta_R)) \\ \quad + ({}^M Y_{marca} - {}^M Y_R) * ({}^C R_R(3,1) * \sin({}^M \theta_R) + {}^C R_R(3,2) * \cos({}^M \theta_R)) \\ \quad + {}^C R_R(3,3) * {}^M Z_{marca} + {}^C P_R(3) \end{array} \right. \quad (4.8)$$

O modelo do sistema 4.8 é incorporado ao filtro de Kalman através da matriz jacobiana $H_R(2, M)$ (Equação 4.9), onde $M \in [1, 3]$.

$$H_R = \frac{\partial {}^I P_{marca}}{\partial ({}^M X_R, {}^M Y_R, {}^M \theta_R)} = \left\{ \begin{array}{l} \frac{\partial u}{\partial ({}^M X_R)} = F * \frac{k1 * {}^C Z_{marca} - k2 * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial u}{\partial ({}^M Y_R)} = F * \frac{k3 * {}^C Z_{marca} - k4 * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial u}{\partial ({}^M \theta_R)} = F * \frac{k5 * {}^C Z_{marca} - k6 * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M X_R)} = F * \frac{k7 * {}^C Z_{marca} - k2 * {}^C Y_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M Y_R)} = F * \frac{k8 * {}^C Z_{marca} - k4 * {}^C Y_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M \theta_R)} = F * \frac{k9 * {}^C Z_{marca} - k6 * {}^C Y_{marca}}{({}^C Z_{marca})^2} \end{array} \right. \quad (4.9)$$

onde:

$$\begin{aligned}
k1 &= {}^C R_R(1,2) * \sin({}^M \theta_R) - {}^C R_R(1,1) * \cos({}^M \theta_R) \\
k2 &= {}^C R_R(3,2) * \sin({}^M \theta_R) - {}^C R_R(3,1) * \cos({}^M \theta_R) \\
k3 &= -{}^C R_R(1,1) * \sin({}^M \theta_R) - {}^C R_R(1,2) * \cos({}^M \theta_R) \\
k4 &= -{}^C R_R(3,1) * \sin({}^M \theta_R) - {}^C R_R(3,2) * \cos({}^M \theta_R) \\
k5 &= -\sin({}^M \theta_R) * [{}^C R_R(1,1) * ({}^M X_{marca} - {}^M X_R) + {}^C R_R(1,2) * ({}^M Y_{marca} - {}^M Y_R)] \\
&\quad + \cos({}^M \theta_R) * [{}^C R_R(1,1) * ({}^M Y_{marca} - {}^M Y_R) - {}^C R_R(1,2) * ({}^M X_{marca} - {}^M X_R)] \\
k6 &= -\sin({}^M \theta_R) * [{}^C R_R(3,1) * ({}^M X_{marca} - {}^M X_R) + {}^C R_R(3,2) * ({}^M Y_{marca} - {}^M Y_R)] \\
&\quad + \cos({}^M \theta_R) * [{}^C R_R(3,1) * ({}^M Y_{marca} - {}^M Y_R) - {}^C R_R(3,2) * ({}^M X_{marca} - {}^M X_R)] \\
k7 &= {}^C R_R(2,2) * \sin({}^M \theta_R) - {}^C R_R(2,1) * \cos({}^M \theta_R) \\
k8 &= -{}^C R_R(2,1) * \sin({}^M \theta_R) - {}^C R_R(2,2) * \cos({}^M \theta_R) \\
k9 &= -\sin({}^M \theta_R) * [{}^C R_R(2,1) * ({}^M X_{marca} - {}^M X_R) + {}^C R_R(2,2) * ({}^M Y_{marca} - {}^M Y_R)] \\
&\quad + \cos({}^M \theta_R) * [{}^C R_R(2,1) * ({}^M Y_{marca} - {}^M Y_R) - {}^C R_R(2,2) * ({}^M X_{marca} - {}^M X_R)]
\end{aligned}$$

As marcas utilizadas são um conjunto de pontos de interesse encontrados no ambiente de navegação do robô móvel. Tendo definidas as marcas, estas são representadas no vetor de estados como ${}^M P_{marca} = ({}^M X_{marca}, {}^M Y_{marca}, {}^M Z_{marca})$. A covariância Σ_{marca} de cada marca é inicializada utilizando parâmetros empíricos, obtidos depois de testes que mostraram que distância entre a marca e o robô tem influência em Σ_{marca} e que a variância em X e Z é proporcionalmente menor que a variância em Y :

$$\Sigma_{marca} = \begin{bmatrix} (e^{(dist/40)-1})^2 & 0 & 0 \\ 0 & (e^{(dist/10)-1})^2 & 0 \\ 0 & 0 & (e^{(dist/40)-1})^2 \end{bmatrix}$$

onde $dist$ é a distância euclidiana entre a posição da marca e o robô.

O modelo do sistema 4.8 é incorporado ao filtro de Kalman através da matriz jacobiana $H_{marca}(2, K)$, onde K varia de 4 até N (tamanho do vetor de estados).

$$H_{marca} = \frac{\partial {}^I P_{marca}}{\partial ({}^M X_{marca}, {}^M Y_{marca}, {}^M Z_{marca})} = \begin{cases} \frac{\partial u}{\partial ({}^M X_{marca})} = F * \frac{-k1 * {}^C Z_{marca} + k2 * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial u}{\partial ({}^M Y_{marca})} = F * \frac{-k3 * {}^C Z_{marca} + k4 * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial u}{\partial ({}^M Z_{marca})} = F * \frac{{}^C R_R(1,3) * {}^C Z_{marca} - {}^C R_R(3,3) * {}^C X_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M X_{marca})} = F * \frac{-k7 * {}^C Z_{marca} + k2 * {}^C Y_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M Y_{marca})} = F * \frac{-k8 * {}^C Z_{marca} + k4 * {}^C Y_{marca}}{({}^C Z_{marca})^2} \\ \frac{\partial v}{\partial ({}^M Z_{marca})} = F * \frac{{}^C R_R(2,3) * {}^C Z_{marca} - {}^C R_R(3,3) * {}^C Y_{marca}}{({}^C Z_{marca})^2} \end{cases}$$

No SLAM visual, a matriz de covariância do ruído de medição Q , deve variar de acordo com a distância medida entre a marca e o robô, e com a posição da marca na

imagem. [Wu & Zhang 2007] apresentou um trabalho onde resultados experimentais mostraram uma relação linear entre a variância de Q e a distância. Neste trabalho foi utilizado um valor empírico para a variância de (u, v) , obtido após vários testes.

4.3 Resultados

O sistema de SLAM visual monocular proposto neste trabalho foi testado em um robô móvel que navegou por um corredor sujeito a variação de iluminação. Testes mostraram que a qualidade das características obtidas com o SURF exercem grande influência na localização do robô. Quanto pior a qualidade, maior o ruído na localização do robô. Portanto, só foram selecionadas características em que a distância euclidiana entre os vetores de descrição fosse menor que 0,15, e esse valor é a *taxa* do SURF. As que não obedeceram essa restrição, foram descartadas.

O valor da taxa foi obtido a partir de sucessivos testes, a partir das imagens obtidas no ambiente de navegação. Chegou-se a conclusão que para valores menores a percentagem de características correlacionadas foi muito baixa e para valores maiores a quantidade de falsas correlações aumentava, comprometendo diretamente na técnica de inicialização.

Quando uma característica foi detectada em três imagens consecutivas, esta automaticamente foi inicializada como uma marca, a partir da técnica baseada em retas 3-D, obtendo assim a informação de profundidade. Foram feitos testes de inicialização detectando uma mesma característica por seis imagens consecutivas, porém os resultados não foram satisfatórios devido ao alto índice de reflectância dos objetos presentes no ambiente de navegação do robô (Figura 4.7), o que prejudicou a repetibilidade das características.



Figura 4.7: Reflexo da luz nos objetos prejudicou o *matching* de características.

Os resultados mostraram que o erro de localização obtido com o SLAM visual foi significativamente menor que o erro de localização obtido com odometria (Figuras 4.8 e 4.9).

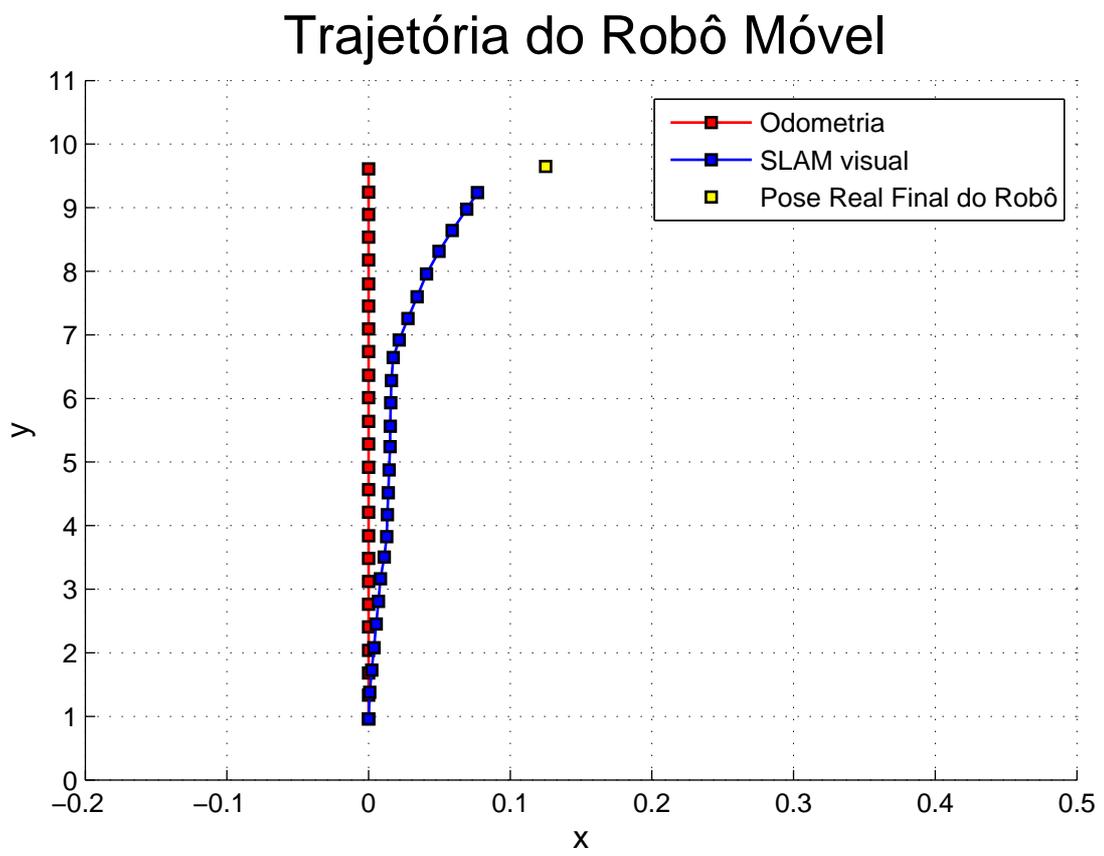


Figura 4.8: Trajetória realizada pelo robô móvel.

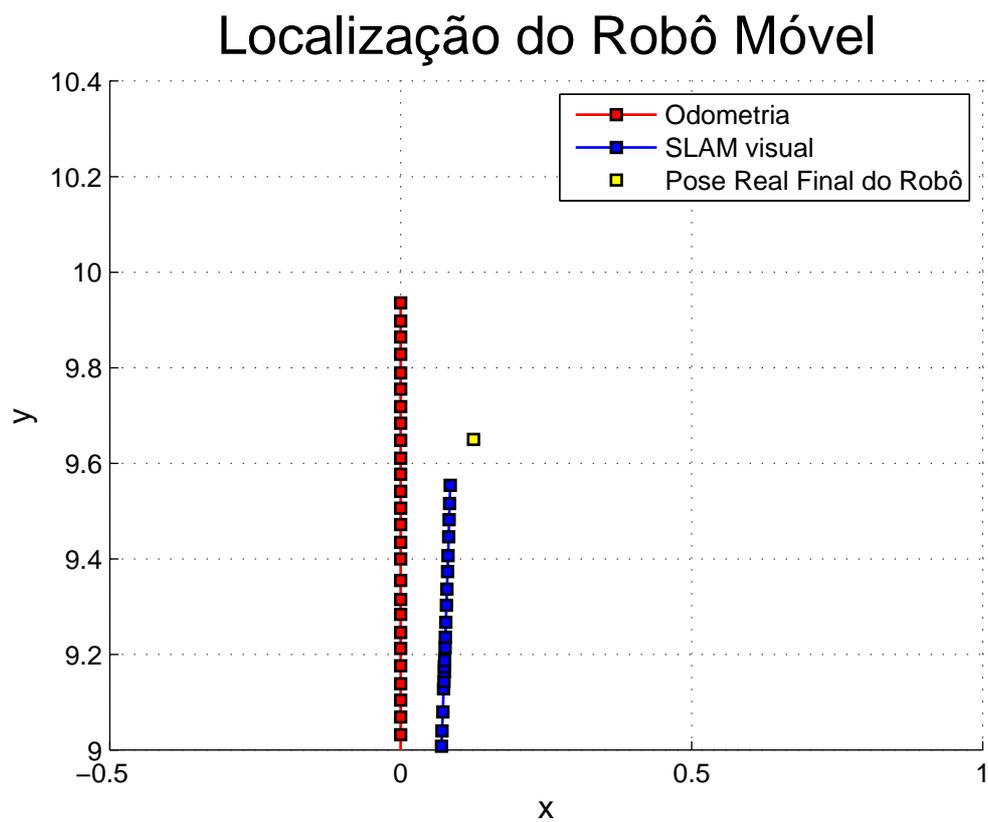
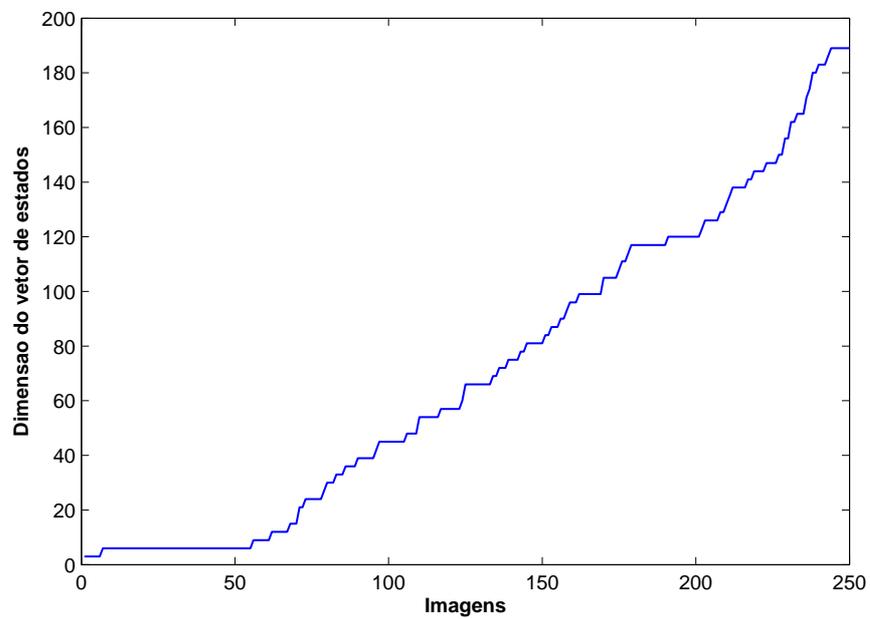


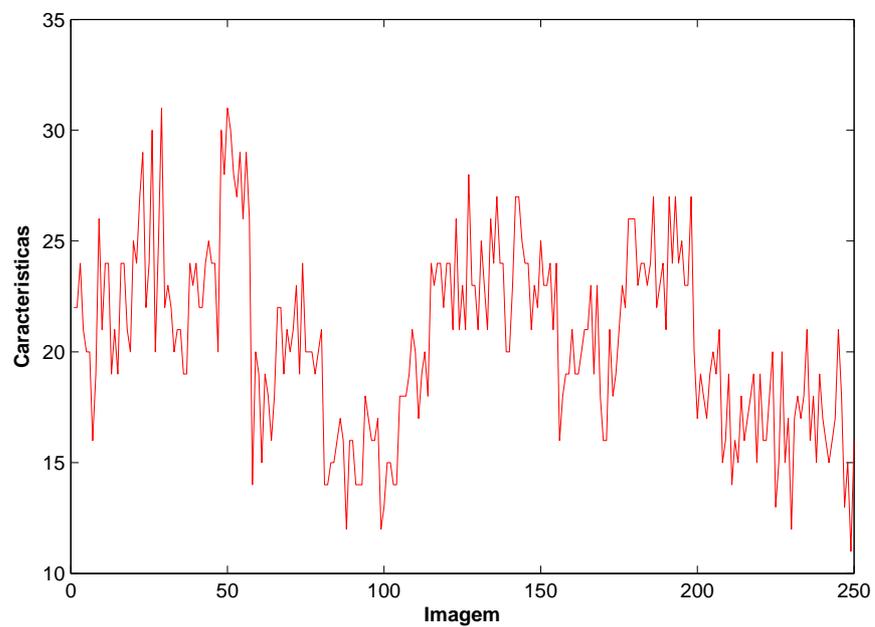
Figura 4.9: Zoom no final da trajetória mostrada na Figura 4.8. Comparação entre a localização por odometria e por SLAM visual. O erro entre a pose final real do robô e a pose dada pela odometria foi de $0,3m$. O erro de localização dado pelo SLAM visual foi de $0,09m$.

A Figura 4.10a mostra o comportamento do vetor de estados. O seu crescimento ocorre devido a adição de novas marcas. Um problema a ser tratado como trabalho futuro é a realização de um sub-mapeamento, a fim de evitar um crescimento demasiado. A Figura 4.10b mostra quantas características foram encontradas para cada imagem obtida e a Figura 4.11 mostra quantas características foram correlacionadas, e a porcentagem de correlação para cada imagem.

As Figuras 4.12 e 4.13 mostram trajetórias geradas realizando uma alteração na taxa do SURF (distância euclidiana entre os vetores de descrição). A Figura 4.14 mostra o comportamento do vetor de estados, a diferença na quantidade de características detectadas e na quantidade de *matches* realizados por imagem. A procura por características mais semelhantes, através da redução da taxa que expressa a distância euclidiana entre os vetores de descrição SURF, não contribuiu para a melhoria dos resultados devido à redução de características encontradas e na dificuldade da correlação proporcionada pela luminosidade no ambiente de teste.



(a) Vetor de estados.



(b) Detecção de pontos de interesse para cada imagem.

Figura 4.10: Comportamento do vetor de estados e a detecção de pontos de interesse durante a movimentação do robô.

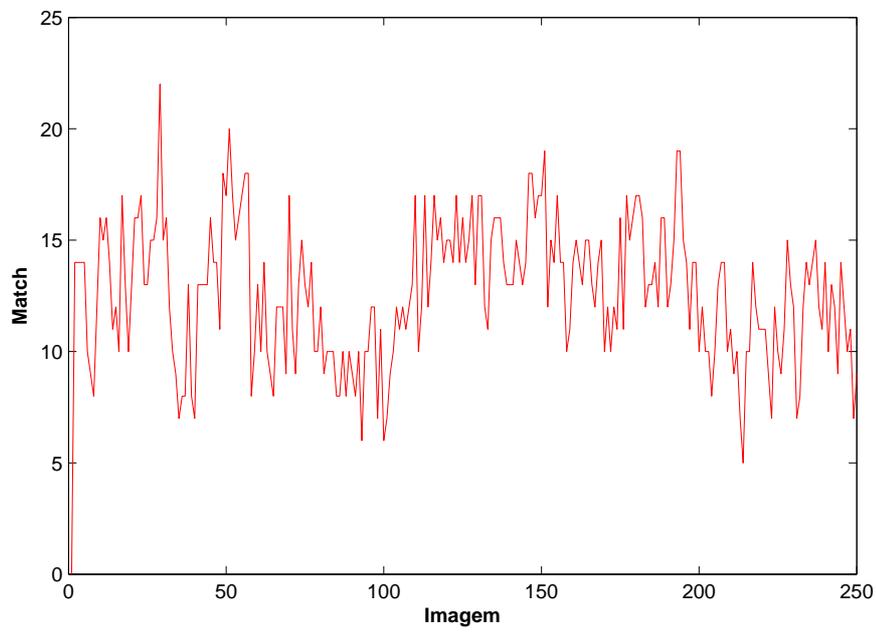
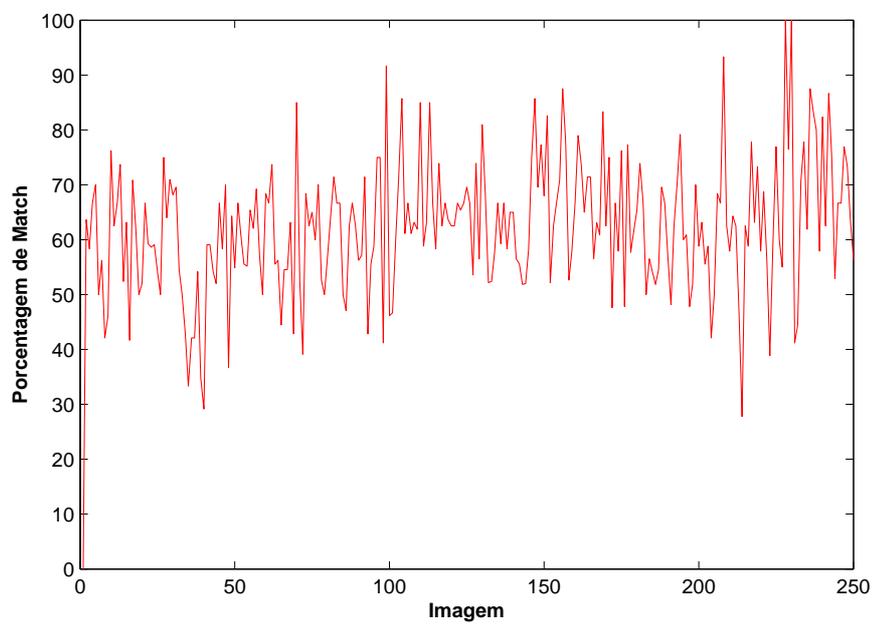
(a) Quantidade de *matches* por imagem(b) Porcentagem de *matches* por imagem

Figura 4.11: Comportamento da correlação de características para cada imagem.

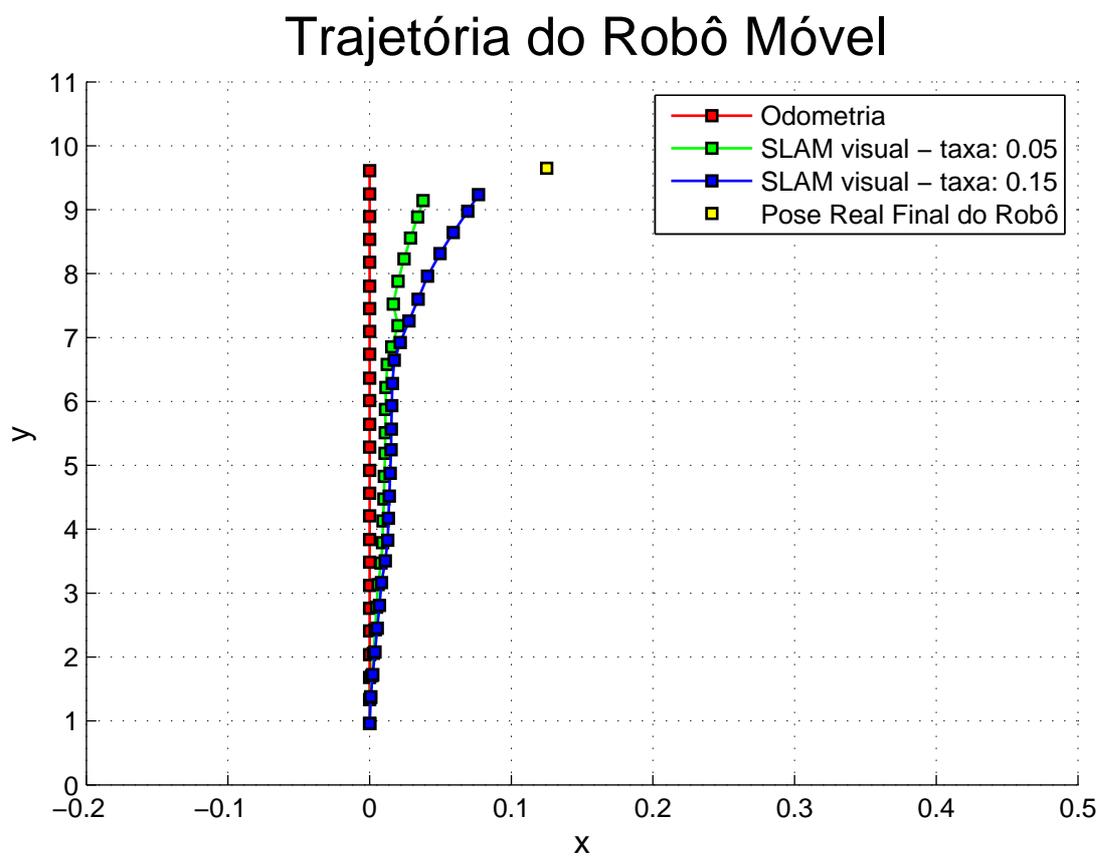


Figura 4.12: Comparação entre as trajetórias calculadas por odometria e SLAM visual.

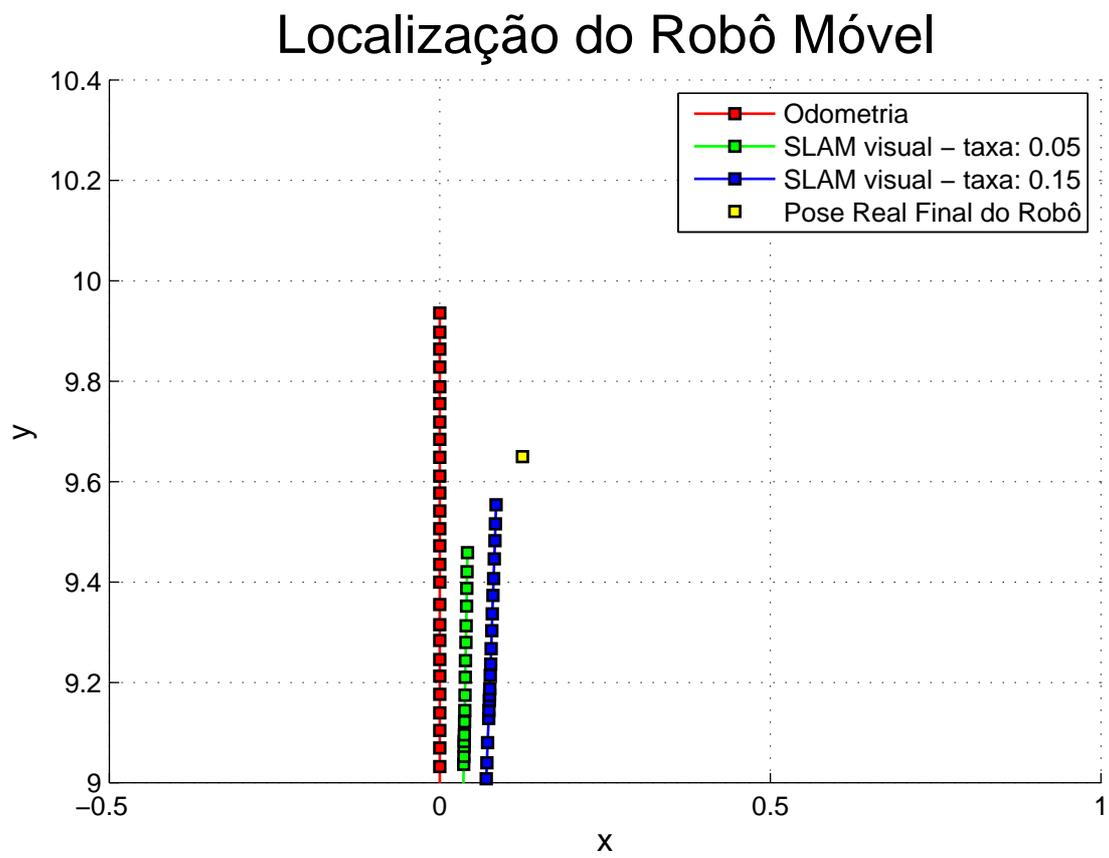
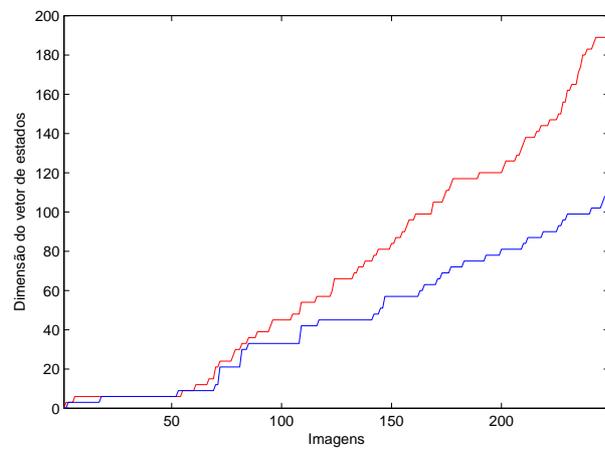
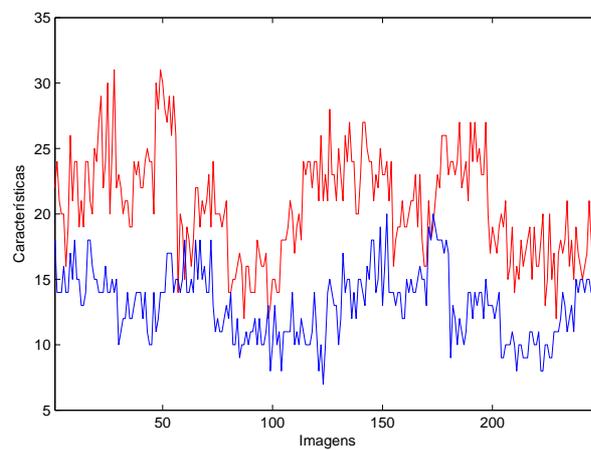


Figura 4.13: Zoom no final das trajetórias mostradas na Figura 4.12.



(a) Vetores de estados.



(b) Características detectadas nas imagens.

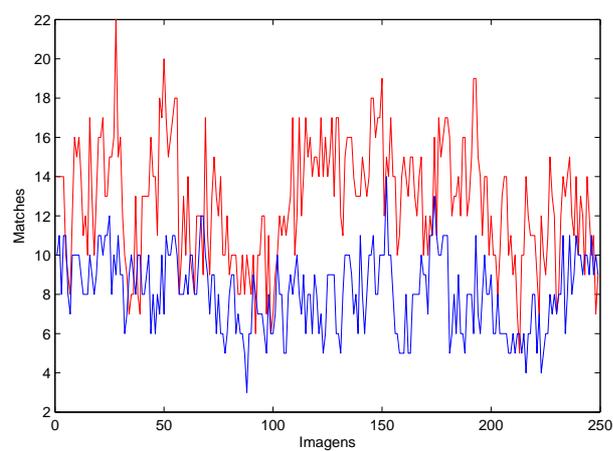
(c) Comportamento dos *matches* nas imagens.

Figura 4.14: Comparação entre o SLAM visual com taxa de 0,05 (em azul) e SLAM visual com taxa de 0,15 (em vermelho).

Capítulo 5

Conclusão e Trabalhos Futuros

E

STE trabalho apresentou um sistema de SLAM visual monocular que utiliza a técnica de inicialização atrasada de marcas para construir um mapa local 3-D do ambiente de navegação. Esse sistema foi testado em um robô móvel, e corrigiu satisfatoriamente a sua pose e a das marcas.

A implementação do sistema de visão foi feita utilizando as funções disponíveis nas bibliotecas *OpenCV* e *OpenSURF*, pois estas proveem uma computação eficiente dos algoritmos que foram utilizados neste trabalho.

As principais contribuições foram: criação de uma nova técnica de inicialização atrasada de marcas baseada na intersecção de retas 3-D, que utiliza o descritor SURF para detectar e correlacionar pontos de interesse nas imagens, com a finalidade de obter informações sobre a profundidade de uma marca visual; a modelagem do sensor para a fase de atualização do filtro de Kalman estendido e a criação de um sistema de SLAM visual monocular que utiliza as técnicas criadas com aplicação para robôs móveis.

Para fins de comparação, foi implementada a técnica de inicialização atrasada de marcas com partículas. Os resultados mostraram que a técnica proposta neste trabalho foi mais eficiente porque não há necessidade de uma inicialização prévia de parâmetros (como número de partículas e distância máxima de alcance de marcas), a determinação da posição da marca foi mais precisa, e o tempo de processamento foi menor.

Os principais problemas que surgiram neste trabalho se deveram às condições de iluminação no ambiente de navegação do robô móvel, porque estes fizeram com que a localização e correlação de pontos de interesse nas imagens fosse prejudicada, devido ao alto índice de reflectância dos objetos presentes no ambiente.

5.1 Trabalhos Futuros

Como trabalhos futuros pretende-se:

- Adaptar o sistema de SLAM visual proposto para um Veículo Aéreo Não Tripulado (VANT), fazendo alterações na modelagem do sensor e nos parâmetros do descritor SURF, a fim de obter uma configuração ótima para a navegação do VANT em um ambiente externo;

- Fazer um estudo detalhado sobre a influência da posição de um ponto de interesse na imagem sobre a matriz de covariância do ruído de medição (matriz Q no filtro de Kalman estendido);
- Determinar uma regra para exclusão de marcas antigas do vetor de estados, a fim de melhorar a capacidade computacional do filtro de Kalman estendido;
- Realizar uma correlação estatística das marcas dentro do filtro de Kalman estendido, utilizando a distância de Mahalanobis, a fim de evitar que uma marca já observada anteriormente seja adicionada novamente ao vetor de estados.

Referências Bibliográficas

- Aiube, F. A. L., T. K. N. Baidya & E. A. H. Tito (2006), ‘Processos estocásticos dos preços das commodities: uma abordagem através do filtro de partículas’, *Rev. Bras. Econ.* **60**(3). Rio de Janeiro.
- Artieda, Jorge, José M. Sebastian, Pascual Campoy, Juan F. Correa, Iván F. Mondragón, Carol Martínez & Miguel Olivares (2009), ‘Visual 3-D SLAM from UAVs’, *Jornal of Intelingent Robots and Systems* (55), 299–321.
- Ballesta, M., A. Gil, O. M. Mozos & O. Reinoso (2007), ‘Local descriptors for visual SLAM’, *Workshop on Robotics and Mathematics* . Coimbra, Portugal.
- Barfoot, T. D. (2005), ‘Online visual motion estimation using FastSLAM with SIFT features’, *International Coference on Intelligent Robots and Systems* pp. 3076–3082.
- Bauer, J., N. Sunderhauf & P. Protzel (n.d.), ‘Comparing several implementations of two recently published feature detectors’, *International Conference on Intelligent and Autonomous Systems* . Toulouse, França.
- Bay, Herbert, Andreas Ess, Tinne Tuytelaars & Luc Van Gool (2006), ‘Surf: Speeded-up robust features’, *Lecture notes in computer science* **3951**(404).
- Belo, Felipe Augusto Weilemann (2006), ‘Desenvolvimento de algoritmos de exploração e mapeamento visual para robôs móveis de baixo custo’. Dissertação de Mestrado. PUC-RJ.
- Bryson, M. & S. Sukkariéh (2005), ‘Bearing-only SLAM for an airborne vehicle’, *Australasian Conference on Robotics and Automation* . Sidney, Austrália.
- Chen, Z., J. Samarabandu & R. Rodrigo (2007), ‘Recent advances in simultaneous localization and mapbuilding using computer vision’, *Advanced Robotics* **3**(4).
- Civera, Javier, Andrew J. Davison & J. M. M. Montiel (2008), *IEEE Transactions on Robotics* **24**(5), 932–945.
- Davison, A.J. (2003), ‘Real-time simultaneous localisation and mapping with a single camera’, *IEEE International Conference on Computer Vision* **2**, 1403–1410.
- Davison, A.J. & D. W. Murray (2002), ‘Simultaneous localization and map-building using active vision’, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7), 865–880.

- Davison, A.J., I. Reid, N. Molton & O. Stasse (2007), ‘MonoSLAM: real-time single camera SLAM’, *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1052–1067.
- Elinas, P., R. Sim & J.J. Little (2006), ‘ σ -SLAM: Stereo vision SLAM using the rao-blackwellised particle filter and a novel mixture proposal distribution’, *International Conference on Robotics and Automation* . Orlando, FL, EUA.
- Evans, Christopher (2009), Notes on the opensurf library, Relatório Técnico CSTR-09-001, University of Bristol.
*<http://www.chrisevansdev.com>
- Gil, Arturo, Oscar Martinez Mozos, Monica Ballesta & Oscar Reinoso (2009), ‘A comparative evaluation of interest point detectors and local descriptors for visual SLAM’, *Machine Vision and Applications* .
- Hähnel, D., D. Fox, W. Burgard & S. Thrun (2003), A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements, *em* ‘Proceedings of the Conference on Intelligent Robots and Systems (IROS)’.
- Julier, S. J. & J. K. Uhlmann (1997), ‘A new extension of the Kalman filter to nonlinear systems’, *International Symposium on Aerospace/Defense Sensing, Simulation and Controls* **3**, 26.
- Kim, J. H. & S. Sukkarieh (2003), ‘Airborne simultaneous localisation and map building’, *IEEE International Conference on Robotics and Automation* **1**, 406–411. Taipei, Taiwan.
- Lemaire, T., C. Berger, I. Jung & S. Lacroix (2007b), ‘Vision-based SLAM: Stereo and monocular approaches’, *International Journal of Computer Vision* **74**(3), 343–364.
- Lemaire, T. & S. Lacroix (2007a), ‘Monocular-vision based SLAM using line segments’, *International Conference on Robotics and Automation* . Roma, Itália.
- Lowe, D. G. (1999), ‘Object recognition from local scale-invariant features’, *International Conference on Computer Vision* **2**, 1150–1157. Corfu, Grécia.
- Lucas, André Emanuel Simões (2009), ‘Mapeamento e localização baseados em mosaicos visuais’. Dissertação de Mestrado, Universidade Técnica de Lisboa.
- Montemerlo, M., S. Thrun, D. Koller & B. Wegbreit (2002), ‘FastSLAM: A factored solution to the simultaneous localization and mapping problem’, *National Conference on Artificial Intelligence* . Edmonton, Canadá.
- Mouragnon, E., M. Lhuillier, M. Dhome, F. Dekeyser & P. Sayd (2006), ‘Monocular vision based SLAM for mobile robots’, *International Conference on Pattern Recognition* . Hong Kong, China.

- Mozos, O. M., A. Gil, M. Ballesta & O. Reinoso (2007), ‘Interest point detectors for visual SLAM’, *Lecture Notes in Computer Science* **4788**, 170–179.
- Nieto, J., J. Guivant, E. Nebot & S. Thrun (2003), Real time data association for Fast-SLAM, em ‘Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)’. Taipei, Taiwan.
- Salvi, J., Y. Petillo, S. Thomas & J. Aulinas (2008), ‘Visual SLAM for underwater vehicles using video velocity log and natural landmarks’, *OCEANS* pp. 1–6. 15 a 18 de Setembro. Quebec, Canadá.
- Santana, André Macêdo (2007), Localização e Planejamento de Caminhos para um Robô Humanóide e um Robô Escravo com Rodas, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte.
- Sim, R., P. Elinas, M. Griffin & J. J. Little (2005), ‘Vision-based SLAM using the rao-blackwellised particle filter’, *IJCAI Workshop on Reasoning with Uncertainty in Robotics* pp. 9–16. Edinburgo, Escócia.
- Sünderhalf, Niko, Sven Lange & Peter Protzel (2007), ‘Using the unscented kalman filter in Mono-SLAM with inverse depth parametrization for autonomous airship control’, *IEEE International Workshop on Safety, Security and Rescue Robotics*. Roma, Itália.
- Thrun, Sebastian, Wolfram Burgard & Dieter Fox (2006), *Probabilistic robotics*, MIT Press.
- Trucco, Emanuele & Alessandro Verri (1998), *Introductory techniques for 3-D computer vision*, Prentice Hall.
- Wu, Jing & Hong Zhang (2007), ‘Camera Sensor Model for Visual SLAM’, *Fourth Canadian Conference on Computer and Robot Vision*. Ontario, Canadá.
- Zhang, Z., Y. Huang, C. Li & Y. Kang (2008), ‘Monocular vision simultaneous localization and mapping using surf’, *WCICA* pp. 1651–1656. 25 a 27 de Junho. Chongqing, China.

Apêndice A

Filtro de Kalman

O filtro de Kalman (FK) foi criado por P. Swerling (1958) e por R. Kalman (1960) como uma solução recursiva para filtragem linear de dados discretos. Utilizando um estimador preditivo de estados, implementado a partir de equações matemáticas, o filtro de Kalman faz cálculos eficientes de previsão, porque o erro quadrático é minimizado [Aiube et al. 2006].

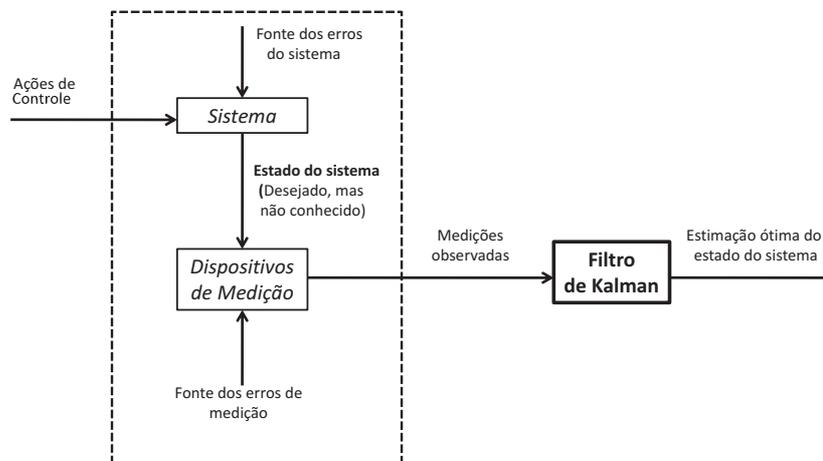


Figura A.1: O problema da filtragem

A Figura A.1 mostra o contexto em que o filtro de Kalman é utilizado. Um sistema físico (por exemplo, um robô móvel) é guiado por um conjunto de entradas externas ou controles e suas saídas são calculadas através de medições obtidas por dispositivos ou sensores, sendo que o conhecimento sobre o comportamento do sistema é obtido somente pelas entradas e pelas saídas observadas. Baseado nas informações disponíveis (modelo, entradas de controle e observações), é necessário obter uma estimativa do estado do sistema que otimiza um dado critério.

A.1 Filtro de Kalman Discreto - FDK

O modelo de espaço de estados do filtro de Kalman discreto busca definir a relação entrada-saída de um sistema linear indiretamente, através de um conjunto de variáveis internas x_t denominadas *estados*, em um determinado instante de tempo t . Os estados são influenciados pelos valores anteriores x_{t-1} e pelas entradas do sistema u_t . Os estados e as entradas influenciam as saídas z_t do sistema. Tal modelo é definido por duas equações:

- Equação de processo:

$$x_t = A_t \cdot x_{t-1} + B_t \cdot u_t + \xi_t \quad (\text{A.1})$$

- Equação de observação (ou de medida):

$$z_t = C_t \cdot x_t + \eta_t \quad (\text{A.2})$$

onde $x_t \in R^k$ é o vetor de estados; $A_{k \times k}$ é a matriz de transição de estados; $B_{k \times m}$ é a matriz de coeficientes de entrada; $u_t \in R^m$ é o vetor das entradas de controle; $\xi \in R^k$ é o vetor de ruídos do processo, que introduz a incerteza nas transições de estados; $z_t \in R^n$ é o vetor de medições; $C_{n \times k}$ é a matriz de observação e $\eta \in R^n$ é o vetor de erros de medição.

O FDK pode ser sub-dividido em duas etapas: a etapa de *predição* e a etapa de *atualização*. Essas etapas levam em consideração as propriedades estatísticas do ruído. Um modelo interno do sistema é usado para atualização e um esquema de realimentação realiza as medições.

A crença $bel(x_t)$ no tempo t é representada através da média μ_t e da covariância Σ_t do vetor de estados x_t . A entrada do filtro de Kalman discreto é a crença no tempo $t - 1$, representada por μ_{t-1} e Σ_{t-1} . Para atualizar estes parâmetros, o filtro precisa de u_t e de z_t . A saída é a crença no tempo t , representada por μ_t e Σ_t . A etapa de predição é descrita por:

$$\begin{cases} \bar{\mu}_t = A_t \cdot \mu_{t-1} + B_t \cdot u_t \\ \bar{\Sigma}_t = A_t \cdot \Sigma_{t-1} \cdot A_t^T + R_t \end{cases} \quad (\text{A.3})$$

A crença predita $\bar{\mu}$ e $\bar{\Sigma}$ é calculada representando a crença $\overline{bel}(x_t)$ um passo a frente, mas antes de incorporar a medida z_t . Esta crença é obtida incorporando o controle u_t . A média é atualizada utilizando uma versão determinística da equação de processo, com a média μ_t substituída pelo estado x_{t-1} . A atualização da covariância considera o fato de que estados dependem de estados anteriores através da matriz linear A . A matriz $R_{k \times k}$ é a covariância do ruído de processo (ξ). A etapa de atualização é descrita por:

$$\begin{cases} K_t = \bar{\Sigma}_t \cdot C_t^T \cdot (C_t \cdot \bar{\Sigma}_t \cdot C_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t \cdot (z_t - C_t \cdot \bar{\mu}_t) \\ \Sigma_t = (I - K_t \cdot C_t) \cdot \bar{\Sigma}_t \end{cases} \quad (\text{A.4})$$

Em seguida, a crença $\overline{bel}(x_t)$ é transformada na crença desejada $bel(x_t)$ incorporando a medida z_t no conjunto de equações A.4. A variável K_t , chamada de *ganho de Kalman*, especifica com que grau a medida deve ser incorporada ao novo estado. A média μ_t é manipulada ajustando-a proporcionalmente a K_t e à diferença entre a medida atual z_t e a medida esperada $C_t \cdot \bar{\mu}_t$. A nova covariância da crença posterior é calculada, ajustando-se para o novo ganho de informação resultantes da medida. A matriz $Q_{m \times m}$ é a covariância do ruído de medição (η).

Este modelo permite a elaboração de um algoritmo computacional capaz de estimar valores ótimos do vetor de estados para os quais a saída será ótima. Assim, é possível prever estados futuros a partir dos estados atuais, permitindo a implementação de sistemas com atualizações em tempo real.

Algoritmo 2 FILTRO DE KALMAN

Entrada: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$

- 1: $\bar{\mu}_t = A_t \cdot \mu_{t-1} + B_t \cdot u_t$;
- 2: $\bar{\Sigma}_t = A_t \cdot \Sigma_{t-1} \cdot A_t^T + R_t$;
- 3: $K_t = \bar{\Sigma}_t \cdot C_t^T \cdot (C_t \cdot \bar{\Sigma}_t \cdot C_t^T + Q_t)^{-1}$;
- 4: $\mu_t = \bar{\mu}_t + K_t \cdot (z_t - C_t \cdot \bar{\mu}_t)$;
- 5: $\Sigma_t = (I - K_t \cdot C_t) \cdot \bar{\Sigma}_t$;

Retorno: μ_t, Σ_t

A.2 Filtro de Kalman Estendido - FKE

Para os casos em que o sistema dinâmico é não linear, o problema da filtragem é solucionado com a utilização do *Filtro de Kalman Estendido*, onde a função não-linear g é aproximada através de uma função linear que é tangente a g na média de sua gaussiana. Para essa aproximação ser válida, esta linearização deve ser uma boa aproximação do modelo não-linear em todo o domínio das incertezas associados à estimação do estado. O modelo do sistema para o FKE é dado a seguir:

$$\begin{cases} x_t = g(u_{t-1}, x_{t-1}) + \xi_t \\ z_t = h(x_t) + \eta_t \end{cases} \quad (\text{A.5})$$

onde:

- $g(u_{t-1}, x_{t-1})$: função não-linear que representa o modelo do sistema;
- $h(x_t)$: função não-linear que representa o modelo das medições.

Como os valores instantâneos de cada ruído ξ_t e η_t não são conhecidos, utiliza-se um resultado aproximado dos vetores de estado e de medida para realizar a linearização,

através de expansões por séries de Taylor. Como resultado obtém-se as relações dadas a seguir:

$$\text{Etapa de Predição} \begin{cases} \bar{\mu}_t = g(u_{t-1}, \mu_{t-1}) \\ \bar{\Sigma}_t = G \cdot \Sigma_{t-1} \cdot G_t^T + R_t \end{cases} \quad (\text{A.6})$$

$$\text{Etapa de Atualização} \begin{cases} K_t = \bar{\Sigma}_t \cdot H_t^T \cdot (H_t \cdot \bar{\Sigma}_t \cdot H_t^T + Q_t)^{-1} \\ \mu_t = \bar{\mu}_t + K_t \cdot (z_t - h(\bar{\mu}_t)) \\ \Sigma_t = (I - K_t \cdot H_t) \cdot \bar{\Sigma}_t \end{cases} \quad (\text{A.7})$$

onde o jacobiano $G_{k \times k}$ é a derivada da função g com respeito a x_{t-1} calculado em u_t e μ_{t-1} ; e $H_{m \times k}$ é o jacobiano de h calculado a partir da média predita $\bar{\mu}_t$, que lineariza o vetor de medições.

O algoritmo do Filtro de Kalman Estendido é similar ao algoritmo do Filtro de Kalman, porém existem diferenças importantes que serão mostradas a seguir:

- **Predição de estados:**

- FK: $A_t \cdot \mu_{t-1} + B_t \cdot u_t$
- FKE: $g(u_{t-1}, \mu_{t-1})$

- **Predição de medidas:**

- FK: $C_t \cdot \bar{\mu}_t$
- FKE: $h(\bar{\mu}_t)$

Algoritmo 3 FILTRO DE KALMAN ESTENDIDO

Entrada: $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$

- 1: $\bar{\mu}_t = g(u_{t-1}, \mu_{t-1})$;
- 2: $\bar{\Sigma}_t = G \cdot \Sigma_{t-1} \cdot G_t^T + R_t$;
- 3: $K_t = \bar{\Sigma}_t \cdot H_t^T \cdot (H_t \cdot \bar{\Sigma}_t \cdot H_t^T + Q_t)^{-1}$;
- 4: $\mu_t = \bar{\mu}_t + K_t \cdot (z_t - h(\bar{\mu}_t))$;
- 5: $\Sigma_t = (I - K_t \cdot H_t) \cdot \bar{\Sigma}_t$;

Retorno: μ_t, Σ_t

As predições lineares do FK são substituídas pelas generalizações não-lineares no FKE. No filtro de Kalman estendido, são utilizados jacobianos G_t e H_t correspondendo às matrizes lineares A_t , B_t e C_t no filtro de Kalman. O jacobiano G_t corresponde às matrizes A_t e B_t , e o jacobiano H_t corresponde à matriz C_t . Para uma melhor descrição do algoritmo e para uma melhor compreensão sobre a derivação matemática do filtro de Kalman, consultar [Thrun et al. 2006].