

Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia Elétrica

**LOCALIZAÇÃO DE UM ROBÔ MÓVEL USANDO
ODOMETRIA E MARCOS NATURAIS**

Clauber Gomes Bezerra

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Rio Grande do Norte, como parte dos requisitos necessários para obtenção do grau de Mestre em Ciências.

Orientador: Prof. Dr. Pablo Javier Alsina
Co-orientador: Prof. Dr. Adelardo A. D. Medeiros

Natal, Março de 2004

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

Aprovada em 08 de Março de 2004 pela comissão examinadora,
formada pelos seguintes membros:

Prof. Dr. Pablo Javier Alsina (Orientador)
Departamento de Engenharia de Computação e Automação - UFRN

Prof. Dr. Adelardo A. D. Medeiros (Co-Orientador)
Departamento de Engenharia de Computação e Automação - UFRN

Prof. Dr. Adrião Duarte Dória Neto (Examinador Interno)
Departamento de Engenharia de Computação e Automação - UFRN

Prof. Dr. Teodiano Freire Bastos Filho (Examinador Externo)
Departamento de Engenharia Elétrica - UFES

Resumo

Diversos métodos de navegação de robôs móveis requerem a medição da posição e orientação do robô no seu espaço de trabalho. No caso de robôs móveis com rodas, técnicas baseadas em odometria permitem determinar a localização do robô através da integração de medições dos deslocamentos incrementais de suas rodas. No entanto, essa técnica está sujeita a erros que se acumulam com a distância percorrida pelo robô, o que inviabiliza o seu uso exclusivo. Outros métodos se baseiam na detecção de marcos naturais ou artificiais, cuja localização é conhecida, presentes no ambiente. Apesar desta técnica não gerar erros cumulativos, ela pode requisitar um tempo de processamento bem maior do que o uso de odometria. Assim, muitos métodos fazem uso de ambas as técnicas, de modo a corrigir periodicamente os erros de odometria, através de medições obtidas a partir dos marcos. De acordo com esta abordagem, propomos neste trabalho um sistema híbrido de localização para robôs móveis com rodas em ambientes internos, baseado em odometria e marcos naturais, onde os marcos adotados são linhas retas definidas pelas junções existentes no piso do ambiente, formando uma grade bi-dimensional no chão. Para a detecção deste tipo de marco, a partir de imagens digitais, é utilizada a transformada de Hough, associada a heurísticas que permitem a sua aplicação em tempo real. Em particular, para reduzir o tempo de busca dos marcos, propomos mapear erros de odometria em uma região da imagem capturada que possua grande probabilidade de conter o marco procurado.

Abstract

Several methods of mobile robot navigation request the mensuration of robot position and orientation in its workspace. In the wheeled mobile robot case, techniques based on odometry allow to determine the robot localization by the integration of incremental displacements of its wheels. However, this technique is subject to errors that accumulate with the distance traveled by the robot, making unfeasible its exclusive use. Other methods are based on the detection of natural or artificial landmarks present in the environment and whose location is known. This technique doesn't generate cumulative errors, but it can request a larger processing time than the methods based on odometry. Thus, many methods make use of both techniques, in such a way that the odometry errors are periodically corrected through mensurations obtained from landmarks. According to this approach, this work proposes a hybrid localization system for wheeled mobile robots in indoor environments based on odometry and natural landmarks. The landmarks are straight lines defined by the junctions in environment's floor, forming a bi-dimensional grid. The landmark detection from digital images is performed through the Hough transform. Heuristics are associated with that transform to allow its application in real time. To reduce the search time of landmarks, we propose to map odometry errors in an area of the captured image that possesses high probability of containing the sought mark.

Sumário

1	Introdução	1
1.1	Métodos de Localização de Robôs Móveis	5
1.2	Motivação	7
1.3	Objetivos	10
1.4	Organização da Dissertação	11
2	Localização de Robôs Móveis	12
2.1	Odometria	14
2.1.1	Erros de Odometria	18
2.1.2	Correção dos Erros de Odometria	21
2.2	Localização Baseada em Marcos	23
2.2.1	Marcos Artificiais	25
2.2.2	Marcos Naturais	26
2.3	Discussão	27
3	Processamento Digital de Imagens	29
3.1	Modelo da Câmera	30
3.2	Representação de Imagens	32
3.3	Segmentação	33
3.3.1	Histograma	34
3.3.2	Limiarização	35
3.4	A Transformada de Hough	37

3.4.1	Transformada de Hough para Retas	39
3.4.2	Representação Normal de uma Reta	40
3.4.3	Processamento da Transformada de Hough	42
3.4.4	Variações da Transformada de Hough	45
3.5	Discussão	46
4	Sistema Proposto	48
4.1	Sistema de Odometria	51
4.2	Estimativa dos Erros de Odometria	52
4.2.1	Região de Incerteza	58
4.3	Sistema de Visão	61
4.3.1	Modelagem da Câmera	62
4.4	Mapeamento do Erro de Odometria na Imagem	68
4.5	Detecção dos Marcos	70
4.6	Localização Absoluta do Robô	72
4.7	Discussão	74
5	Resultados Experimentais	76
5.1	Plataforma Robótica Utilizada	76
5.2	Localização Padrão do Robô	80
5.3	Sistema de Odometria	81
5.4	Cálculo da Estimativa do Erro	87
5.5	Processamento das Imagens	92
5.5.1	Segmentação das Imagens	92
5.5.2	Transformada de Hough	93
5.6	Resultados Finais	97
6	Conclusões Gerais	101
	Referências Bibliográficas	105

Lista de Figuras

1.1	Arquitetura de um sistema de navegação.	4
1.2	Estrutura do sistema proposto.	9
2.1	Modelo da plataforma robótica utilizada.	13
2.2	Deslocamento de um robô utilizando odometria.	14
2.3	Movimento infinitesimal do robô.	16
2.4	Regiões de incerteza durante o movimento do robô.	21
2.5	Correção do erro de odometria.	22
2.6	Localização usando marcos.	25
3.1	Diagrama de funcionamento do sistema de visão utilizado.	30
3.2	Formação da imagem em uma câmera perspectiva.	31
3.3	Modelo de projeção em perspectiva da câmera.	32
3.4	(a) Uma imagem e (b) seu histograma.	35
3.5	(a) Uma imagem, (b) seu histograma e (c) a imagem limiarizada. . .	37
3.6	(a) Conjunto de pontos em uma imagem digital e (b) as retas detectadas pela transformada de Hough.	38
3.7	Exemplo de detecção de uma reta. (a) reta no plano cartesiano e (b) Interseção das retas no plano de parâmetros.	40
3.8	Representação normal de uma reta.	41
3.9	Transformada de Hough utilizando a representação normal: (a) Imagem com uma reta. (b) Transformada de Hough obtida para três pontos.	42
3.10	(a) Discretização de uma imagem e (b) do seu espaço de parâmetros.	43

3.11	(a) Uma imagem contendo duas retas e (b) as retas detectadas.	45
4.1	Diagrama de funcionamento do sistema de localização.	50
4.2	Região de incerteza ao redor da localização estimada pela odometria.	60
4.3	Estrutura do sistema de visão do robô.	62
4.4	Referenciais do ambiente, do robô e da câmera.	63
4.5	Transformação geométrica entre o referencial do ambiente e do robô.	63
4.6	Transformações geométricas entre os referencial do robô e da câmera.	64
4.7	Mapeamento do erro de odometria na imagem.	69
4.8	Detecção dos marcos através da transformada de Hough.	71
4.9	Grade do Piso	73
5.1	Ponto de localização do robô utilizado.	78
5.2	Plataforma robótica utilizada nos experimentos.	79
5.3	Esquema de comunicação do LNP.	80
5.4	(a) Estrutura do sistema de visão global e (b) rótulo utilizado por esse sistema.	82
5.5	Comparação entre a localização calculada pela odometria (linha pontilhada) e a localização padrão do robô (linha cheia): (a) caminho retilíneo percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.	83
5.6	Comparação entre a localização calculada pela odometria (linha pontilhada) e a localização padrão do robô (linha cheia): (a) caminho em forma de “U” percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.	86
5.7	Método UMBmark: (a) sentido horário e (b) sentido anti-horário.	88
5.8	Resultados obtidos na estimação do erro de odometria. (a) caminho em forma de “U” percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.	91

5.9	Comparação entre a segmentação de: (a) e (b) uma imagem completa e (c) e (d) apenas uma região.	94
5.10	(a) Uma imagem contendo duas retas e (b) o resultado incorreto da detecção das retas.	95
5.11	Intervalos de cálculo das retas no espaço de parâmetros.	96
5.12	Comparação entre a localização calculada pela odometria sem correção (linha pontilhada), com correção absoluta (linha tracejada) e a localização padrão do robô (linha cheia): (a) caminho retilíneo percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.	100

Lista de Tabelas

5.1	Parâmetros do modelo de odometria do robô.	79
5.2	Erros de odometria obtidos para um caminho retilíneo.	84
5.3	Erros de odometria obtidos para um caminho em forma de “U”.	87
5.4	Resultados do UMBmark: (a) sentido horário e (b) sentido anti-horário.	89
5.5	Parâmetros utilizados no cálculo da estimativa do erro.	90
5.6	Resultados obtidos com o mapeamento do erro de odometria na imagem.	97
5.7	Parâmetros utilizados pelo sistema de visão.	98
5.8	Resultados obtidos utilizando o método absoluto.	99

Capítulo 1

Introdução

As aplicações envolvendo robôs móveis autônomos vêm crescendo significativamente nos últimos anos. Um dos fatores que impulsiona esse crescimento é a grande diversidade e relevância de tarefas que essa classe de robôs pode realizar, já que, ao contrário dos robôs manipuladores (Craig, 1986), eles possuem a capacidade de se locomover livremente pelo espaço de trabalho, limitados apenas por eventuais obstáculos. Além disso, constantes avanços tecnológicos vêm sendo realizados nos últimos anos, o que induz a uma redução dos preços dos robôs e uma significativa melhoria em seu desempenho (Franchin, 2003b).

Hoje em dia, existe uma grande variedade de plataformas robóticas móveis, como é o caso dos humanóides, veículos terrestres, aeroespaciais e aquáticos. Dentre os veículos terrestres, podemos citar os robôs móveis com rodas, alvo de estudo neste trabalho, que se destacam pela simplicidade de construção e eficiência. Os robôs, estão cada vez mais sendo empregados para auxiliar ou substituir o homem em diversas tarefas. Dentre as aplicações dos robôs móveis, podemos citar o transporte de carga, limpeza e manutenção de ambientes, segurança, diversão, o patrulhamento e a exploração de ambientes perigosos ou nocivos ao ser humano (Silva, 2001).

A navegação é um dos principais problemas em robótica móvel, sendo um tema bastante abordado em trabalhos científicos nesta área (Vassallo, 1998; Pedrosa, 2001; Menezes, 1999; Grisetti et al., 2000; Pacheco e Costa, 2002; Sarcinelli-filho et al., 2002; Lora et al., 1998). A navegação autônoma de um robô móvel, pode ser descrita como o processo utilizado pelo mesmo para mover-se em seu ambiente de trabalho, geralmente povoado de obstáculos, de uma posição e orientação iniciais para uma posição e orientação finais. Neste processo, o robô deve percorrer um caminho realizável pelo mesmo, obedecendo as suas restrições cinemáticas e dinâmicas e, quando necessário, desviando de obstáculos. Durante esse percurso, o robô deve ser capaz de identificar constantemente sua posição e orientação dentro do ambiente.

Para solucionar um problema como este, o robô é projetado com base em uma arquitetura, que define o software e o hardware utilizados para executar a sua tarefa. Atualmente existem três paradigmas para uma arquitetura robótica: deliberativo, reativo e híbrido (deliberativo/reativo) (Murphy, 2000; Ribeiro et al., 2001). Em determinadas aplicações o conhecimento desses paradigmas e a utilização do mais adequado, faz com que problemas sejam resolvidos com uma maior facilidade (Bandeira, 2002).

Dentre esses três paradigmas, o deliberativo é o mais antigo. Ele é baseado em uma seqüência de primitivas: *SENTIR* → *PLANEJAR* → *AGIR*. Segundo este paradigma, o robô utiliza os seus dados sensoriais para planejar e depois agir baseado nesse planejamento. Porém, esse paradigma possui uma limitação, já que se durante o planejamento ocorrerem mudanças significativas no ambiente, a ação executada não será a mais adequada para a situação. Nesse caso, a velocidade do planejamento pode influenciar na capacidade do robô de reagir a situações urgentes.

Visando contornar essas deficiências, foi proposto o paradigma reativo que não

utiliza planejamento. Nesse paradigma, o robô é ativado de acordo com um conjunto de reações (pares *SENTIR* \leftrightarrow *AGIR*), que especificam a ação a ser realizada para uma determinada situação sob a orientação de diferentes comportamentos, geralmente obedecendo uma determinada prioridade. Dessa forma, para comportamentos distintos, uma mesma informação sensorial pode resultar em diferentes ações. Claramente esse paradigma é bem limitado, já que não há nenhum planejamento.

Já o paradigma híbrido, tenta combinar os pontos fortes dos outros dois paradigmas. A tarefa a ser realizada pelo robô é decomposta em subtarefas. Para cada uma dessas subtarefas é feito um planejamento inicial, atribuindo um comportamento mais adequado para as mesmas. A partir daí, a execução dos comportamentos para cada subtarefa é realizada de acordo com o paradigma reativo. Essa arquitetura é do tipo *PLANEJAR* \rightarrow *SENTIR* \leftrightarrow *AGIR*, onde todo o planejamento é realizado em uma única etapa.

Dentre os modelos de arquiteturas citados, o presente trabalho se aplica melhor ao paradigma deliberativo. Esse paradigma tem sido bastante utilizado como base de arquiteturas de navegação de robôs móveis, dividindo esse problema em níveis progressivos de abstração. Um exemplo desse tipo de arquitetura, para a navegação de um robô móvel, é mostrada na figura 1.1. Nessa arquitetura, o processo de navegação é dividido em cinco níveis hierárquicos: modelagem do ambiente, localização, geração de caminho, geração de trajetória e execução (Alsina et al., 2002). Em cada um desses níveis, as seguintes tarefas são realizadas:

- **Modelagem do ambiente:** neste nível, o robô utiliza sensores para coletar informações e criar um modelo contendo as principais características estruturais, incluindo possíveis obstáculos, do ambiente em que ele deseja se locomover;

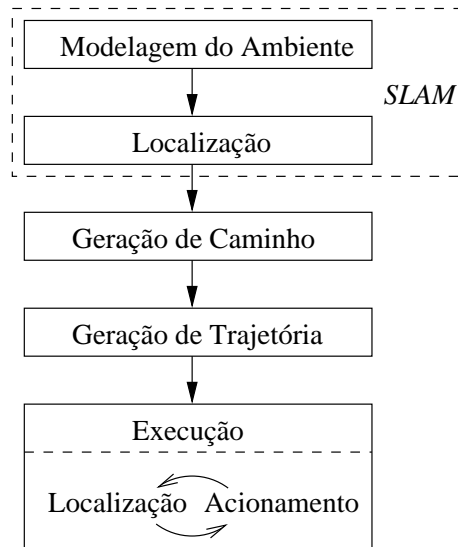


Figura 1.1: Arquitetura de um sistema de navegação.

- **Localização:** é determinada a posição e orientação do robô no espaço, com base no modelo obtido no nível de modelagem do ambiente. Muitas vezes os níveis de modelagem do ambiente e localização são executados simultaneamente, em uma abordagem conhecida como SLAM (*Simultaneous Localization And Mapping* - Localização e Mapeamento Simultâneos);
- **Geração de caminho:** o robô calcula uma curva geométrica que o leve da sua localização atual até a sua localização final, sem colidir com nenhum obstáculo. Para isso, várias técnicas podem ser utilizadas como mapa de rotas, decomposição em células e campo de potencial (Latombe, 1998);
- **Geração de trajetória:** o caminho calculado no nível anterior é adaptado levando-se em conta as restrições temporais impostas ao robô, tornando-se uma trajetória. Com isso, são calculadas velocidades que devem ser submetidas ao robô em cada instante de tempo para que o mesmo possa percorrer o caminho

no intervalo de tempo especificado;

- **Execução de trajetória:** neste último nível, os atuadores do robô são controlados de forma que a trajetória percorrida por ele seja a mais próxima possível da calculada no nível anterior. Desvios em relação à trajetória planejada devem ser detectados e corrigidos em tempo real. Para detectar estes desvios, é necessário monitorar constantemente a localização atual do robô.

Nota-se, através deste exemplo, que a localização é uma etapa de fundamental importância no processo de navegação do robô, interferindo de forma significativa no desempenho do mesmo. Durante toda a execução da trajetória é necessário que seja conhecida constantemente a localização do robô, para que se possa realizar um controle da trajetória percorrida.

1.1 Métodos de Localização de Robôs Móveis

Localizar um robô móvel consiste em determinar a sua posição e orientação no espaço em um determinado instante de tempo. Os métodos de localização de robôs móveis podem ser classificados em duas grandes categorias: métodos de localização relativa e métodos de localização absoluta (Borenstein, Everett e Feng, 1996; Borenstein, Everett, Feng e Wehe, 1996). Os métodos de localização relativa utilizam as localizações obtidas em instantes anteriores para estimar a localização atual do robô, sendo a odometria e a navegação inercial dois métodos baseados nesse princípio (Borenstein, Everett e Feng, 1996; Borenstein, Everett, Feng e Wehe, 1996). Devido ao seu baixo custo, a odometria é um método de localização bastante utilizado em robôs móveis com rodas. Esse método utiliza sensores, *encoders* (codificadores) óticos por exemplo, para medir a rotação das rodas do robô e a partir do modelo cinemático deste

calcular a sua posição e orientação. No entanto, a odometria pode gerar erros que se propagam cumulativamente com a distância percorrida (Valgas-filho, 2002). Com isso, o uso de odometria pode ser tornar proibitivo quando o robô percorre grandes distâncias.

Já os métodos de localização absoluta, utilizam apenas as informações atuais dos seus sensores para determinar a localização do robô em relação a um referencial fixo absoluto. Exemplos desse tipo de localização são o uso de balizas ativas, *map-matching* (casamento de mapas) e marcos artificiais ou naturais (Borenstein, Everett e Feng, 1996; Borenstein, Everett, Feng e Wehe, 1996). No método das balizas ativas, a localização é calculada medindo a direção de incidência, ou a distância para três ou mais balizas ativas que emitem ou refletem sinais de luz ou rádio, a partir de localizações conhecidas no ambiente. Já no método de *map-matching*, o robô utiliza a informação de seus sensores para construir um modelo do ambiente. A localização do robô é então estimada comparando-se o mapa construído correntemente pelo robô com um modelo do ambiente pré-armazenado. Outra forma de se obter a localização do robô consiste em reconhecer marcos, que são objetos ou características distinguíveis presentes no ambiente, dispostos em localizações conhecidas. Os marcos podem ser artificiais, quando introduzidos no ambiente com a finalidade de auxiliar na navegação do robô, ou naturais quando já se encontravam no ambiente antes do surgimento do problema de navegação. Apesar desse método não gerar erros cumulativos, ele pode necessitar de um tempo de processamento bem maior do que o uso de odometria.

1.2 Motivação

Nesta dissertação, será abordado o problema da localização de robôs móveis. Vários são os trabalhos existentes na literatura envolvendo esse tema (Lages, 1998; Se et al., 2001; Lora et al., 1997; Komoriya e Oyama, 1994; Adams et al., 1994; Betke e Gurvits, 1994; Mächler, 1998; Clerentin et al., 2002). Nesse contexto, a odometria é um método muito utilizado para auxiliar a navegação de veículos autônomos. A maioria desses trabalhos, utiliza um método de localização relativa, geralmente odometria, que possui um período de amostragem baixo, em conjunto com um método de localização absoluta, com um período de amostragem maior, para corrigir periodicamente a posição relativa do robô.

Por exemplo, Lora et al. (1997) apresentam um sistema de localização, para auxiliar na navegação de um veículo autônomo com acionamento diferencial. O sistema é baseado em uma fusão sensorial, via filtro de Kalman (Haykin, 2002), da informação de orientação proveniente de dois subsistemas com períodos de amostragem diferentes. O primeiro é um sistema de odometria, que utiliza a leitura de dois *encoders*, localizados nas rodas direita e esquerda do veículo, e o segundo é um bússola digital. As orientações obtidas pela odometria e pela bússola são utilizadas para se obter uma estimativa do erro de orientação cometido usando a odometria. Essa estimativa é usada para corrigir a orientação do veículo e em seguida a sua posição, através de um processo retroativo.

Nos trabalhos que utilizam odometria, poucos são os que se preocupam em analisar como se propaga o erro à medida que o robô se desloca no ambiente (Mächler, 1998; Valgas-filho, 2002; Kelly, 2000; Kelly, 2001; Chong e Kleeman, 97). É importante entender como esses erros se propagam, na medida em que a informação

da estimativa do erro acumulado pode ser utilizada para acelerar o método de localização absoluta utilizado. Recentemente Kelly (2000; 2001) descreveu analiticamente como os erros de odometria se propagam em determinados modelos cinemáticos. Kelly propõe a discretização e linearização do conjunto de equações diferenciais não-lineares do modelo de odometria e representa analiticamente a relação existente entre o distúrbio ocorrido nas variáveis de entrada (velocidades medidas pelos sensores) e o distúrbio nas variáveis de saída (posição e orientação do veículo).

O sistema de localização proposto neste trabalho, ilustrado na figura 1.2, utiliza odometria e detecção de marcos naturais. O método de odometria é utilizado, principalmente, devido a sua simplicidade e baixo custo. No entanto, de acordo com o citado anteriormente, este método está sujeito a erros que o tornam ineficiente quando utilizado isoladamente. Daí surge a necessidade de utilizar esse método em conjunto com um método de localização absoluta, de forma a corrigir periodicamente a posição obtida por odometria. A partir da odometria, obtém-se uma localização aproximada do robô em determinado instante e, periodicamente, a partir de imagens capturadas por uma câmera digital posicionada no robô, são detectados marcos no ambiente para se obter uma localização absoluta do robô, corrigindo o erro de odometria.

O sistema é projetado para ser utilizado em um robô móvel com rodas, em um ambiente interno semi-estruturado, ou seja, onde algumas de suas características são conhecidas previamente. Os marcos são detectados através da transformada de Hough para retas, utilizada devido às características dos marcos escolhidos, que são linhas retas definidas pelas divisões do piso do ambiente no qual o robô deverá navegar. A maior contribuição deste trabalho consiste no desenvolvimento de uma técnica para redução do tempo de processamento da imagem contendo o marco. Esta redução é

obtida com base em um conjunto de heurísticas e na determinação de uma região de interesse na imagem, a qual é obtida através do mapeamento do erro de odometria estimado em um subconjunto da imagem, onde existe grande probabilidade de se encontrar o marco procurado.

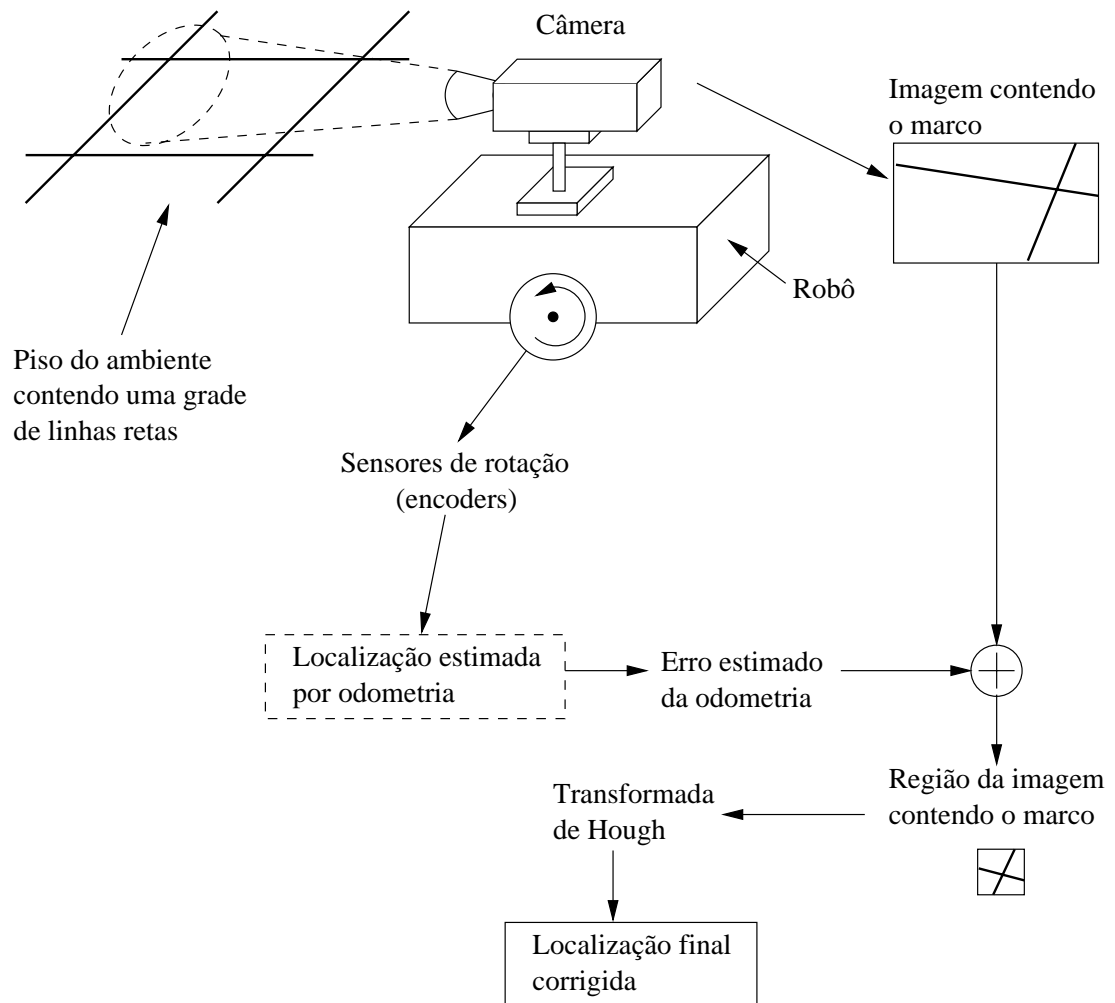


Figura 1.2: Estrutura do sistema proposto.

1.3 Objetivos

O principal objetivo deste trabalho é desenvolver um sistema de localização baseado em odometria e marcos naturais para um robô móvel com rodas. Para se alcançar o objetivo principal traçado, podemos destacar alguns objetivos específicos:

- Utilizar odometria para se obter uma localização aproximada do robô em determinado instante;
- Detectar marcos, a partir de imagens capturadas por uma câmera digital, através da transformada de Hough para retas;
- Análise da propagação dos erros de odometria ocorridos para determinadas trajetórias;
- Determinar uma região de busca dos marcos na imagem, a partir da localização obtida por meio da odometria, utilizando a estimativa do erro propagado até o instante, de forma a reduzir o tempo de processamento da transformada de Hough;
- Definir outras heurísticas, que associadas à transformada de Hough, permitam a sua aplicação em tempo real;
- Implementar o sistema de localização proposto em um robô móvel real, que possui sensores acoplados às suas rodas, para medir o deslocamento das mesmas, e uma câmera digital embarcada.

1.4 Organização da Dissertação

Este texto está organizado em seis capítulos. Neste capítulo 1, foi traçada uma breve introdução a respeito da robótica, enfatizando os robôs móveis, sua navegação e em particular o problema de localização, que é a finalidade deste trabalho. No capítulo 2, são relatados alguns trabalhos relacionados à localização de robôs móveis e dois dos principais métodos utilizados para isso, odometria e reconhecimento de marcos naturais. O capítulo 3 descreve alguns conceitos relativos ao processamento digital de imagens, como também as técnicas utilizadas para a detecção dos marcos do sistema, em especial a transformada de Hough. No capítulo 4, são expostos os métodos utilizados para a implementação do sistema e é feita uma descrição detalhada do mesmo. No capítulo 5, são descritos os resultados experimentais obtidos pelo sistema. Além disso, é feita uma breve descrição da plataforma de teste utilizada na implementação, os robôs Lego, detalhando também o protocolo que permite a comunicação, via infravermelho, entre o robô e um computador. Finalmente, no capítulo 6, são detalhadas algumas conclusões a respeito do trabalho desenvolvido, através de uma análise dos resultados obtidos.

Capítulo 2

Localização de Robôs Móveis

Uma das principais tarefas de um sistema de navegação de um robô móvel é a determinação da localização do robô, ou seja, determinar a sua posição e orientação no ambiente de trabalho. No desenvolvimento deste trabalho, utilizamos um modelo de plataforma robótica com acionamento diferencial, ilustrada na figura 2.1. Esse é um tipo de plataforma bastante comum, caracterizada pelo uso de duas rodas colocadas nas laterais do robô, acionadas individualmente por dois motores independentes. Determinar a localização desse tipo de plataforma, consiste em se obter a posição do seu ponto central (x, y) e o ângulo θ de orientação do robô em relação a um referencial fixo no espaço de trabalho.

Adotaremos a seguinte nomenclatura para descrever as variáveis cinemáticas do robô:

- b é o comprimento do eixo;
- r_e e r_d são os raios das rodas esquerda e direita, respectivamente;
- ω_e e ω_d são as velocidades angulares das rodas esquerda e direita, respectivamente;

- v_e e v_d são as velocidades lineares das rodas esquerda e direita, respectivamente;
- v é a velocidade linear do robô;
- ω é a velocidade angular do robô.

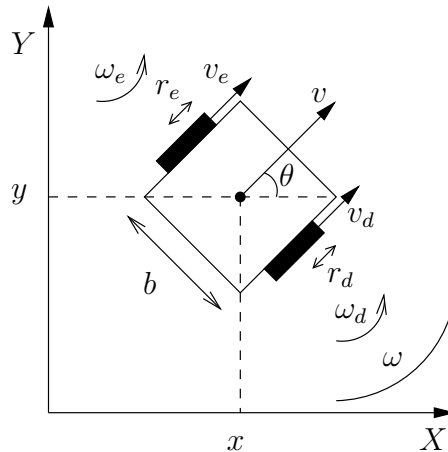


Figura 2.1: Modelo da plataforma robótica utilizada.

Para se localizar, o robô utiliza seus sensores de modo a obter informações a respeito do ambiente ao seu redor. Diferentes tipos de sensores têm sido utilizados para realizar essa tarefa, como por exemplo *encoders*, *lasers*, câmeras digitais, sonares, bússolas, dentre outros. É comum também o uso de dois ou mais sensores simultaneamente, de forma combinada.

Dependendo do tipo de sensor utilizado, existem diversos métodos para determinar a localização de um robô móvel, cada um com suas vantagens e desvantagens. Neste capítulo, apresentamos dois desses métodos, nos quais se baseia o sistema de localização proposto neste trabalho. O primeiro deles é um método de localização relativa, que utiliza medidas de odometria para estimar a localização do veículo. Já o segundo é um método de localização absoluta, baseado na detecção de marcos visuais presentes no ambiente.

2.1 Odometria

O método de odometria consiste na determinação da posição e orientação do robô através da integração dos deslocamentos incrementais de suas rodas, medidos a partir de um referencial fixo. Para exemplificar esse método, considere um robô se deslocando no seu ambiente seguindo uma determinada trajetória, como mostrado na figura 2.2. No instante t_0 o robô se encontra na posição inicial P_0 e inicia o deslocamento. No instante t_1 o robô se encontra na posição P_1 , que pode ser calculada como a soma entre a posição anterior do robô P_0 e o deslocamento realizado pelo mesmo (D_1) entre os instantes t_0 e t_1 . No instante seguinte t_2 , a posição atual do robô (P_2) é novamente calculada com base na sua posição anterior e no deslocamento efetuado desde o instante anterior (D_2). Dessa forma, a posição atual do robô pode ser calculada com base no acúmulo dos deslocamentos efetuados pelo robô, com relação a sua posição inicial.

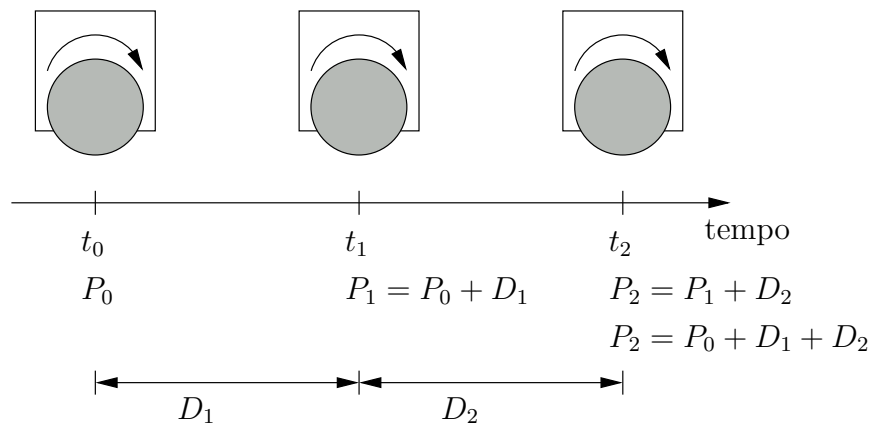


Figura 2.2: Deslocamento de um robô utilizando odometria.

Para medir o deslocamento do robô é necessário a utilização de um sensor que possa medir a rotação das suas rodas. Em aplicações envolvendo robôs móveis, um

tipo de sensor de odometria bastante utilizado é o codificador ou *encoder* ótico. Esse tipo de sensor funciona com base na transmissão e recepção de luz através de um disco perfurado, que gira acoplado ao eixo que movimentava a roda do robô. Os *encoders* óticos podem ser divididos em dois grupos: *encoders* relativos ou incrementais e *encoders* absolutos. No caso de *encoders* incrementais, utilizados neste trabalho, pode-se determinar a velocidade angular da roda com base na frequência do trem de pulsos gerados no receptor de luz do *encoder* (Franchin, 2003a).

Para a plataforma mostrada na figura 2.1, as medidas de odometria são feitas a partir de dois *encoders* incrementais acoplados a cada um dos motores, o quais permitem medir as rotações da roda correspondente. A partir das informações medidas dos *encoders* e conhecendo alguns parâmetros cinemáticos do robô, é possível deduzir as equações que irão determinar a sua posição e orientação em um dado instante, incrementalmente a partir de uma localização inicial conhecida.

Aproximando o movimento executado pelo robô por um arco de circunferência, como mostrado na figura 2.3, onde r é o raio de giro do robô, teremos que a sua velocidade linear será:

$$v = \omega r \quad (2.1)$$

Da mesma forma, as velocidades lineares de cada roda serão dadas por:

$$v_d = \omega \left(r + \frac{b}{2} \right) \quad (2.2)$$

$$v_e = \omega \left(r - \frac{b}{2} \right) \quad (2.3)$$

Somando as equações 2.2 e 2.3 e substituindo o resultado na equação 2.1, teremos:

$$v = \frac{v_d + v_e}{2} \quad (2.4)$$

Já a velocidade angular do robô, pode ser obtida subtraindo-se a equação 2.3 da equação 2.2:

$$w = \frac{v_d - v_e}{b} \quad (2.5)$$

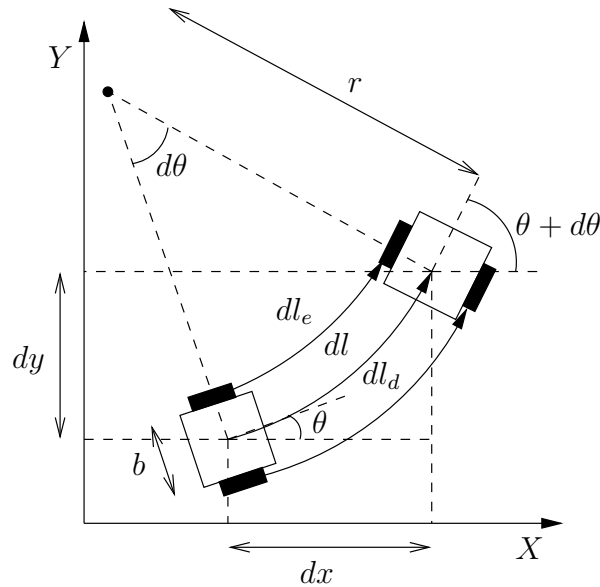


Figura 2.3: Movimento infinitesimal do robô.

Considerando que o movimento mostrado na figura 2.3 é executado em um intervalo de tempo infinitesimal dt , podemos decompor aproximadamente o deslocamento linear do robô, dl nas suas componentes horizontal (dx) e vertical (dy). Com isso, teremos que essas componentes e o deslocamento angular $d\theta$ do robô, serão:

$$dx = dl \cos \theta = v dt \cos \theta$$

$$dy = dl \sin \theta = v dt \sin \theta$$

$$d\theta = \omega dt$$

Discretizando essas três equações, através do método de integração de Euler, teremos:

$$x(t + \Delta t) = x(t) + v\Delta t \cos \theta(t) \quad (2.6)$$

$$y(t + \Delta t) = y(t) + v\Delta t \sen \theta(t) \quad (2.7)$$

$$\theta(t + \Delta t) = \theta(t) + \omega\Delta t \quad (2.8)$$

Discretizando também as velocidades lineares de cada roda do robô em função dos seus respectivos deslocamentos dl_d e dl_e , obtemos:

$$v_d = \frac{dl_d}{dt} = \frac{\Delta l_d}{\Delta t} \quad (2.9)$$

$$v_e = \frac{dl_e}{dt} = \frac{\Delta l_e}{\Delta t} \quad (2.10)$$

Considerando N_d e N_e o número de pulsos lidos nos *encoders* direito e esquerdo e N_{res} a resolução dos *encoders*, que indica o número de pulsos em um giro da roda, podemos obter o deslocamento de cada roda do robô como sendo:

$$\Delta l_d = \frac{N_d}{N_{res}} 2\pi r_d \quad (2.11)$$

$$\Delta l_e = \frac{N_e}{N_{res}} 2\pi r_e \quad (2.12)$$

Substituindo as equações 2.12 e 2.11 nas equações 2.9 e 2.10 respectivamente, obtemos:

$$v_d = \frac{N_d}{N_{res}} \frac{2\pi r_d}{\Delta t} \quad (2.13)$$

$$v_e = \frac{N_e}{N_{res}} \frac{2\pi r_e}{\Delta t} \quad (2.14)$$

Substituindo agora, as equações 2.13 e 2.14 nas equações 2.6, 2.7 e 2.8, utilizando as equações 2.4 e 2.5, podemos obter a localização do robô em relação a um referencial inicial, acumulando-se as distâncias percorridas pelo mesmo e as variações de

orientação a cada instante de tempo, de acordo com as equações:

$$x(t + \Delta t) = x(t) + (N_d r_d + N_e r_e) \frac{\pi}{N_{res}} \cos \theta(t) \quad (2.15)$$

$$y(t + \Delta t) = y(t) + (N_d r_d + N_e r_e) \frac{\pi}{N_{res}} \sen \theta(t) \quad (2.16)$$

$$\theta(t + \Delta t) = \theta(t) + \frac{2\pi}{b N_{res}} (N_d r_d - N_e r_e) \quad (2.17)$$

2.1.1 Erros de Odometria

A odometria é um método simples e bastante utilizado para determinar a localização de um robô móvel. No entanto, essa técnica está sujeita a erros que fazem com que ela forneça apenas uma estimativa da localização exata do robô em um dado instante. Por exemplo, imagine que o robô está percorrendo um ambiente com um piso muito escorregadio. No pior caso, quando as rodas se moverem irão escorregar de tal forma que não haverá deslocamento do robô. Nesse caso, apesar do robô não ter realizado nenhum deslocamento no ambiente houve um deslocamento das suas rodas, o que implicaria em uma medida errada de odometria. Como a odometria determina a localização do robô com base no acúmulo das informações obtidas nos instantes anteriores, um erro em determinado instante também compromete as medições dos instantes seguintes. Esse problema poderia ser contornado caso a média dos erros fosse zero, assim a integração ajudaria a reduzir o erro. No entanto, esses erros são polarizados, ou seja, possuem média diferente de zero.

Os erros de odometria podem ser classificados como erros sistemáticos e erros não-sistemáticos, sendo os principais deles causados por (Valgas-filho, 2002; Borenstein, Everett e Feng, 1996):

- Erros sistemáticos:

- Diferença entre os diâmetros das rodas esquerda e direita;
 - Média do diâmetro das rodas diferente do diâmetro nominal;
 - Comprimento do eixo diferente do comprimento nominal;
 - Desalinhamento das rodas;
 - Resolução finita do encoder;
 - Taxa de amostragem do encoder finita;
- Erros não-sistemáticos:
 - Terrenos irregulares;
 - Objetos inesperados no chão;
 - Escorregamento das rodas devido a:
 - * Terrenos escorregadios;
 - * Aceleração acentuada;
 - * Forças externas;
 - * Forças internas;
 - * Nenhum contato entre a roda e o chão.

Os erros sistemáticos são aqueles causados por incertezas nos parâmetros do modelo cinemático do robô, que são utilizados para calcular a sua posição e orientação. Esse tipo de erro pode ter conseqüências desastrosas, já que ele ocorre durante toda a navegação do robô e se acumula constantemente, gerando grandes distorções na determinação da localização atual. Em plataformas robóticas com acionamento diferencial, as principais causas de erros sistemáticos são erros nas medições dos diâmetros das

rodas de tração e do comprimento do eixo do robô. Borenstein e Feng (1996; 1994) propuseram um método de calibração de odometria, em robôs móveis com acionamento diferencial, visando diminuir os erros sistemáticos. Esse método se concentra principalmente em reduzir os erros causados pela incerteza desses parâmetros (comprimento do eixo e diâmetros das rodas). Para isso, o robô é submetido a trajetórias quadrangulares em ambos os sentidos, horário e anti-horário, de forma que os erros de localização medidos são utilizados para ajustar os seus parâmetros cinemáticos.

Já os erros não-sistemáticos são imprevisíveis, causados por situações que surgem inesperadamente, como o escorregamento das rodas, imperfeições no solo e fenômenos similares. Ao contrário dos erros sistemáticos, esse tipo de erro não ocorre constantemente durante a navegação. Borenstein (1995) desenvolveu uma complexa plataforma robótica, com o objetivo de reduzir também os erros não-sistemáticos. A plataforma é composta por duas bases com acionamento diferencial independentes e interligadas através de um eixo prismático. Cada uma das bases possui dois *encoders* acoplados as suas rodas, que possibilitam a obtenção da localização de cada base através de odometria. Além disso, a plataforma possui três outros *encoders* chamados de *encoders* internos: um *encoder* linear ligado ao eixo, que permite a medição da distância entre as duas bases e dois *encoders* rotacionais absolutos em cada base, que medem a orientação das mesmas com relação ao eixo. É proposto um método para corrigir os erros de odometria com base na medição desses *encoders* internos.

Obviamente por estarem constantemente presentes, os erros sistemáticos são os que influenciam mais na estimativa final da posição e orientação do robô, já que, pela própria natureza do método, os erros vão sendo acumulados ao integrar-se os deslocamentos para a obtenção da posição e orientação. É possível, como descrito,

através de procedimentos de calibração adequados ou da construção de plataformas robóticas mais robustas, reduzir os erros de odometria.

2.1.2 Correção dos Erros de Odometria

Devido aos erros de odometria, a localização do robô calculada a partir dos *encoders* possui um possível erro associado. Esse erro pode ser representado por uma região elíptica ao redor da posição atual do robô (Tonouchi et al., 1994), como mostra a figura 2.4. Mächler (1998) apresentou um estudo da propagação dos erros de odometria em que esse erro era melhor modelado em uma região em forma de “banana”. Essa região indica o grau de incerteza da odometria, aumentando juntamente com a distância percorrida pelo robô.

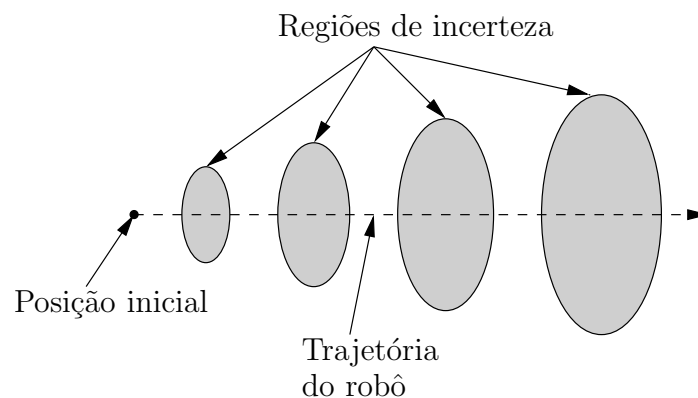


Figura 2.4: Regiões de incerteza durante o movimento do robô.

Para corrigir esse erro, é possível a utilização de outros tipos de sensores, em conjunto com os odômetros, proporcionando um método de localização absoluta. Por exemplo, Mächler (1998) fez uma correção de odometria com base em uma grade de linhas pretas desenhadas em um piso branco. As linhas eram detectadas por um sensor de luz, localizado na parte inferior do robô, direcionado para o chão. Quando uma linha era detectada pelo sensor de luz, o robô utilizava a estimativa da sua posição,

calculada por odometria, para determinar se era uma linha vertical ou horizontal e melhorar essa estimativa.

Lages (1998) também utilizou um sistema de odometria para a estimação da posição e orientação de um robô móvel. Foram implementados dois métodos para corrigir a localização estimada, ambos através de uma fusão sensorial utilizando filtro de Kalman. O primeiro deles utiliza uma bússola digital e o segundo uma câmera, implementando um sistema de visão computacional baseado em marcos visuais, onde os marcos são compostos por pequenos círculos dispostos no ambiente.

É necessário portanto, a utilização de um método de localização absoluta, em conjunto com a odometria, de forma a zerar periodicamente os erros de odometria acumulados. Como pode ser visto na figura 2.5, ao percorrer uma trajetória, o robô obtêm uma informação de odometria que vai se tornando cada vez mais incorreta no decorrer do tempo. Nos instantes t_1 e t_2 o erro de odometria é corrigido por meio de um método de localização absoluta, como a detecção de marcos visuais no ambiente que será discutida na próxima seção.

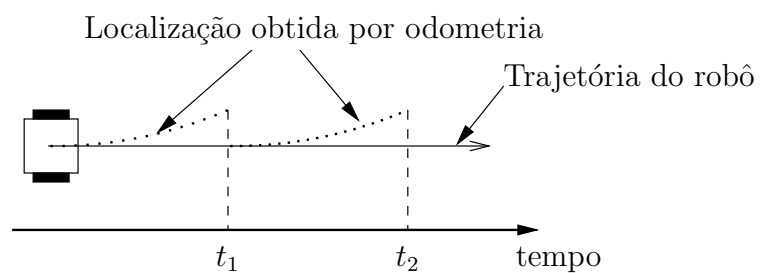


Figura 2.5: Correção do erro de odometria.

2.2 Localização Baseada em Marcos

Na robótica móvel, muitos dos trabalhos que envolvem visão computacional se concentram na idéia de reconstrução geométrica do ambiente, fazendo com que o problema de navegação seja resolvido através de um modelo matemático do mesmo em três dimensões (Buffa et al., 1993; Faugeras et al., 1995; Little et al., 1998). Nesses sistemas, são utilizadas mais de uma câmera digital como base de um modelo de visão estéreo para se obter informações tridimensionais. A partir desta representação geométrica, define-se a localização absoluta do robô com relação a um referencial fixo no ambiente. No entanto, este método possui a desvantagem de que a reconstrução 3D do ambiente necessita de muito tempo de processamento e esforço computacional. O tempo gasto com o processamento das imagens para a extração das características através de segmentação, detecção de contornos, correspondência entre pontos de imagens estéreo e definição de mapas de disparidade e profundidade é muito grande devido à complexidade desses cálculos, que juntos tornam-se computacionalmente dispendiosos.

Porém, em muitos casos, não se torna necessária a reconstrução 3D para a solução do problema. Basta extrair das imagens aquilo que realmente é necessário e relevante para se determinar a localização do robô. Dessa forma, a maioria dos sistemas implementados utiliza uma única câmera a bordo do robô, onde em alguns casos é utilizado um sistema de visão omnidirecional, com a câmera posicionada voltada para cima, apontando para um espelho que proporciona uma visão panorâmica do ambiente (Vlassis et al., 2001). Esses sistemas monoculares procuram encontrar e explorar características específicas do ambiente. O tipo de características e informações exploradas nas imagens variam conforme o ambiente de trabalho (estruturado ou não, externo ou interno, etc.) e a tarefa a ser executada pelo robô (Jarvis, 1983). Essas

características são chamadas de marcos e podem ser naturais ou artificiais.

Marcos são características distinguíveis que um robô pode reconhecer em seu ambiente, a partir dos dados coletados por seus sensores (Borenstein, Everett e Feng, 1996). Geralmente os marcos estão em posições fixas e conhecidas em relação ao referencial do ambiente. As informações dos sensores são utilizadas para reconhecer os marcos e determinar as suas posições com relação ao referencial do robô. Conhecendo a posição dos marcos com relação ao referencial do ambiente e com relação ao referencial do robô, é possível determinar a posição do robô com relação ao referencial do ambiente.

A maioria dos sistemas de localização baseados em marcos, utilizam como sensor uma câmera digital. Esse tipo de sensor fornece uma grande quantidade de informação a respeito do ambiente, o que faz com que o método necessite de um tempo alto para o processamento da imagem, dificultando a sua implementação em tempo real. Devido a isso, muitas vezes é necessário que o robô saiba aproximadamente qual é a sua posição atual, geralmente através de odometria, para que apenas uma região da imagem onde o marco provavelmente se encontra seja analisada. Além disso, essa estimativa da posição atual do robô também é útil para que o mesmo possa direcionar a câmera adequadamente para captar os marcos.

Além dos sensores, um sistema de localização baseado em marcos deve possuir um método para identificar os marcos no ambiente e um método para calcular a posição e orientação do robô a partir desses marcos. Diferentes métodos têm sido utilizados para realizar essas duas tarefas, que variam substancialmente dependendo das características do ambiente, dos sensores e dos marcos utilizados.

O problema de localização do robô, através da detecção de marcos, pode ser resumido através do relacionamento entre os referenciais envolvidos: do ambiente $\{W\}$, do robô $\{R\}$, da câmera $\{C\}$ e do marco $\{M\}$. Esses relacionamentos são ilustrados na figura 2.6. O objetivo é se obter a localização do robô com relação ao ambiente (espaço de trabalho) ${}^W_R T$. A localização da câmera em relação ao robô, ${}^R_C T$ é conhecida. A câmera é então utilizada para captar uma imagem do marco, a partir da qual é determinada a localização da câmera com relação ao marco, ${}^C_M T$. Sendo conhecida a localização do marco com relação ao ambiente, ${}^W_M T$ determina-se ${}^W_R T$ como sendo ${}^W_R T = {}^W_M T \cdot {}^M_C T \cdot {}^C_R T$.

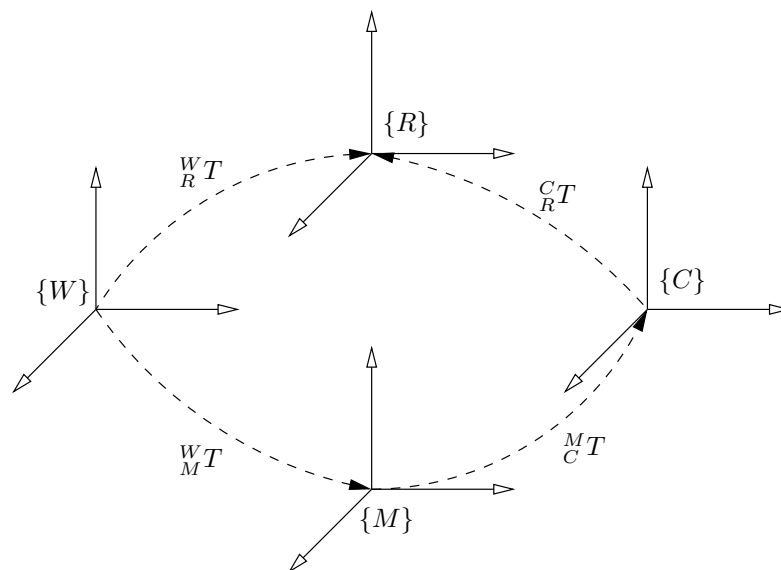


Figura 2.6: Localização usando marcos.

2.2.1 Marcos Artificiais

Os marcos artificiais são objetos ou marcas especialmente desenvolvidos para auxiliar a navegação do robô e que são colocados no ambiente com esse único propósito. Por exemplo, Jang et al. (2002) utilizaram como marcos a figura de um quadrado, dividido

horizontalmente em dois retângulos coloridos (um vermelho e outro azul) para facilitar a sua detecção. Essa figura foi fixada a dois planos perpendiculares, formando um ângulo de 45° com cada um deles. Foi utilizado um robô móvel equipado com uma câmera digital que captava imagens do ambiente. Após a detecção de um marco na imagem, estando o mesmo em uma posição conhecida no ambiente, a posição do robô era determinada com base na posição da câmera com relação ao marco. Já a orientação do robô era determinada com base na linha formada pelo encontro entre os dois planos, já que a posição dessa linha com respeito ao quadrado varia de acordo com a orientação da câmera.

De forma semelhante, Fairfield (2002) utilizou pequenos anéis verdes de plástico dispostos pelo ambiente como marcos artificiais. Coelho e Campos (1998) propoem um sistema para prover em tempo real a posição e orientação de um dirigível com base em imagens captadas por uma câmera. Para isso, eles utilizam um marco visual com propriedades geométricas previamente conhecidas. A partir da imagem do objeto, são calculadas a posição e orientação. O marco é formado por três círculos com distâncias conhecidas entre os centros. Cada círculo é rotulado com um número de orifícios variando de zero a dois. Os círculos são pretos e o marco é colocado contra um fundo claro, a fim de facilitar a segmentação da imagem. Os marcos artificiais são geralmente bem simples e fáceis de detectar, já que são elaborados visando simplificar ao máximo esse processo. Porém, esse tipo de marco possui uma desvantagem, a necessidade de alterar o ambiente em que o robô irá navegar.

2.2.2 Marcos Naturais

Os marcos naturais são objetos ou características já existentes no ambiente e que não possuem originalmente a função de habilitar a navegação do robô. Quando se utiliza

esse tipo de marco não é necessário alterar o ambiente em que o robô irá navegar. Launay et al. (2001) utilizaram como marcos naturais as lâmpadas fluorescentes localizadas no teto do ambiente. Em muitos trabalhos que utilizam marcos naturais para a localização de robôs móveis em ambientes internos, são utilizados como marcos as arestas presentes no ambiente (Vassallo, 1998; Howard e Kitchen, 1997; Clerentin et al., 2002; Branca et al., 2000). Essas arestas são geralmente provenientes de descontinuidades físicas como o encontro entre paredes, portas, janelas, teto ou o piso.

Muitas vezes, a movimentação do robô é efetivada através da identificação de pontos em segmentos de reta. Grupos de retas podem identificar, devido a projeção perspectiva, pontos de fuga que indiquem a direção e o sentido da navegação. Por exemplo, na interseção de duas retas paralelas, como as laterais de uma estrada ou de um corredor (Vassallo, 1998), que se encontram no infinito. O ponto de fuga definido no infinito pode ser determinado na imagem, indicando a direção da navegação.

2.3 Discussão

A odometria é um método que determina a localização de um robô móvel com rodas, baseado na integração das medidas provenientes dos *encoders* das rodas do robô. No entanto, esse não é um método adequado para a estimação da localização de um robô móvel quando usado isoladamente. Os erros inerentes desse método são inevitáveis e representam um problema para trajetórias longas, visto que o seu acúmulo faz com que a posição e orientação medidas divirjam muito da posição e orientação reais. Em função desse problema, é raro encontrar sistemas de navegação de robôs baseados somente nesse método. É comum encontrar sistemas que acrescentam também informação de outros sensores, como uma câmera digital por exemplo, que corrigem

a estimativa da localização obtida por odometria. Essa correção pode ser feita a uma taxa de amostragem sub-múltipla da taxa de amostragem dos *encoders*. Como é possível estimar a evolução temporal da incerteza de odometria, a correção pode ser feita quando a região de incerteza atingir um limite de tolerância especificado.

Capítulo 3

Processamento Digital de Imagens

A visão é um dos sentidos mais importantes do ser humano. É através dela que são realizadas diversas tarefas do dia-a-dia, como por exemplo identificar pessoas, locais e objetos, estimar distâncias, velocidade e altura. A visão computacional pode ser definida como um conjunto de métodos e técnicas através dos quais sistemas computacionais podem interpretar imagens (Wangenheim, 2003). É comum o uso de sistemas de visão computacional para a solução de problemas em diversas áreas. Em robótica móvel, por exemplo, esses sistemas são utilizados para extrair informação do ambiente, auxiliando na determinação da posição e orientação do robô e na percepção de obstáculos.

Para perceber o ambiente ao seu redor, um robô é equipado com um conjunto de sensores que provêm informação a respeito do ambiente. O tipo de sensores aplicados, como também a natureza de suas respostas, definem a maneira como a informação extraída é trabalhada e utilizada pelo sistema. Os sensores de visão mais comumente utilizados em robótica móvel são as câmeras digitais. Esse sensor, provê uma grande quantidade de informação a respeito do ambiente (Schilling, 1990), na forma de imagens digitais. As informações sobre o ambiente em três dimensões (3D) são extraídas

a partir de sua projeção em uma imagem de duas dimensões (2D).

Discutiremos neste capítulo, alguns dos fundamentos e técnicas de processamento digital de imagens aplicados no trabalho. A figura 3.1 ilustra o funcionamento do sistema de visão proposto. Esse sistema tem por objetivo principal proporcionar um método de localização absoluta para o robô, baseado na detecção de marcos no ambiente. Inicialmente é adquirida uma imagem em tons de cinza do ambiente, através de uma câmera digital posicionada no robô. Essa imagem, contendo um marco, é segmentada resultando em uma imagem binarizada. Após isso, a essa imagem binarizada é aplicada a transformada de Hough para retas, com o objetivo de detectar os marcos utilizados, que são linhas retas no piso do ambiente onde o robô irá navegar. Com isso, são extraídas das imagens informações do ambiente, de forma adequada à sua percepção autônoma pelo robô, para que assim seja possível determinar a sua localização.

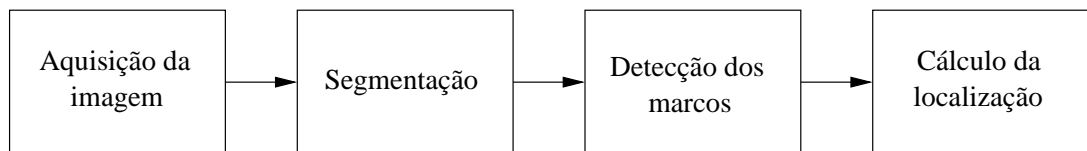


Figura 3.1: Diagrama de funcionamento do sistema de visão utilizado.

3.1 Modelo da Câmera

Para que se possa determinar a localização de um robô, utilizando as informações providas por um sistema de visão computacional, é necessário a compreensão dos aspectos geométricos do processo de formação da imagem capturada por uma câmera digital. A partir dessas imagens, é possível determinar o posicionamento de um marco com relação à câmera. De forma resumida, estamos interessados em determinar

um modelo de projeção do mundo real 3D em uma imagem 2D. O processo de projeção causa a perda da informação de profundidade do ambiente. No entanto, essa informação pode ser obtida através da utilização de múltiplas câmeras, de múltiplas imagens de uma mesma câmera ou ainda do conhecimento das relações existentes entre cada ponto da imagem e os objetos que compoem a cena.

Um modelo de câmera bastante utilizado em trabalhos que utilizam visão computacional é o *pinhole* (Gonzalez e Woods, 2000), com projeção em perspectiva, o qual também será adotado neste trabalho. Esse modelo, ilustrado na figura 3.2, se baseia na passagem dos raios luminosos por um pequeno orifício na entrada da câmera e por sua projeção em uma superfície plana. O plano I se chama plano da imagem, no qual se forma a imagem através da projeção. O plano F é conhecido como plano focal, onde se localiza o orifício da câmera.

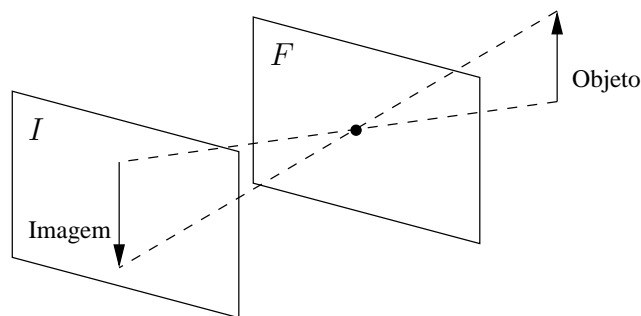


Figura 3.2: Formação da imagem em uma câmera perspectiva.

Considere agora a projeção de um ponto no mundo real p_c em um ponto da imagem p_i , conforme é ilustrado na figura 3.3. O ponto p_c possui coordenadas (x_c, y_c, z_c) , definidas no referencial da câmera $\{C\}$. Já o ponto p_i possui coordenadas (x_i, y_i) definidas no referencial da imagem $\{I\}$. A origem de $\{C\}$ representa o centro ótico, cuja distância ao referencial $\{I\}$ é λ (distância focal). A reta que passa pelo centro ótico e é perpendicular ao plano I é chamado de eixo ótico.

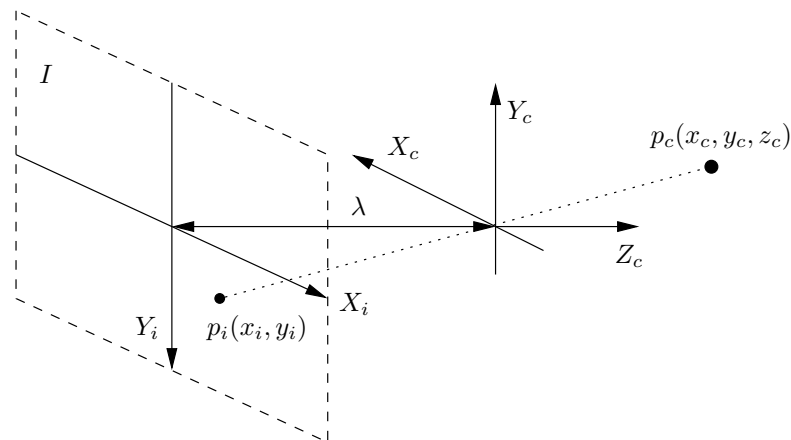


Figura 3.3: Modelo de projeção em perspectiva da câmera.

É fácil mostrar, por semelhança entre triângulos, que a relação existente entre as coordenadas desses pontos nos dois referenciais são dadas pelas equações:

$$x_i = \frac{\lambda}{z_c} x_c \quad \text{e} \quad y_i = \frac{\lambda}{z_c} y_c \quad (3.1)$$

3.2 Representação de Imagens

No processo de armazenamento, as imagens contínuas são convertidas para uma representação discreta. Na forma contínua, a imagem é representado por uma função bidimensional $I(u, v)$, onde I representa a intensidade de brilho de um ponto tridimensional que compõe a cena e (u, v) são as coordenadas espaciais da projeção desse ponto na imagem.

Um computador digital processa dados discretos representados de forma binária. Por esse motivo, é necessário que as imagens sejam digitalizadas para o seu posterior processamento, ou seja, é necessário que as imagens contínuas sejam discretizadas espacialmente e quantizadas em intensidade. A imagem digitalizada é representada

por uma função discreta $I(x, y)$, onde x e y representam as coordenadas discretas de cada ponto da imagem digital, chamado de *pixel* (*picture element*), e I representa a intensidade de brilho de cada *pixel*. O valor de I também é conhecido como o nível de cinza do *pixel*, sendo também um valor discretizado assumindo valores em um intervalo finito.

Para exemplificar, considere uma imagem em escala de cinza com dimensões Largura x Altura = 512x256 *pixels*. Nesse caso, considerando a origem do sistema de coordenadas da imagem no seu centro, teremos que as coordenadas dos *pixels* poderão assumir os seguintes valores: $x = -256, -255, -254, \dots, 255$ e $y = -128, -127, -126, \dots, 127$. Já os níveis de cinza para cada ponto poderão assumir os valores: $I(x, y) = 0, 1, 2, \dots, 255$. Dessa forma, o valor de brilho mínimo, 0, corresponde ao preto e o valor de brilho máximo, 255, corresponde ao branco.

Uma imagem digital pode ser considerada como sendo uma matriz, cujos índices de linhas e de colunas identificam um *pixel* na imagem, e o correspondente valor do elemento da matriz identifica o nível de cinza naquele *pixel*. Quanto mais pixels uma imagem possuir melhor é a sua resolução e qualidade. No entanto, um aumento no número de *pixels* também acarreta um aumento no tempo de processamento da mesma.

3.3 Segmentação

Após a obtenção de uma imagem digital, o próximo passo trata de realizar um pré-processamento nesta imagem. O objetivo desse pré-processamento é melhorar a imagem, de forma a facilitar a detecção dos marcos pelo método utilizado, a transformada de Hough, que será detalhada na próxima seção. Esse pré-processamento realiza uma

segmentação na imagem, utilizando basicamente técnicas de limiarização para a remoção de ruído e isolamento de regiões que possuam uma grande probabilidade de conter a informação desejada.

Geralmente, em processamento digital de imagens, a segmentação automática é uma tarefa bastante difícil. No entanto, essa tarefa é de extrema importância, já que um procedimento de segmentação robusto auxilia substancialmente a solução bem sucedida do problema. Neste trabalho, onde o objetivo é o reconhecimento de marcos que são constituídos por linhas retas no chão do ambiente, a segmentação é utilizada para extrair ao máximo os *pixels* que constituem as retas daqueles que constituem o piso.

3.3.1 Histograma

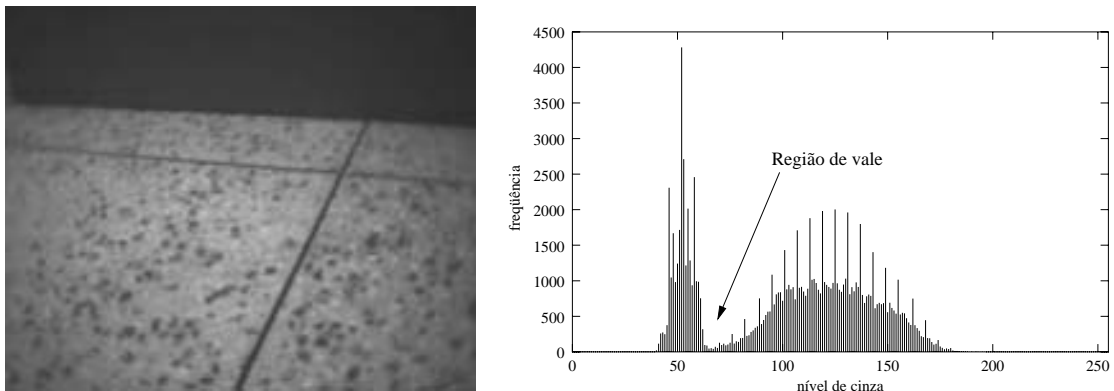
Considere r uma variável que representa os níveis de cinza dos *pixels* de uma imagem digital, onde r varia em um intervalo discreto de tamanho L : $[0, l-1]$. O histograma de uma imagem digital, com níveis de cinza variando no intervalo citado, é representado pela função discreta:

$$h(r) = \frac{n_r}{n}$$

onde n_r é o número de *pixels* na imagem com o nível de cinza r e n é o número total de *pixels* na imagem (Gonzalez e Woods, 2000).

O histograma representa a frequência de ocorrência dos níveis de cinza em uma imagem, nos dando a estimativa da probabilidade de ocorrência de um determinado nível de cinza. Através de sua análise, é possível obter informações úteis que podem ser aplicadas na segmentação da imagem, como a concentração de *pixels* em torno de

uma região de níveis de cinza, caracterizando a existência de elementos na imagem. Na figura 3.4 é apresentado o histograma de uma imagem. Esse histograma apresenta uma região de vale, que corresponde a uma região com pouca frequência de *pixels* e que separa duas outras regiões com grande frequência de *pixels*. Nesse caso, a região de vale separa os *pixels* pertencentes ao chão dos *pixels* pertencentes a parede.



(a)

(b)

Figura 3.4: (a) Uma imagem e (b) seu histograma.

3.3.2 Limiarização

A limiarização consiste no processo onde uma imagem digital com vários níveis de cinza é transformada em uma imagem binária, que possui apenas *pixels* pretos e brancos. Na limiarização, são analisadas as similaridades entre os níveis de cinza, de forma a extrair os objetos de interesse através da seleção de um valor limiar T , que separa os agrupamentos de níveis de cinza. Uma imagem limiarizada $g(x, y)$ é definida como:

$$\begin{aligned}
 g(x, y) &= 1 \text{ se } f(x, y) > T \\
 &= 0 \text{ caso contrário}
 \end{aligned}$$

onde $f(x, y)$ corresponde ao nível de cinza do ponto. Ao final deste processo, os *pixels* rotulados com 1 correspondem aos objetos e os *pixels* rotulados com 0 correspondem ao fundo. O objetivo desta transformação é eliminar da imagem todos os *pixels* que não forem relevantes a análise do problema (Gonzalez e Woods, 2000).

Embora esse processo seja simples e bastante rápido, ele apresenta dificuldades na determinação do valor T , ou seja, determinar o ponto de separação das regiões do fundo e do objeto. A técnica mais simples para determinar esse valor é a do particionamento do histograma da imagem por um limiar único T , através da busca de uma região de vale. A partir desse valor, a segmentação é realizada varrendo-se a imagem, *pixel a pixel*, e rotulando-se cada *pixel* como sendo do objeto ou do fundo, dependendo se o nível de cinza daquele *pixel* for maior ou menor que T .

Porém, como o sucesso desse método depende inteiramente da qualidade do particionamento do histograma, ele depende também da qualidade e da natureza da imagem. Um fator que influencia bastante na qualidade da imagem é a iluminação. No caso de uma iluminação não uniforme, o estabelecimento de um limiar se torna mais difícil. A figura 3.5, mostra o histograma de uma imagem onde não é possível perceber uma região de vale. Com isso, torna-se difícil realizar a limiarização mostrada para separar as retas mostradas na figura do restante do piso.

Nesse caso, as vezes é necessário a utilização de dois limiares, fazendo com que os *pixels* pertencentes aos objetos sejam identificados quando o seu nível de cinza estiver dentro do intervalo formado pelos dois limiares.

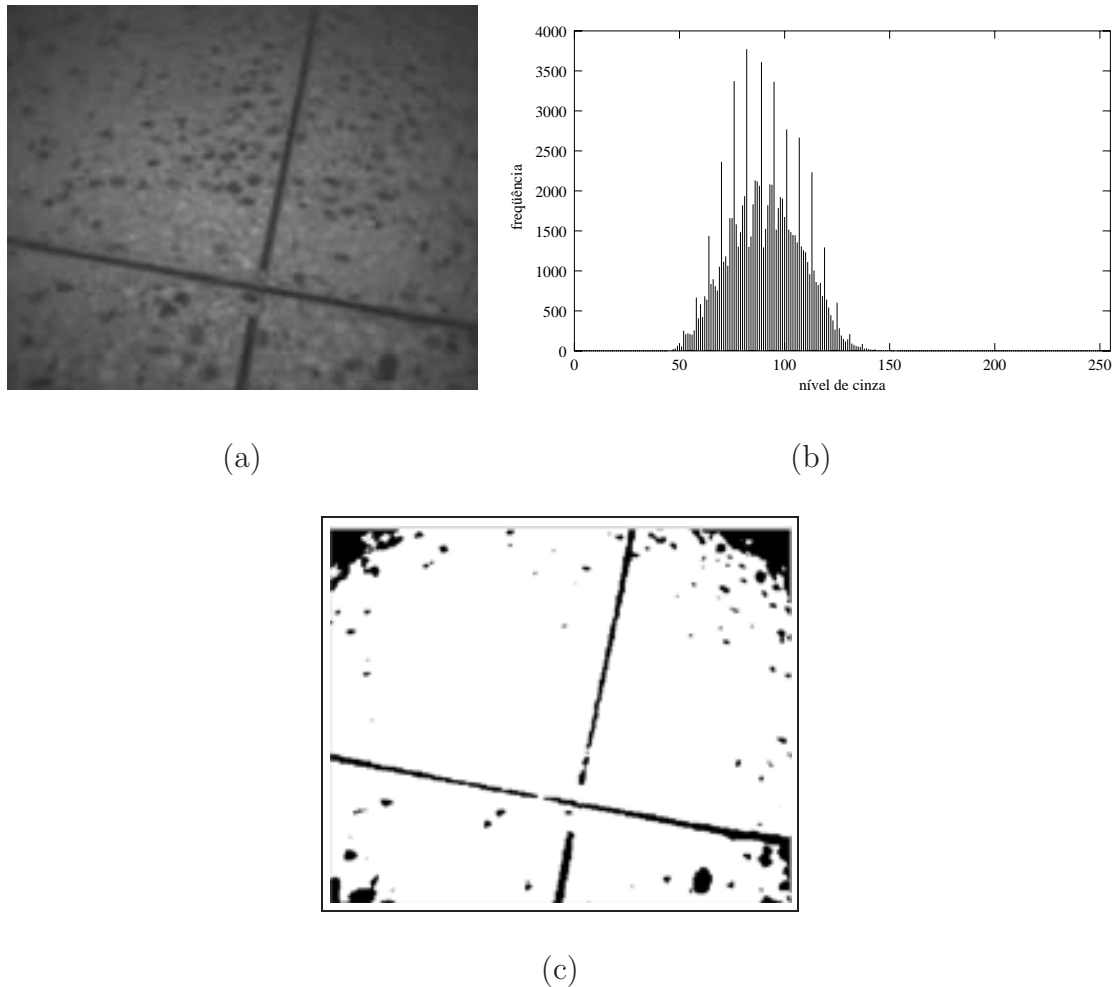


Figura 3.5: (a) Uma imagem, (b) seu histograma e (c) a imagem limiarizada.

3.4 A Transformada de Hough

A transformada de Hough (Hough, 1962) é um método utilizado para detectar em uma imagem digital uma classe de formas geométricas conhecida e que pode ser representada como uma curva paramétrica, como por exemplo retas, círculos e elipses. Esse método vem sendo muito utilizado na detecção de características marcantes, presentes em imagens digitais, como em aplicações que envolvem a navegação de robôs móveis em ambientes interiores (Grisetti et al., 2000; Grisetti et al., 2002; Chen e

Tsai, 1998; Schmidt, 2000). O resultado da aplicação dessa transformada na detecção de retas é ilustrado na figura 3.6. Dado um conjunto de pontos em uma imagem, são determinadas as retas mais marcantes formadas por esses pontos, ou seja, as retas formadas pelas maiores concentrações de pontos co-lineares.

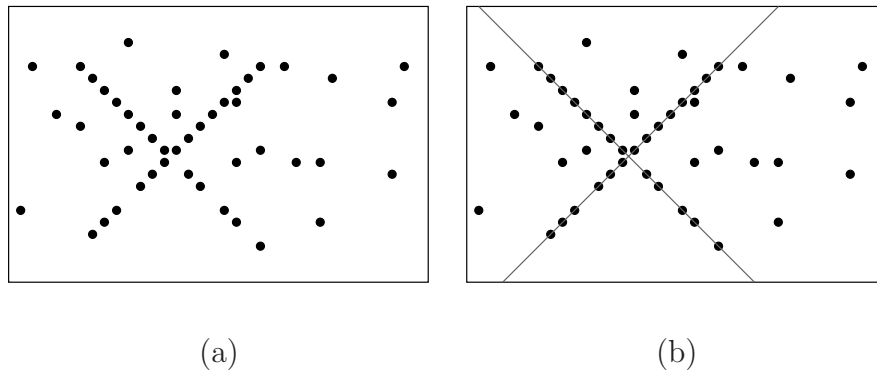


Figura 3.6: (a) Conjunto de pontos em uma imagem digital e (b) as retas detectadas pela transformada de Hough.

Essa transformada realiza um mapeamento entre o espaço cartesiano da imagem e o espaço de parâmetros em que a curva foi definida. Dessa forma, os pontos da curva procurada são concentrados no espaço de parâmetros de acordo com as características que os unem no espaço cartesiano. No caso de retas, um conjunto de pontos co-lineares no espaço cartesiano irá gerar uma acumulação em um ponto no espaço de parâmetros. Esse ponto acumulador define os parâmetros da reta que passa por esses pontos cartesianos. Em uma imagem digital, geralmente a transformada de Hough é utilizada após um pré-processamento da mesma, como uma limiarização ou o cálculo do gradiente, limitando o número de *pixels* que são processados pela transformada. No caso de uma limiarização, são processados apenas os *pixels* cujo nível de cinza se encontra abaixo de um limiar.

3.4.1 Transformada de Hough para Retas

Uma possível parametrização para uma reta no plano X - Y é dada pela equação:

$$y = ax + b \quad (3.2)$$

onde os parâmetros a e b indicam, respectivamente, a inclinação da reta com relação ao eixo X e o ponto onde a reta intercepta o eixo Y . Cada reta no plano cartesiano X - Y , pode ser representada por um único ponto no plano A - B (plano de parâmetros), correspondente aos parâmetros dessa reta. A transformada de Hough consiste em calcular, para cada ponto analisado, os parâmetros de todas as possíveis retas que passam por esse ponto. Cada uma dessas retas é calculada de acordo com a equação:

$$b = y - ax \quad (3.3)$$

Com isso, para cada ponto analisado no plano cartesiano, o conjunto de todos os pares de parâmetros calculados formará uma reta no plano de parâmetros. Repetindo esse processo para todos os pontos do plano cartesiano, obteremos várias retas no plano de parâmetros (uma para cada ponto do plano cartesiano). A interseção dessas retas, indicará uma co-linearidade entre pontos no plano cartesiano, sendo que o ponto de interseção corresponderá aos parâmetros a e b da reta que passa por esses pontos co-lineares.

Para exemplificar o método de Hough, considere a reta r mostrada na figura 3.7(a). Para um ponto (x_i, y_i) pertencente a essa reta, o conjunto dos parâmetros calculados formam uma reta no plano de parâmetros, da mesma forma ocorre para um ponto (x_j, y_j) . Como pode ser visto na figura 3.7(b), as duas retas formadas se interceptam no ponto $p = (a_1, b_1)$, que correspondem aos parâmetros da reta r . Se repetirmos

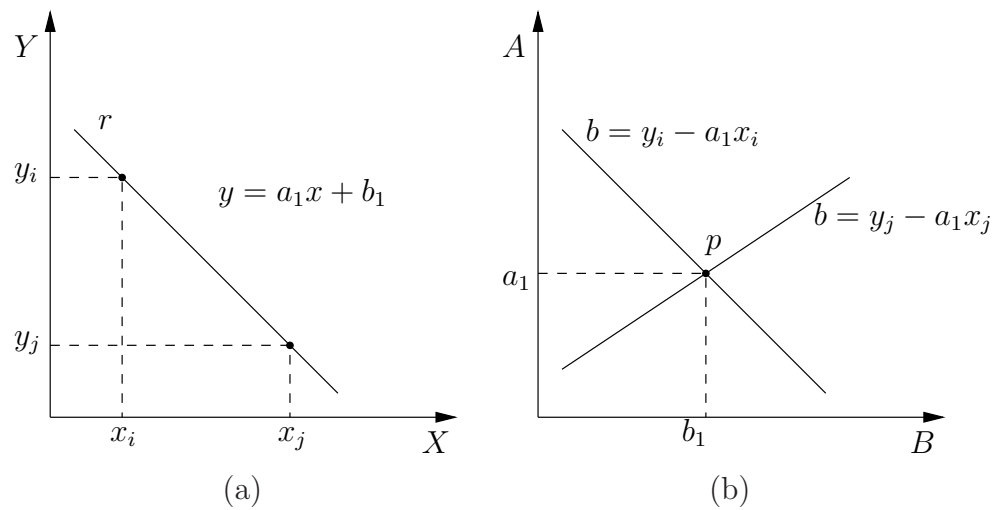


Figura 3.7: Exemplo de detecção de uma reta. (a) reta no plano cartesiano e (b) Interseção das retas no plano de parâmetros.

esse procedimento para os demais pontos de r , outras retas se formarão no plano de parâmetros e todas elas irão se interceptar no ponto p .

3.4.2 Representação Normal de uma Reta

A representação de uma reta em função dos parâmetros a e b mostrada na equação 3.2 não é adequada para implementar computacionalmente a transformada de Hough, já que esses parâmetros podem assumir infinitos valores. Isso pode ser verificado quando uma reta é paralela ao eixo Y , onde o valor de a tende a infinito. Para solucionar esse problema, foi proposta a utilização de outra representação para a reta (Duda e Hart, 1972). A representação adotada foi a representação normal, dada pela equação:

$$\rho = x \cos \phi + y \sin \phi \quad (3.4)$$

A representação normal, proporciona um espaço de parâmetros finito, com seus limites conhecidos. Os parâmetros ρ e ϕ representam, respectivamente, o comprimento do vetor, que passa pela origem, normal à reta e o ângulo formado entre esse vetor

e o eixo X , conforme mostra a figura 3.8. Utilizando essa representação, não teremos mais a formação de retas no espaço de parâmetros e sim de curvas senoidais. Um conjunto de pontos colineares no plano cartesiano irá gerar um conjunto de senóides no espaço de parâmetros, que terão um ponto de interseção que identifica a presença da reta que une os pontos no espaço cartesiano.

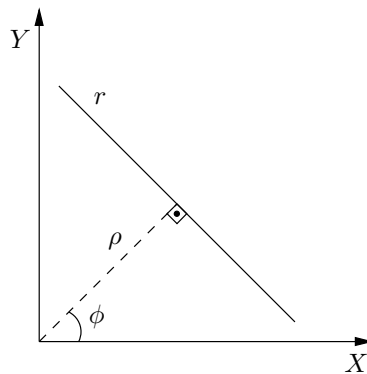


Figura 3.8: Representação normal de uma reta.

Como exemplo, considere agora o conjunto de pontos mostrados na figura 3.9(a). Para o ponto A , o conjunto dos parâmetros calculados formam uma senóide no plano de parâmetros, de acordo com a equação 3.4. Da mesma forma ocorre para os pontos B , C , D , E e F , como pode ser visto na figura 3.9(b). As senóides formadas se interceptam em vários pontos, indicando uma co-linearidade entre pontos do plano cartesiano. A identificação da reta consiste em determinar o ponto do espaço de parâmetros que possui o maior número de intersecções; nesse exemplo, o ponto p , que corresponde aos parâmetros da reta mais visível na imagem, formada pelos pontos A , B , C e D . Note ainda que foi utilizado uma faixa de valores de ϕ entre 0° e 180° , já que para mapear todas as possíveis retas que passam por um ponto, é necessário apenas uma variação de 180° desse ângulo, pois uma mesma reta pode ser representada por dois ângulos ϕ_1 e $\phi_1 + 180^\circ$.

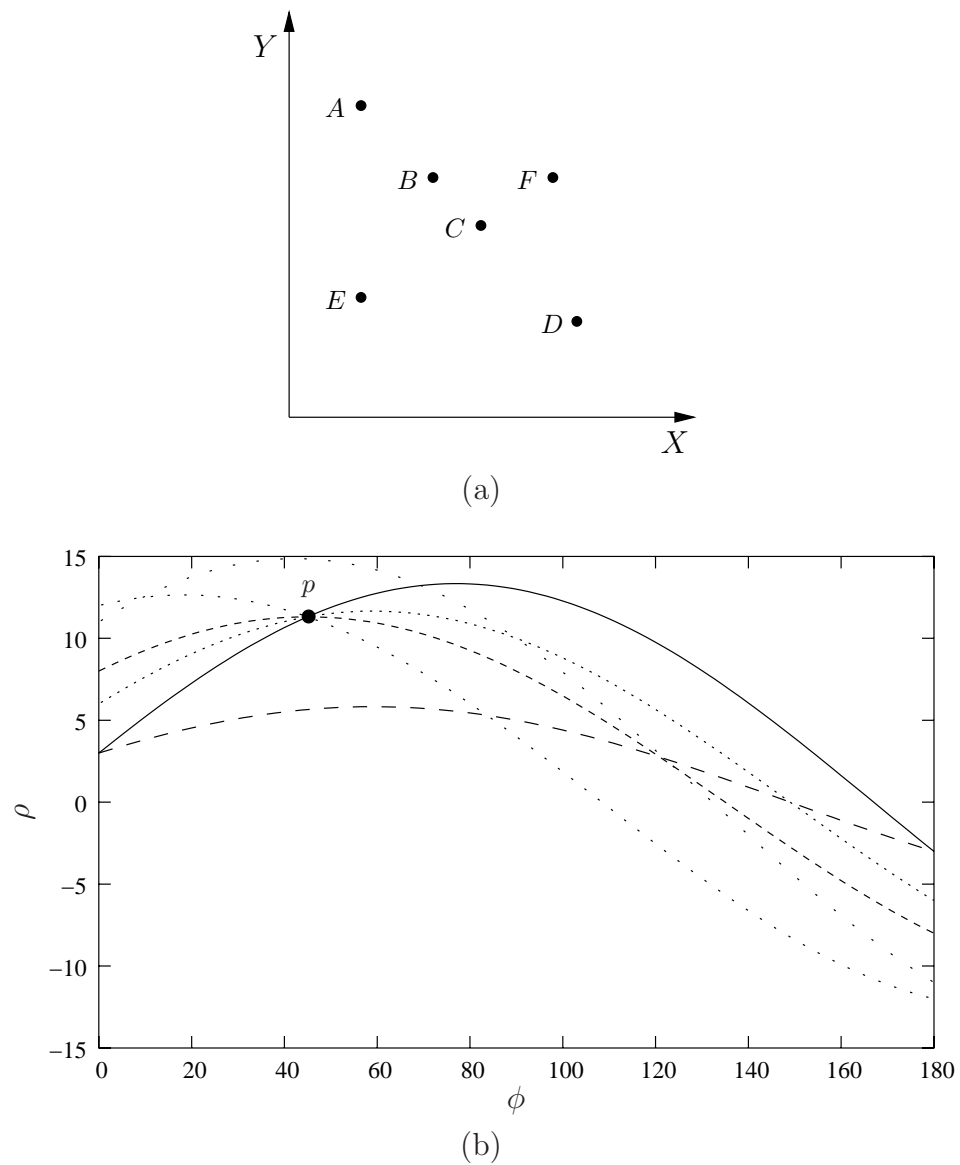
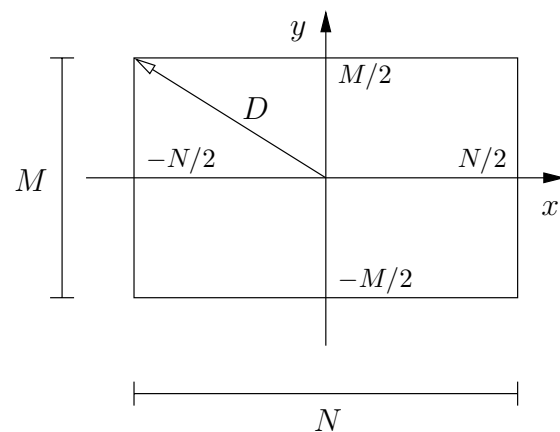


Figura 3.9: Transformada de Hough utilizando a representação normal: (a) Imagem com uma reta. (b) Transformada de Hough obtida para três pontos.

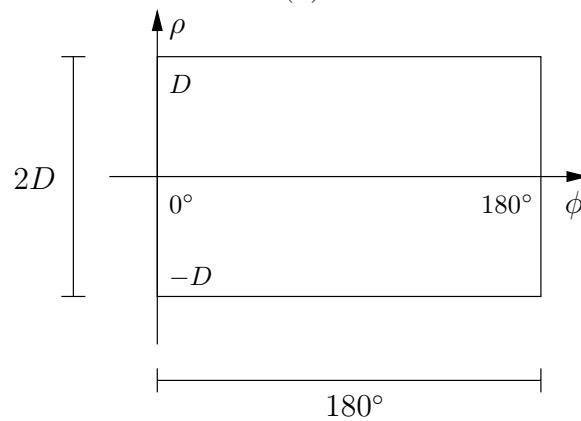
3.4.3 Processamento da Transformada de Hough

Em termos computacionais, quando a transformada de Hough é aplicada em imagens digitais, o plano cartesiano e o plano de parâmetros precisam ser representados

de forma quantizada. O plano cartesiano corresponderá à imagem digital, que é discretizada de acordo com o exposto na seção 3.2. Uma imagem com dimensões $M \times N$ *pixels*, pode ser representada em x e em y através dos intervalos $[-M/2, M/2]$ e $[-N/2, N/2]$, respectivamente, considerando a origem no centro da imagem, como mostrado na figura 3.10(a).



(a)



(b)

Figura 3.10: (a) Discretização de uma imagem e (b) do seu espaço de parâmetros.

De forma semelhante, o espaço de parâmetros também é quantizado, sendo representado por uma matriz bi-dimensional, chamada matriz acumuladora, onde cada

elemento dessa matriz corresponde a um ponto do espaço de parâmetros. O parâmetro ρ terá seus limites definidos de acordo com a diagonal D , de um dos quadrantes definidos pelo sistema de coordenadas da imagem. Assim sendo, ρ varia dentro do intervalo $[-D, D]$, como mostrado na figura 3.10(b). Já o ângulo ϕ , de acordo com o citado anteriormente, terá uma variação dentro do intervalo $[0, 180^\circ]$.

A quantização dos parâmetros ρ e ϕ ($\Delta\rho$ e $\Delta\phi$ respectivamente) necessita de um estudo mais aprofundado, já que eles afetam o tempo de processamento e a acurácia da transformada de Hough. Quando a transformada é calculada para um ponto (x, y) da imagem, para cada valor de ϕ dentro do seu intervalo, o valor correspondente de ρ é calculado, através da equação 3.4. O elemento correspondente a esses dois valores, ρ e ϕ , na matriz acumuladora, é incrementado em uma unidade. Ao final desse processo, é necessário uma varredura na matriz acumuladora em busca dos seus máximos. Esses máximos corresponderão aos pontos de interseção das senóides, que são as retas detectadas na imagem. A figura 3.11, mostra o resultado da detecção de duas retas em uma imagem de dimensões 352×268 *pixels*, utilizando $\Delta\rho = \Delta\phi = 1$. Vale salientar que o tempo de processamento, utilizado na detecção dessas duas retas foi bastante alto, cerca de $500ms$.

Dessa forma, uma quantização apropriada de ρ e ϕ deve ser escolhida. O tempo de processamento pode aumentar, caso sejam aumentados o número de níveis de quantização do espaço de parâmetros. No entanto, caso esse número seja diminuído, o valor dos parâmetros calculados será deteriorado. É necessário, portanto, um estudo para estimar uma relação eficaz entre $\Delta\rho$ e $\Delta\phi$ (Hanahara et al., 1988; Veen e Groen, 1981).

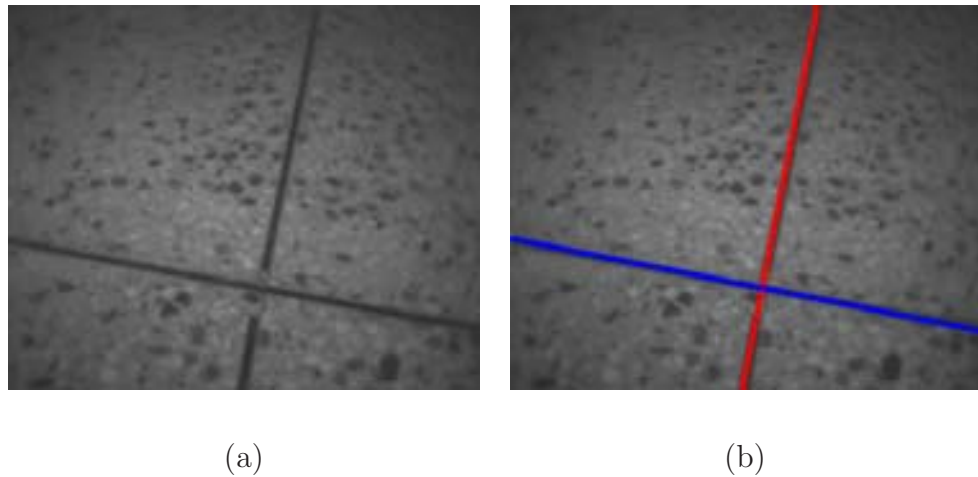


Figura 3.11: (a) Uma imagem contendo duas retas e (b) as retas detectadas.

3.4.4 Variações da Transformada de Hough

Como a transformada de Hough trabalha com base no cálculo de todas as possíveis retas que passam pelos pontos analisados, o seu tempo de processamento em uma imagem torna-se elevado. Visando reduzir esse problema foram desenvolvidas algumas variações dessa técnica, dentre elas podemos citar a transformada probabilística de Hough (PHT - *Probabilistic Hough Transform*) e a transformada randômica de Hough (RHT - *Randomized Hough Transform*), que utilizam apenas parte do conjunto de dados para realizar a transformada de Hough. Tanto a PHT quanto a RHT possuem ainda um conjunto de versões que utilizam modificações nos algoritmos padrões.

A transformada probabilística (Kiryati et al., 1991) afirma que resultados semelhantes aos da transformada padrão, na detecção de uma curva, podem ser obtidos realizando a transformada padrão apenas em uma percentagem de *pixels* da imagem, onde os *pixels* são escolhidos de acordo com uma função de densidade de probabilidade uniforme definida em cima da imagem. Já a transformada randômica (Xu et al., 1990) trabalha escolhendo randômicamente um par de pontos da imagem e calculando os

parâmetros da reta que passa por eles. O elemento correspondente na matriz acumuladora é incrementado. Esse procedimento é repetido um número n de vezes, onde n é um valor bem menor que o número de pares de pontos da imagem. Após isso, os parâmetros da reta procurada são determinados, achando-se o valor máximo no acumulador. Caso seja necessário achar outra reta, os pontos pertencentes à primeira reta são removidos de forma a diminuir os pontos a serem analisados, e o processo descrito é repetido.

Após uma análise desses dois algoritmos, Kiryati et al. (2000) concluíram que para imagens de alta qualidade a RHT é consideravelmente mais rápida do que a PHT. No entanto, no processamento de imagens ruidosas e com erros, a PHT é mais robusta do que a RHT. Todas essas variações da transformada de Hough visam acelerar o processamento da mesma, através da diminuição da quantidade de *pixels* analisados. Neste trabalho, essa redução será obtida através da utilização de uma técnica proposta, que mapeia o erro de odometria em uma região a ser analisada na imagem, dessa forma será utilizada apenas a transformada de Hough padrão.

3.5 Discussão

Neste capítulo, mostramos algumas técnicas de processamento de imagens que serão utilizadas no sistema de localização proposto. Essas técnicas serão aplicadas de forma a implementar um sistema de localização absoluta para o robô. A transformada de Hough é a principal dessas técnicas, que será utilizada na detecção dos marcos utilizados pelo sistema, que são linhas retas no piso do ambiente de trabalho do robô. Percebe-se que a transformada de Hough exige um tempo de processamento alto para as imagens, não atendendo aos requisitos da aplicação. Daí surge a necessidade de

incorporar ao sistema informação adicional, baseada em um conjunto de heurísticas que, associadas a transformada de Hough, possibilitará uma detecção mais eficaz dos marcos, de acordo com o proposto no próximo capítulo.

Capítulo 4

Sistema Proposto

Neste capítulo, será apresentada uma descrição detalhada do sistema de localização implementado. O sistema é proposto de forma a determinar a localização de um robô móvel com rodas, utilizando uma plataforma com acionamento diferencial, em um ambiente semi-estruturado, onde são conhecidas previamente algumas de suas características. A estratégia adotada associa um método de localização relativa e um método de localização absoluta. O método de localização relativa utilizado é a odometria, enquanto que o método de localização absoluta consiste na detecção de marcos visuais no ambiente, através do processamento de imagens digitais obtidas por uma câmera colocada no robô. Esses dois métodos são associados visando explorar as características mais vantajosas de cada um deles, de forma a implementar um sistema adequado e robusto para solucionar o problema. Dessa forma, é proposta a redução do tempo de processamento na detecção dos marcos e obtenção da localização absoluta do robô. Essa redução é obtida através do processamento de apenas uma pequena região da imagem, que possui grande probabilidade de conter o marco procurado. Essa região é obtida através de um mapeamento da estimativa do erro máximo cometido no cálculo da localização do robô pela odometria na imagem capturada.

É importante citar que não é objetivo deste trabalho propor alguma técnica para o controle durante a navegação do robô. Espera-se que esteja à disposição um controlador já implementado e eficiente. Medeiros (1998b) apresenta algumas das arquiteturas de controle mais utilizadas em robótica móvel. Existe ainda um grande número de trabalhos nessa área, que propõem controladores e estratégias de controle facilmente aplicáveis ao sistema proposto.

O ciclo de funcionamento do sistema de localização implementado, ilustrado no diagrama da figura 4.1, pode ser descrito da seguinte forma:

1. Determinação da localização relativa do robô utilizando odometria;
2. Estimção do erro máximo de odometria calculado;
3. Aquisição de uma imagem do piso do ambiente;
4. Mapeamento do erro máximo de odometria em uma região da imagem com maior probabilidade de conter o marco procurado;
5. Segmentação dessa região da imagem;
6. Aplicação da transformada de Hough a essa região da imagem segmentada, a fim de detectar o marco procurado;
7. Determinação da localização absoluta do robô com base na localização do marco identificado no ambiente;
8. Correção da localização relativa a partir da localização absoluta.

Esse procedimento é constantemente repetido enquanto o robô se movimenta pelo ambiente. No entanto, vale ressaltar que o período de amostragem dos passos 1 e 2

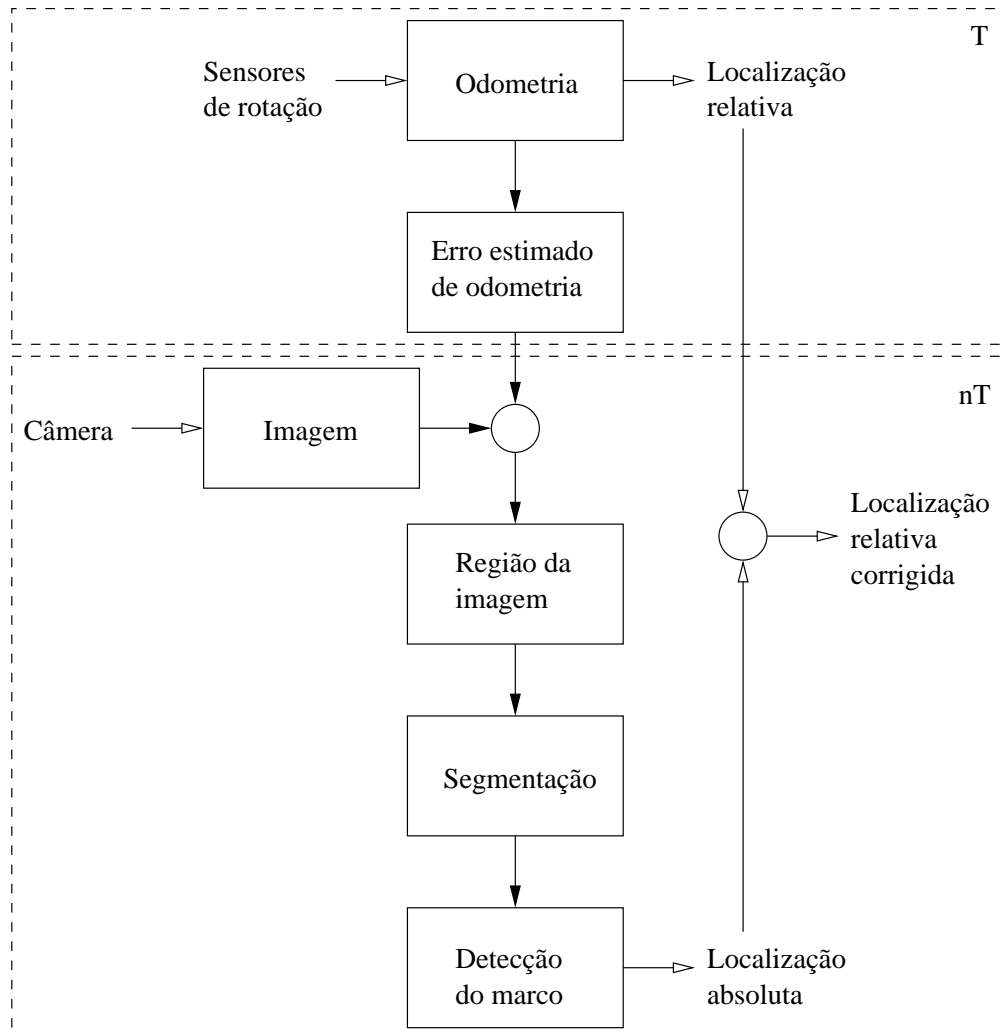


Figura 4.1: Diagrama de funcionamento do sistema de localização.

(T) é menor do que os demais passos (nT) , ou seja, os passos 1 e 2 serão executados algumas vezes (n vezes) antes que os demais passos sejam executados. Nas próximas seções, serão detalhadas cada uma das etapas envolvidas nesse processo.

4.1 Sistema de Odometria

Neste trabalho, será utilizada uma plataforma robótica com acionamento diferencial, cujo modelo de odometria foi apresentado no capítulo 2. O método de odometria é utilizado pelo sistema para obter uma localização relativa do robô. Essa localização é obtida utilizando dois sensores, um em cada roda, a partir dos quais são medidas as rotações de cada uma das rodas do robô. Cada um desses sensores é acoplado ao eixo de rotação da roda correspondente, possibilitando dessa forma obter a informação a respeito da posição de cada uma das rodas em determinado instante de tempo.

O cálculo da localização do robô usando odometria é integrativo. Com isso, para iniciar esse cálculo é necessário saber a sua localização inicial $(x(0), y(0), \theta(0))$. A partir disso a localização atual é calculada com base na localização anterior e no número de pulsos lidos em cada sensor de rotação (N_d e N_e). Para realizar esse cálculo é necessário também o conhecimento dos seguintes parâmetros: o comprimento do eixo do robô (b), os raios de suas rodas (r_d e r_e) e a resolução dos sensores (N_{res}), que é o número de pulsos lidos em uma revolução das rodas. Esse cálculo é feito utilizando as equações de odometria para uma plataforma robótica com acionamento diferencial, deduzidas no capítulo 2 e repetidas aqui por convêniência:

$$\begin{aligned}
 x(t + \Delta t) &= x(t) + (N_d r_d + N_e r_e) \frac{\pi}{N_{res}} \cos \theta(t) \\
 y(t + \Delta t) &= y(t) + (N_d r_d + N_e r_e) \frac{\pi}{N_{res}} \sen \theta(t) \\
 \theta(t + \Delta t) &= \theta(t) + \frac{2\pi}{b N_{res}} (N_d r_d - N_e r_e)
 \end{aligned} \tag{4.1}$$

De acordo com o citado anteriormente, a odometria é um método de localização

bastante utilizado na robótica móvel. No entanto, devido a ocorrência de erros durante o movimento do robô, a precisão deste método é muito baixa, necessitando que seus valores sejam validados de tempos em tempos. Esses erros são classificados como erros não-sistemáticos e erros sistemáticos. Como discutido no capítulo 2, os erros sistemáticos são previsíveis, ao contrário dos erros não-sistemáticos. Apesar dessa aleatoriedade dos erros não-sistemáticos é possível deduzir um modelo evolutivo para estimar o erro máximo cometido pelo método, de tal forma a englobar essas duas categorias, como será visto na próxima seção.

4.2 Estimativa dos Erros de Odometria

Nesta seção será descrito um método para expressar a incerteza da localização obtida pelo sistema de odometria. Esse método, que será apresentado aqui de forma resumida, foi descrito inicialmente por Chenavier e Crowley (1992), servindo de base para uma série de outros trabalhos, como por exemplo o proposto por Irie et al. (2003). Ele é baseado no cálculo da matriz de covariância da estimativa do erro associado à localização do robô obtida pela odometria. A partir dessa covariância, obtém-se uma região ao redor da localização estimada pela odometria com uma determinada probabilidade de conter a localização real do robô. Para o cálculo da matriz de covariância, inicialmente definimos os vetores de configuração \mathbf{q} e de deslocamento do robô \mathbf{u} como sendo:

$$\begin{aligned}\mathbf{q}(t) &= \begin{bmatrix} x(t) & y(t) & \theta(t) \end{bmatrix}^T \\ \mathbf{u}(t) &= \begin{bmatrix} \Delta l(t) & \Delta \theta(t) \end{bmatrix}^T\end{aligned}$$

Reescrevendo as equações de odometria em função dos deslocamentos linear (Δl)

e angular ($\Delta\theta$) do robô, temos:

$$x(t + \Delta t) = x(t) + \Delta l \cos \theta(t)$$

$$y(t + \Delta t) = y(t) + \Delta l \sin \theta(t)$$

$$\theta(t + \Delta t) = \theta(t) + \Delta\theta(t)$$

que escritas de forma matricial se tornam:

$$\mathbf{q}(t + \Delta t) = \mathbf{q}(t) + \begin{bmatrix} \Delta l \cos \theta(t) \\ \Delta l \sin \theta(t) \\ \Delta\theta(t) \end{bmatrix} \quad (4.2)$$

Dessa forma, a configuração atual do robô pode ser escrita como uma função da sua configuração anterior e do seu vetor de deslocamento, ou seja,

$$\mathbf{q}(t + \Delta t) = f(\mathbf{q}(t), \mathbf{u}(t))$$

No entanto, essa equação não fornece a exata localização do robô e sim uma estimativa dessa localização. Isso ocorre, já que não são considerados alguns fatores que provocam erros de odometria. Dessa forma, é necessário introduzir nessa equação um vetor de ruído \mathbf{n} , devido a deformações, deslizamento das rodas, vibrações e outros erros desconhecidos, resultando em:

$$\mathbf{q}(t + \Delta t) = f(\mathbf{q}(t), \mathbf{u}(t)) + \Delta\mathbf{n}(t) \quad (4.3)$$

O valor $\Delta\mathbf{n}$ é a integração de \mathbf{n} durante o intervalo de tempo Δt . O ruído $\Delta\mathbf{n}$ é considerado Gaussiano, com média zero e covariância dada por uma matriz \mathbf{Q} . Esse

ruído gera erros nas medições dos deslocamentos das rodas e erros no cálculo da localização do robô, ou seja, os valores reais de \mathbf{q} e \mathbf{u} serão dados por:

$$\mathbf{q}(t) = \hat{\mathbf{q}}(t) + \Delta\mathbf{q}$$

$$\mathbf{u}(t) = \hat{\mathbf{u}}(t) + \Delta\mathbf{u}$$

onde $\hat{\mathbf{q}}$ é o valor calculado de \mathbf{q} , $\Delta\mathbf{q}$ é o erro cometido no cálculo de $\hat{\mathbf{q}}$, $\hat{\mathbf{u}}$ é o valor medido de \mathbf{u} e $\Delta\mathbf{u}$ é o erro na medição de $\hat{\mathbf{u}}$. Os vetores $\hat{\mathbf{q}}$ e $\hat{\mathbf{u}}$ são definidos como:

$$\begin{aligned}\hat{\mathbf{q}}(t) &= \begin{bmatrix} \hat{x}(t) & \hat{y}(t) & \hat{\theta}(t) \end{bmatrix}^T \\ \hat{\mathbf{u}}(t) &= \begin{bmatrix} \Delta\hat{l}(t) & \Delta\hat{\theta}(t) \end{bmatrix}^T\end{aligned}$$

Podemos linearizar a equação 4.3, através de sua expansão em uma série de Taylor de primeira ordem, em torno de $\hat{\mathbf{q}}$ e $\hat{\mathbf{u}}$, da seguinte forma:

$$\begin{aligned}\mathbf{q}(t + \Delta t) &= f(\mathbf{q}(t), \mathbf{u}(t)) + \Delta\mathbf{n}(t) \\ &= f(\hat{\mathbf{q}}(t) + \Delta\mathbf{q}(t), \hat{\mathbf{u}}(t) + \Delta\mathbf{u}(t)) + \Delta\mathbf{n}(t) \\ &\approx f(\hat{\mathbf{q}}(t), \hat{\mathbf{u}}(t)) + \mathbf{J}_{f,\mathbf{q}}(t)\Delta\mathbf{q} + \mathbf{J}_{f,\mathbf{u}}(t)\Delta\mathbf{u} + \Delta\mathbf{n}(t)\end{aligned}\quad (4.4)$$

onde $\mathbf{J}_{f,\mathbf{q}}$ e $\mathbf{J}_{f,\mathbf{u}}$ são os jacobianos de f com relação a \mathbf{q} e \mathbf{u} respectivamente, definidos como:

$$\begin{aligned}\mathbf{J}_{f,\mathbf{q}}(t) &= \left. \frac{\partial f(\mathbf{q}, \mathbf{u})}{\partial \mathbf{q}} \right|_{\hat{\mathbf{q}}(t), \hat{\mathbf{u}}(t)} \\ \mathbf{J}_{f,\mathbf{u}}(t) &= \left. \frac{\partial f(\mathbf{q}, \mathbf{u})}{\partial \mathbf{u}} \right|_{\hat{\mathbf{q}}(t), \hat{\mathbf{u}}(t)}\end{aligned}$$

No caso de um robô móvel com acionamento diferencial esses dois jacobianos são calculados a partir da equação 4.2, resultando em:

$$\mathbf{J}_{f,\mathbf{q}}(t) = \begin{pmatrix} 1 & 0 & -\Delta\hat{l}(t) \sin \hat{\theta}(t) \\ 0 & 1 & \Delta\hat{l}(t) \cos \hat{\theta}(t) \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{J}_{f,\mathbf{u}}(t) = \begin{pmatrix} \cos \hat{\theta}(t) & 0 \\ \sin \hat{\theta}(t) & 0 \\ 0 & 1 \end{pmatrix}$$

Utilizando a aproximação 4.4 podemos obter a expressão para o cálculo de $\Delta\mathbf{q}$, da seguinte forma:

$$\begin{aligned} \mathbf{q}(t + \Delta t) &\approx f(\hat{\mathbf{q}}(t), \hat{\mathbf{u}}(t)) + \mathbf{J}_{f,\mathbf{q}}(t)\Delta\mathbf{q} + \mathbf{J}_{f,\mathbf{u}}(t)\Delta\mathbf{u} + \Delta\mathbf{n}(t) \\ &= \hat{\mathbf{q}}(t + \Delta t) + \Delta\mathbf{q}(t + \Delta t) \\ \Delta\mathbf{q}(t + \Delta t) &= \mathbf{J}_{f,\mathbf{q}}(t)\Delta\mathbf{q}(t) + \mathbf{J}_{f,\mathbf{u}}(t)\Delta\mathbf{u}(t) + \Delta\mathbf{n}(t) \end{aligned} \quad (4.5)$$

Podemos definir a matriz $\Sigma_{\mathbf{q}}$ como sendo a matriz de covariância do erro no cálculo de \mathbf{q} , ou seja,

$$\begin{aligned} \Sigma_{\mathbf{q}}(t) &= E(\Delta\mathbf{q}(t), \Delta\mathbf{q}(t)^T) \\ &= \begin{pmatrix} \sigma_x^2(t) & \sigma_{xy}(t) & \sigma_{x\theta}(t) \\ \sigma_{xy}(t) & \sigma_y^2(t) & \sigma_{y\theta}(t) \\ \sigma_{x\theta}(t) & \sigma_{y\theta}(t) & \sigma_\theta^2(t) \end{pmatrix} \end{aligned}$$

onde E representa a operador estatístico de média.

Essa matriz descreve a incerteza associada ao cálculo de $\hat{\mathbf{q}}$. Utilizando a equação 4.5 e sabendo que $\Delta \mathbf{q}$, $\Delta \mathbf{u}$ e $\Delta \mathbf{n}$ não são correlacionados, podemos obter a expressão de $\Sigma_{\mathbf{q}}$:

$$\Sigma_{\mathbf{q}}(t + \Delta t) = \mathbf{J}_{f,\mathbf{q}}(t)\Sigma_{\mathbf{q}}(t)\mathbf{J}_{f,\mathbf{q}}(t)^T + \mathbf{J}_{f,\mathbf{u}}(t)\Sigma_{\mathbf{u}}(t)\mathbf{J}_{f,\mathbf{u}}(t)^T + \mathbf{Q}(t) \quad (4.6)$$

onde $\Sigma_{\mathbf{u}}$ é a matriz de covariância do erro na medição do deslocamento do robô, definida como:

$$\Sigma_{\mathbf{u}}(t) = E(\Delta \mathbf{u}(t)\Delta \mathbf{u}(t)^T)$$

O cálculo da matriz de covariância $\Sigma_{\mathbf{q}}$, utilizando a equação 4.6, necessita do conhecimento das matrizes de covariância $\Sigma_{\mathbf{u}}$ e \mathbf{Q} . Para a determinação de $\Sigma_{\mathbf{u}}$, utilizaremos a seguinte nomenclatura:

- \hat{b} é o comprimento medido do eixo do robô;
- \hat{r}_e e \hat{r}_d são os raios medidos das rodas esquerda e direita, respectivamente;
- $\Delta \hat{l}_e$ e $\Delta \hat{l}_d$ são os deslocamentos lineares estimados das rodas esquerda e direita, respectivamente;
- $\Delta \hat{l}$ é o deslocamento linear estimado do robô;
- $\Delta \hat{\theta}$ é o deslocamento angular estimado do robô.

De acordo com o descrito no capítulo 2, os deslocamentos linear e angular do robô são:

$$\Delta l(t) = \frac{r_d \Delta l_d(t) + r_e \Delta l_e(t)}{2} \quad (4.7)$$

$$\Delta \theta(t) = \frac{r_d \Delta l_d(t) - r_e \Delta l_e(t)}{b} \quad (4.8)$$

Definindo agora $\hat{\mathbf{m}}$ como o vetor dos parâmetros medidos e $\Delta \mathbf{m}$ como o erro na medição de \mathbf{m} , temos:

$$\begin{aligned} \hat{\mathbf{m}} &= \begin{bmatrix} \hat{r}_e & \hat{r}_d & \hat{b} \end{bmatrix}^T \\ \Delta \mathbf{m} &= \begin{bmatrix} \Delta r_e & \Delta r_d & \Delta b \end{bmatrix}^T \end{aligned}$$

A matriz de covariância do erro de velocidade é calculada como:

$$\Sigma_{\mathbf{u}}(t) = \mathbf{J}_{\mathbf{u},\mathbf{m}}(t) \Sigma_{\mathbf{m}} \mathbf{J}_{\mathbf{u},\mathbf{m}}(t)^T$$

onde $\Sigma_{\mathbf{m}}$ é a matriz de covariância do erro dos parâmetros medidos (comprimento do eixo do robô e raio das rodas esquerda e direita). Esses erros não são correlacionados, o que resulta na seguinte expressão para $\Sigma_{\mathbf{m}}$:

$$\begin{aligned} \Sigma_{\mathbf{m}}(t) &= E(\Delta \mathbf{m} \Delta \mathbf{m}^T) \\ &= \begin{pmatrix} \sigma_{r_e}^2 & 0 & 0 \\ 0 & \sigma_{r_d}^2 & 0 \\ 0 & 0 & \sigma_b^2 \end{pmatrix} \end{aligned}$$

Já $\mathbf{J}_{\mathbf{u},\mathbf{m}}$ é o jacobiano de \mathbf{u} com relação a \mathbf{m} , calculado a partir das equações 4.7 e 4.8 como:

$$\begin{aligned} \mathbf{J}_{\mathbf{u},\mathbf{m}}(t) &= \left. \frac{\partial \mathbf{u}(t)}{\partial \mathbf{m}} \right|_{\hat{\mathbf{m}}} \\ &= \begin{pmatrix} \frac{\Delta \hat{l}_e(t)}{2} & \frac{\Delta \hat{l}_d(t)}{2} & 0 \\ -\frac{\Delta \hat{l}_e(t)}{b} & \frac{\Delta \hat{l}_d(t)}{b} & -\frac{\hat{r}_d \Delta \hat{l}_d(t) - \hat{r}_e \Delta \hat{l}_e(t)}{b^2} \end{pmatrix} \end{aligned}$$

Para que possamos calcular a covariância do erro de localização ($\Sigma_{\mathbf{q}}$), a partir da equação 4.6, resta apenas definirmos a matriz Q . Segundo Chenavier e Crowley (1992), Q é modelada por uma matriz diagonal definida como:

$$Q(t) = \begin{pmatrix} K_u |\Delta l \cos(\theta(t))| & 0 & 0 \\ 0 & K_u |\Delta l \sin(\theta(t))| & 0 \\ 0 & 0 & K_{l\theta} |\Delta l| + K_{\theta\theta} |\Delta \theta| \end{pmatrix}$$

onde $K_u(m^2/m)$ é o coeficiente de erro de odometria ao longo de S com respeito a Δl , $K_{l\theta}(rad^2/m)$ é o coeficiente de erro de odometria ao longo de θ com respeito a Δl e $K_{\theta\theta}(rad^2/rad)$ é o coeficiente de erro de odometria ao longo de θ com respeito a $\Delta \theta$. Esses coeficientes variam de robô para robô e são obtidos experimentalmente.

4.2.1 Região de Incerteza

Através do cálculo da matriz $\Sigma_{\mathbf{q}}$, a região de incerteza associada à posição (x, y) do robô, calculada pela odometria, é geralmente expressa através de uma região elíptica ao seu redor. Essa região indica uma área, com uma certa probabilidade, onde se encontra a posição real do robô, sendo calculada usando a seguinte expressão:

$$\begin{pmatrix} x - \hat{x} & y - \hat{y} \end{pmatrix} \begin{pmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{pmatrix}^{-1} \begin{pmatrix} x - \hat{x} \\ y - \hat{y} \end{pmatrix} \leq D^2 \quad (4.9)$$

onde D é a distância de *Mahalanobis* e indica a probabilidade de que a posição real do robô esteja dentro da área calculada.

Para exemplificar, a figura 4.2 ilustra a incerteza do erro de localização do robô após um determinado deslocamento. O robô inicia o deslocamento no estado $\mathbf{q} = (x, y, \theta)$, que é a sua localização real, e termina no estado $\hat{\mathbf{q}} = (\hat{x}, \hat{y}, \hat{\theta})$, que é a sua localização estimada pela odometria após o deslocamento. Como (\hat{x}, \hat{y}) é apenas uma estimativa da posição real, existe uma área ao seu redor com uma determinada probabilidade de conter a posição real do robô, probabilidade essa obtida de acordo com o valor de D . Essa distribuição de probabilidade elíptica é centrada na posição estimada do robô (\hat{x}, \hat{y}) e possui seus semi-eixos com comprimentos δ_x e δ_y . Desenvolvendo a equação 4.9, podemos determinar o comprimento dos eixos através das equações:

$$\delta_x^2 = \frac{D^2(\sigma_x^2\sigma_y^2 - \sigma_{xy}^2)}{\sigma_y^2 \cos^2 \hat{\theta} + \sigma_x^2 \sin^2 \hat{\theta} - 2\sigma_{xy} \sin \hat{\theta} \cos \hat{\theta}}$$

$$\delta_y^2 = \frac{D^2(\sigma_x^2\sigma_y^2 - \sigma_{xy}^2)}{\sigma_y^2 \sin^2 \hat{\theta} + \sigma_x^2 \cos^2 \hat{\theta} + 2\sigma_{xy} \sin \hat{\theta} \cos \hat{\theta}}$$

Utilizando essa mesma métrica, podemos também estimar o erro máximo cometido no cálculo da orientação do robô (δ_θ), resultando em um cone de erro de orientação cuja bissetriz é a orientação estimada ($\hat{\theta}$). O valor de $\delta\theta$ é dado por (Chenavier e Crowley, 1992):

$$\delta_\theta^2 = \sigma_\theta^2 D^2$$

Para simplificar a determinação dos limites da região elíptica, para um posterior mapeamento na imagem, aproximaremos a mesma por uma região retangular, delimitada pelo retângulo circunscrito à região elíptica. O ângulo de inclinação dessa região

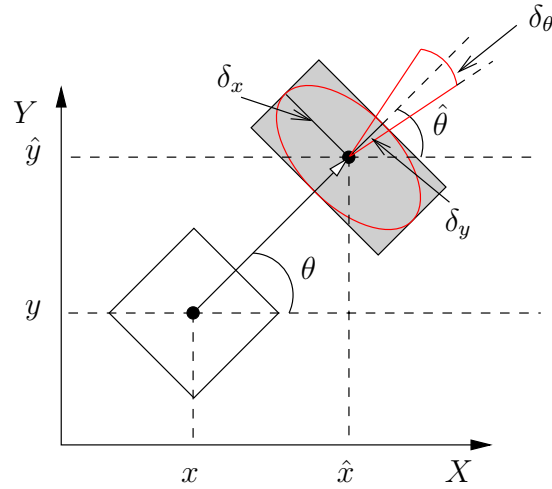


Figura 4.2: Região de incerteza ao redor da localização estimada pela odometria.

com relação ao eixo X é $\hat{\theta}$, e os seus lados tem comprimento $2\delta_x$ e $2\delta_y$. Com isso, temos uma estimativa do erro máximo cometido pela odometria, tanto em x quanto em y , obtendo um intervalo de valores para essas duas variáveis, contendo os valores reais da posição do robô, ou seja:

$$\delta_x + \hat{x} \geq x \geq \hat{x} - \delta_x$$

$$\delta_y + \hat{y} \geq y \geq \hat{y} - \delta_y$$

Da mesma forma podemos definir o intervalo de erro em θ como sendo:

$$\delta_\theta + \hat{\theta} \geq \theta \geq \hat{\theta} - \delta_\theta$$

Assim sendo, esses três valores (δ_x , δ_y e δ_θ) são calculados periodicamente, juntamente com o cálculo da odometria. Dessa forma, temos sempre à disposição a região de incerteza, que será utilizada para auxiliar na determinação da localização absoluta do robô.

4.3 Sistema de Visão

O sistema de visão utilizado é baseado na obtenção de imagens através de uma única câmera digital, posicionada sobre o robô. Além dessa câmera, também é utilizado um computador, onde são processadas as imagens. A principal função deste sistema de visão é prover um método de localização absoluta, baseado na detecção de marcos visuais, em um ambiente interno onde o robô irá navegar, de forma a corrigir periodicamente o erro de odometria. Os marcos utilizados pelo sistema são marcos naturais. Com o uso desse tipo de marco, não são necessárias alterações no ambiente onde o robô irá navegar, possibilitando ainda a adequação do sistema a diferentes ambientes, desde que estes possuam as características exigidas pelo mesmo. As características visuais do ambiente que são exploradas pelo sistema consistem em linhas retas presentes no seu piso. Essas retas, provenientes das divisórias entre as lajotas que formam o piso do ambiente, se encontram em localizações previamente conhecidas formando uma grade bi-dimensional no piso. Os marcos são constituídos pela interseção entre duas dessas retas, o que é feito com um ângulo de 90° .

O referencial adotado para a câmera $\{C\}$, possui o seu eixo Z_c coincidindo com o eixo ótico da mesma, como pode ser observado na figura 4.3. Além disso, esse referencial está posicionado a uma altura h , fixa com relação ao plano do chão e rotacionado de um ângulo α com relação a esse mesmo plano. Esses dois valores (h e α) são conhecidos, permitindo obter uma relação entre o referencial da câmera e os marcos, como será demonstrado mais adiante. A posição da câmera com relação ao robô é fixa e conhecida, permitindo assim obter diretamente a localização do robô a partir da localização da câmera.

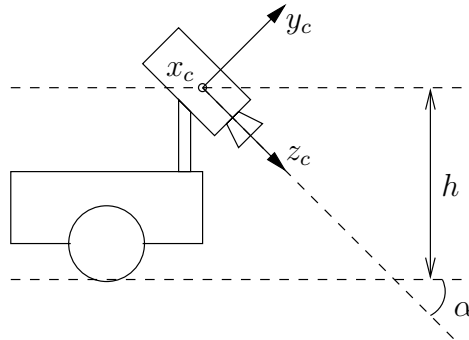


Figura 4.3: Estrutura do sistema de visão do robô.

4.3.1 Modelagem da Câmera

Nesta seção definiremos uma equação que relacione os pontos no espaço 3D com a sua respectiva projeção na imagem capturada pela câmera. Para isso, partiremos do sistema de coordenadas do ambiente e realizaremos uma série de transformações geométricas nesse sistema até chegarmos ao sistema de coordenadas da câmera, levando em consideração a movimentação e a rotação da mesma no espaço. Para determinar essa equação, vamos inicialmente definir os sistemas de coordenadas (referenciais) do ambiente $\{W\}$, do robô $\{R\}$ e da câmera $\{C\}$, conforme mostra a figura 4.4. O referencial $\{W\}$ possui o seu plano $X_w Y_w$ coincidindo com o plano do chão e o seu eixo Z_w perpendicular ao mesmo.

O referencial do robô $\{R\}$, pode ser descrito como sendo $\{W\}$ transladado para o ponto $(x, y, 0)$ e rotacionado de um ângulo θ em torno de Z_w , conforme mostra a figura 4.5. Dessa forma podemos escrever $\{W\}$ em função de $\{R\}$ como sendo:

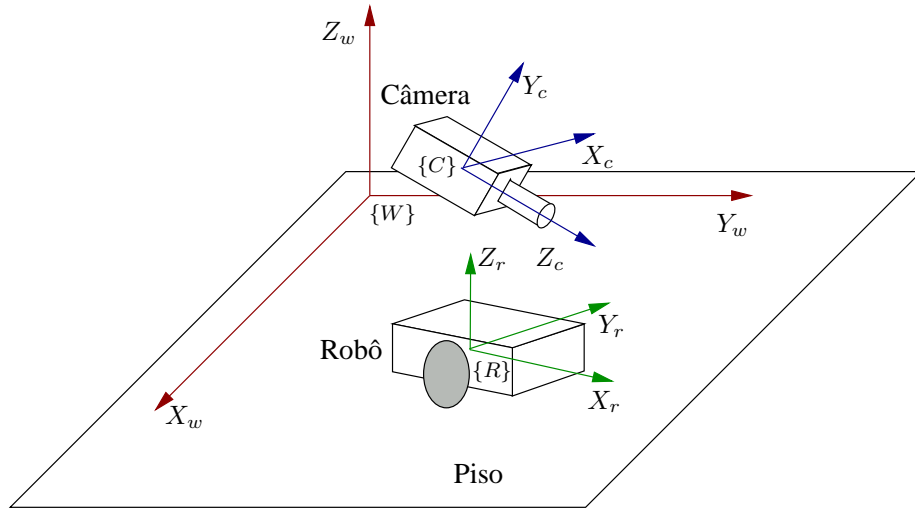


Figura 4.4: Referenciais do ambiente, do robô e da câmera.

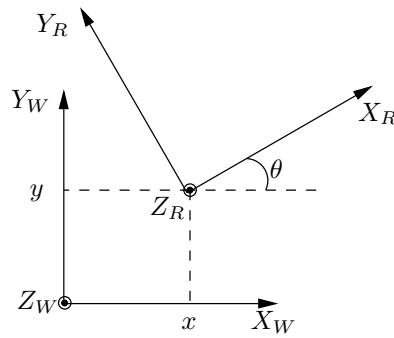


Figura 4.5: Transformação geométrica entre o referencial do ambiente e do robô.

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 & x \\ \text{sen } \theta & \cos \theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} \quad (4.10)$$

O valor (x, y, θ) corresponde à localização do robô no referencial do ambiente. Agora precisamos determinar uma relação entre $\{R\}$ e $\{C\}$. Como a câmera está posicionada a uma altura h do plano do chão, vamos transladar $\{R\}$ em Z_r para uma altura h (figura 4.6(a)), de forma que a sua origem coincida com a origem de $\{C\}$.

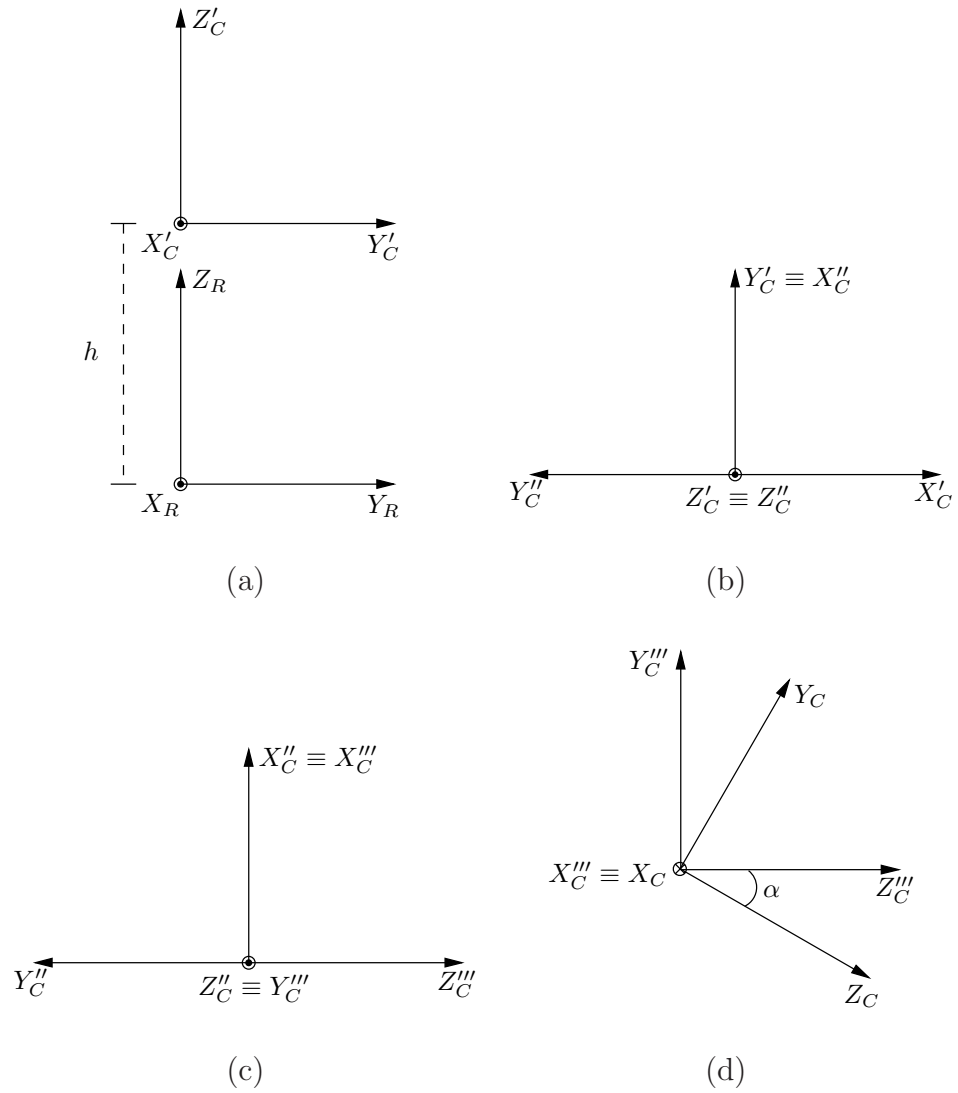


Figura 4.6: Transformações geométricas entre os referencial do robô e da câmera.

Com isso, obtemos o referencial $\{C''\}$ e a equação que o relaciona com $\{R\}$:

$$\begin{bmatrix} X_r \\ Y_r \\ Z_r \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X'_c \\ Y'_c \\ Z'_c \\ 1 \end{bmatrix} \quad (4.11)$$

É necessário agora alinhar os eixos de $\{C'\}$ com os eixos de $\{C\}$. Para isso são necessárias três rotações, inicialmente rotacionamos $\{C'\}$ de um ângulo de 90° em torno de Z'_c , resultando no referencial $\{C''\}$ e em seguida rotacionamos $\{C''\}$ de 90° em torno de X''_c , resultando no referencial $\{C'''\}$, conforme mostram as figuras 4.6(b) e 4.6(c) respectivamente. A equação que relaciona $\{C'\}$ e $\{C'''\}$ é:

$$\begin{bmatrix} X'_c \\ Y'_c \\ Z'_c \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X'''_c \\ Y'''_c \\ Z'''_c \\ 1 \end{bmatrix} \quad (4.12)$$

Finalmente podemos chegar ao referencial da câmera $\{C\}$, rotacionando $\{C'''\}$ de um ângulo α em torno de X'''_c (figura 4.6(d)), obtendo a seguinte equação:

$$\begin{bmatrix} X'''_c \\ Y'''_c \\ Z'''_c \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \text{sen } \alpha & 0 \\ 0 & -\text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Realizando o encadeamento das transformações citadas, teremos a seguinte equação matricial:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 & x \\ \text{sen } \theta & \cos \theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\text{sen } \alpha & 0 \\ 0 & \text{sen } \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

Realizando o produto entre as matrizes, obtemos:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} -\text{sen } \theta & \cos \theta \text{sen } \alpha & \cos \theta \cos \alpha & x \\ \cos \alpha & \text{sen } \theta \text{sen } \alpha & \text{sen } \theta \cos \alpha & y \\ 0 & \cos \alpha & -\text{sen } \alpha & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (4.13)$$

Para simplificar a escrita das equações, utilizamos a seguinte nomenclatura:

- s_θ : é o $\text{sen } \theta$;
- s_α : é o $\text{sen } \alpha$;
- c_θ : é o $\cos \theta$;
- c_α : é o $\cos \alpha$.

Dessa forma, podemos reescrever a equação 4.13 como sendo:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} -s_\theta & c_\theta s_\alpha & c_\theta c_\alpha & x \\ c_\alpha & s_\theta s_\alpha & s_\theta c_\alpha & y \\ 0 & c_\alpha & -s_\alpha & h \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (4.14)$$

Calculando a inversa da matriz de transformação obtemos a equação:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} -s_\theta & c_\theta & 0 & (s_\theta x - c_\theta y) \\ c_\theta s_\alpha & s_\theta s_\alpha & c_\alpha & (-c_\theta s_\alpha x - s_\theta s_\alpha y - c_\alpha h) \\ c_\theta c_\alpha & s_\theta c_\alpha & -s_\alpha & (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h) \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (4.15)$$

Essa equação representa um ponto do mundo nas coordenadas da câmera. Utilizaremos agora essa equação para a obtenção das coordenadas da imagem de um ponto no mundo 3D. Como discutido no capítulo 3, as equações de projeção na imagem de um ponto representado no referencial da câmera são:

$$x_i = \frac{X_c}{Z_c} \lambda \text{ e } y_i = \frac{Y_c}{Z_c} \lambda \quad (4.16)$$

onde λ é a distância focal da câmera, que pode ser obtida experimentalmente. Substituindo as equações 4.15 nas equações 4.16, obtemos:

$$x_i = \frac{-s_\theta X_w + c_\theta Y_w + (s_\theta x - c_\theta y)}{c_\theta c_\alpha X_w + s_\theta c_\alpha Y_w - s_\alpha Z_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

$$y_i = \frac{c_\theta s_\alpha X_w + s_\theta s_\alpha Y_w + c_\alpha Z_w + (-c_\theta s_\alpha x - s_\theta s_\alpha y - c_\alpha h)}{c_\theta c_\alpha X_w + s_\theta c_\alpha Y_w - s_\alpha Z_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

Sabendo ainda que todos os pontos de interesse na projeção da imagem, que são os marcos e a região ao seu redor, estão situados no plano do chão ($Z_w = 0$), teremos finalmente que as equações que mapeiam esses pontos na imagem são:

$$x_i = \frac{-s_\theta X_w + c_\theta Y_w + (s_\theta x - c_\theta y)}{c_\theta c_\alpha X_w + s_\theta c_\alpha Y_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

$$y_i = \frac{c_\theta s_\alpha X_w + s_\theta s_\alpha Y_w + (-c_\theta s_\alpha x - s_\theta s_\alpha y - c_\alpha h)}{c_\theta c_\alpha X_w + s_\theta c_\alpha Y_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

É através destas duas equações, que determinaremos uma região ao redor do marco procurado na imagem, conforme será visto na próxima seção, de tal forma a reduzir a quantidade de dados processados pela transformada de Hough.

4.4 Mapeamento do Erro de Odometria na Imagem

A determinação da localização absoluta do robô é feita a partir da detecção de um marco em uma imagem capturada pelo sistema de visão. A detecção do marco é obtida através da transformada de Hough para retas, que de acordo com o mostrado no capítulo anterior é um algoritmo que possui uma alta complexidade computacional. Um fator que contribui para o aumento no tempo de processamento da transformada de Hough é a grande quantidade de dados que uma imagem digital possui. Assim sendo, faz-se necessário o uso de alguma técnica para reduzir esse tempo. Uma forma de se obter tal redução é reduzir também a quantidade de dados a serem processados. Para obter tal redução, será utilizada a incerteza associada à localização do robô obtida pela odometria, calculada de acordo com a técnica apresentada na seção 4.2.

A figura 4.7 ilustra o método utilizado para reduzir a quantidade de dados a serem processados na imagem. Após se deslocar no ambiente de um ponto A para um ponto $B = (\hat{x}, \hat{y})$, o robô possui uma certa região de incerteza ao redor da sua localização atual, figura 4.7(a). Utilizando essa localização estimada e o mapa do ambiente previamente conhecido, estima-se que exista um marco centrado no ponto $C = (\hat{x}_m, \hat{y}_m)$, com relação ao referencial do ambiente. Como o ponto C é estimado usando a informação de odometria, podemos definir uma segunda região de incerteza como sendo a primeira transladada do ponto B para o ponto C . Obtemos assim uma

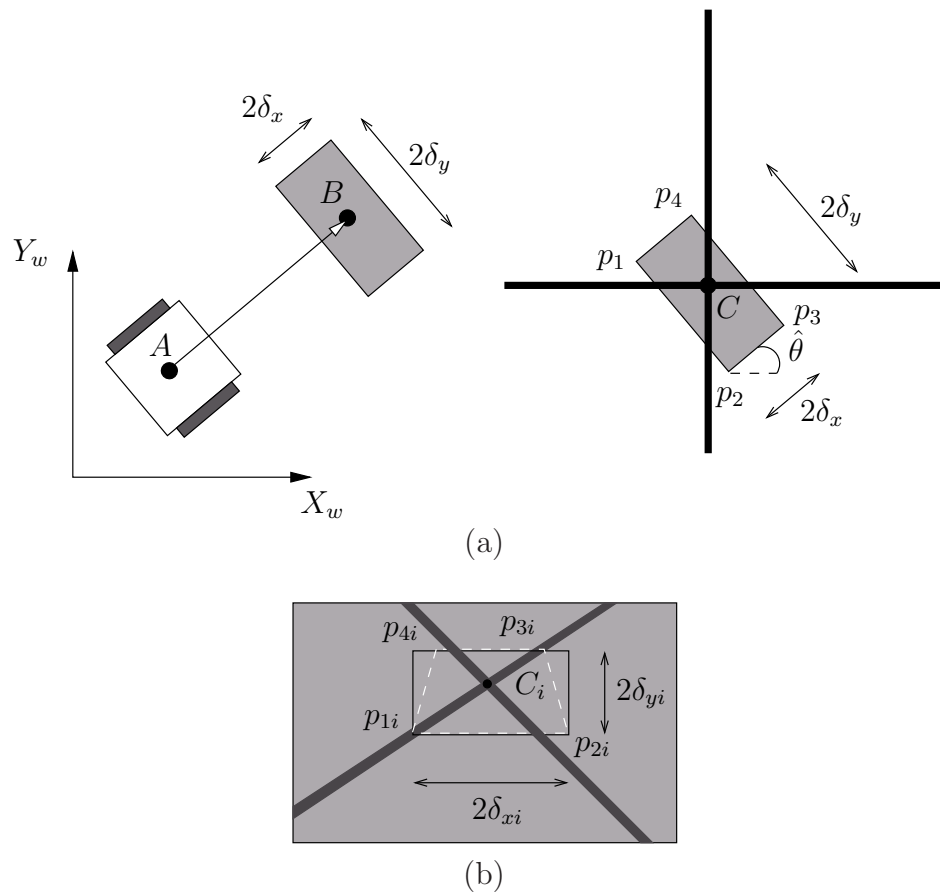


Figura 4.7: Mapeamento do erro de odometria na imagem.

região com uma certa probabilidade de conter o marco. O comprimento dos lados dessa região, que possui uma inclinação $\hat{\theta}$ com relação ao eixo X_w , são definidos como sendo $2\delta_x$ e $2\delta_y$.

Essa região em torno do ponto C possui os vértices $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$, $p_3 = (x_3, y_3)$ e $p_4 = (x_4, y_4)$, definidos através de uma translação de $\{W\}$ para o ponto C seguido de uma rotação de uma angulo $\hat{\theta}$ em torno de Z_w , como sendo:

$$\begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} c_\theta & -s_\theta & 0 & x_m \\ s_\theta & c_\theta & 0 & y_m \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -\delta_x & -\delta_x & \delta_x & \delta_x \\ \delta_y & -\delta_y & -\delta_y & \delta_y \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

A projeção desses quatro pontos gera na imagem os pontos $p_{1i} = (x_{1i}, y_{1i})$, $p_{2i} = (x_{2i}, y_{2i})$, $p_{3i} = (x_{3i}, y_{3i})$ e $p_{4i} = (x_{4i}, y_{4i})$, que definem uma região trapezoidal na mesma (linha tracejada), conforme é mostrado na figura 4.7(b). A região de interesse da imagem (linha cheia) é obtida calculando-se o retângulo circunscrito à região trapezoidal, cujas arestas são $2\delta_{xi} = x_{2i} - x_{1i}$ e $2\delta_{yi} = y_{4i} - y_{1i}$. Utilizando as equações 4.17, de projeção dos pontos do piso do ambiente na imagem, podemos calcular o comprimento dos lados dessa região de interesse como sendo:

$$2\delta_{xi} = \frac{-2\delta_y}{c_\alpha(c_\theta x_m - s_\theta y_m + x - \delta_x) + s_\alpha h} \lambda$$

$$2\delta_{yi} = \frac{2h\delta_x}{c_\alpha(c_\theta x_m - s_\theta y_m + x + \delta_x) + s_\alpha h} \lambda$$

Essa região de interesse, possui arestas $2\delta_{xi}$ e $2\delta_{yi}$, além de ser centrada no ponto $C_i = (x_{mi}, y_{mi})$, que é o ponto correspondente à projeção do ponto C na imagem.

4.5 Detecção dos Marcos

Obtida a região de interesse na imagem capturada pelo sistema de visão, realiza-se o processo de detecção dos marcos. Essa detecção é realizada utilizando as técnicas de processamento de imagens discutidas no capítulo 3. Essas técnicas são aplicadas somente à região de interesse, de forma a reduzir o tempo necessário para o processamento da imagem e detecção dos marcos. O processo de detecção é basicamente

dividido em duas etapas: a binarização da região de interesse e a aplicação da transformada de Hough a essa região binarizada.

A figura 4.8 ilustra o processo de detecção dos marcos. A partir da obtenção da região de interesse da imagem contendo o marco, é realizada uma binarização nessa região, conforme ilustra a figura 4.8(a). Essa binarização é realizada com base no cálculo do histograma da região de interesse, onde o valor limiar L utilizado é calculado como sendo:

$$L = \mu - \sigma$$

onde μ é o valor médio do histograma e σ é o seu desvio padrão.

Após isso, a transformada de Hough é aplicada na região binarizada. A transformada é calculada apenas nos *pixels* cujo nível de cinza se encontra abaixo do valor limiar (*pixels* pretos). O objetivo da aplicação da transformada é a detecção das duas retas que constituem o marco procurado, figura 4.8(b), obtendo-se os valores dos parâmetros (ρ e ϕ) de cada uma dessas retas representadas na imagem.

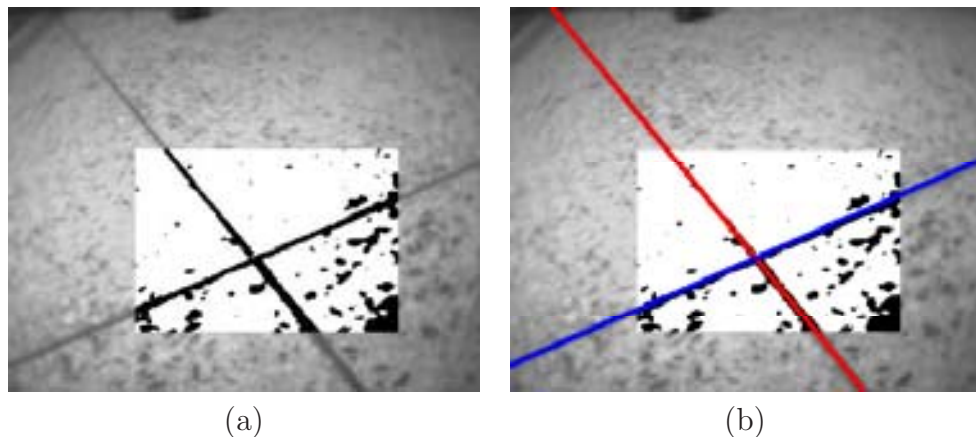


Figura 4.8: Detecção dos marcos através da transformada de Hough.

4.6 Localização Absoluta do Robô

A partir da detecção das retas que constituem o marco na imagem, realizamos o cálculo da localização absoluta do robô. Como citado anteriormente, cada marco é constituído pela interseção entre duas das retas que constituem o piso do ambiente. Essas retas estão dispostas de tal maneira a formar uma grade bi-dimensional no plano do chão ($Z_w = 0$), conforme é ilustrado na figura 4.9. Essas retas podem ainda ser divididas em dois grupos:

- Retas verticais, cujos pontos possuem coordenadas $(k_x, Y_w, 0)$;
- Retas horizontais, cujos pontos possuem coordenadas $(X_w, k_y, 0)$;

Assim sendo, os cruzamentos entre as retas ocorrem entre uma reta vertical e uma reta horizontal, nunca ocorrendo cruzamento entre duas retas do mesmo grupo. Dessa forma, cada marco é constituído por uma reta horizontal e uma reta vertical. Cada uma das retas verticais possui um valor distinto de k_x , que caracteriza a mesma. Da mesma forma, cada reta horizontal possui um valor distinto de k_y que a caracteriza. As retas de cada grupo estão ainda espaçadas por distâncias constantes, $D_x = 106.7cm$ para as retas verticais e $D_y = 96.7cm$ para as retas horizontais. Com isso, os valores de k_x e k_y podem ser escritos como sendo múltiplos inteiros de D_x e D_y , ou seja:

$$k_x = nD_x$$

$$k_y = nD_y$$

onde $n = 0, 1, 2, 3, 4, 5$.

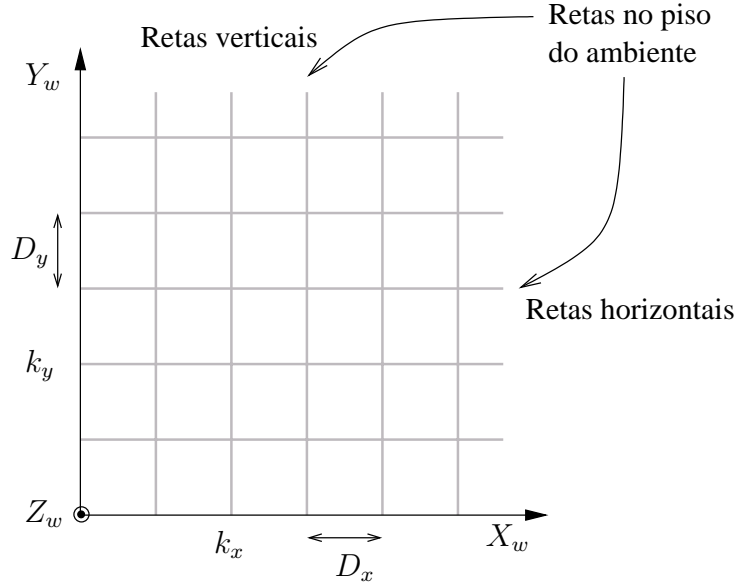


Figura 4.9: Grade do Piso

Para realizar o cálculo da localização absoluta do robô, é necessário descobrir a projeção na imagem de cada uma das retas que constituem o marco. Essas projeções são obtidas substituindo as coordenadas dos pontos de cada uma das retas, representadas no referencial do ambiente, nas equações 4.17. Dessa forma, obtemos as equações para as duas categorias de retas:

reta vertical:

$$x_i = \frac{-s_\theta k_x + c_\theta Y_w + (s_\theta x - c_\theta y)}{c_\theta c_\alpha k_x + s_\theta c_\alpha Y_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

$$y_i = \frac{c_\theta s_\alpha k_x + s_\theta s_\alpha Y_w + (-c_\theta s_\alpha x - s_\theta s_\alpha y - c_\alpha h)}{c_\theta c_\alpha k_x + s_\theta c_\alpha Y_w + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

reta horizontal:

$$x_i = \frac{-s_\theta X_w + c_\theta k_y + (s_\theta x - c_\theta y)}{c_\theta c_\alpha X_w + s_\theta c_\alpha k_y + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

$$y_i = \frac{c_\theta s_\alpha X_w + s_\theta s_\alpha k_y + (-c_\theta s_\alpha x - s_\theta s_\alpha y - c_\alpha h)}{c_\theta c_\alpha X_w + s_\theta c_\alpha k_y + (-c_\theta c_\alpha x - s_\theta c_\alpha y + s_\alpha h)} \lambda$$

Finalmente, combinando as equações de x_i e y_i , eliminamos X_w para as equações das retas horizontais e Y_w para as retas verticais, obtendo as equações que representam

cada reta na imagem:

reta vertical:

$$\lambda(k_y s_\alpha - y s_\alpha - h s_\theta c_\alpha) = x_i h c_\theta + y_i (k_y c_\alpha - y c_\alpha + h s_\theta s_\alpha)$$

reta horizontal:

$$\lambda(-k_x s_\alpha + x s_\alpha + h c_\theta c_\alpha) = x_i h s_\theta + y_i (-k_x c_\alpha + x c_\alpha - h c_\theta s_\alpha)$$

A detecção do marco pela transformada de Hough, nos fornece os parâmetros de cada uma dessas retas na imagem. Designando (ρ_v, ϕ_v) e (ρ_h, ϕ_h) , respectivamente, como sendo os parâmetros da reta vertical e horizontal detectadas na imagem, podemos montar o seguinte sistema de equações não-lineares:

$$\begin{cases} \rho_v = \lambda(k_y s_\alpha - y s_\alpha - h s_\theta c_\alpha) \\ \tan \phi_v = \frac{(k_y c_\alpha - y c_\alpha + h s_\theta s_\alpha)}{h c_\theta} \\ \rho_h = \lambda(-k_x s_\alpha + x s_\alpha + h c_\theta c_\alpha) \\ \tan \phi_h = \frac{(-k_x c_\alpha + x c_\alpha - h c_\theta s_\alpha)}{h s_\theta} \end{cases} \quad (4.17)$$

Conhecendo-se os valores de k_x , k_y , α e h , podemos obter a localização absoluta do robô solucionando este sistema de equações em função da distância focal da câmera (λ) e de (x, y, θ) , que são as variáveis que determinam a localização do robô.

4.7 Discussão

Neste capítulo, mostramos os detalhes do sistema de localização proposto neste trabalho. Além da localização relativa obtida por odometria, o sistema calcula a localização absoluta do robô através da detecção de duas retas no piso do ambiente. A detecção das retas é feita através da transformada de Hough, aplicada a uma imagem digital obtida pelo sistema de visão do robô. Para acelerar o tempo de processamento das

imagens é realizado um refinamento nas mesmas, de forma a selecionar apenas uma pequena quantidade de *pixels* a serem processados. Isso é obtido utilizando a informação de odometria e a predição do erro máximo cometido neste método, obtendo uma pequena região a ser processada na imagem. No próximo capítulo serão descritos os resultados obtidos na utilização deste sistema, aplicado a um robô móvel com rodas em um ambiente interno semi-estruturado.

Capítulo 5

Resultados Experimentais

Neste capítulo, apresentaremos os resultados obtidos com o sistema de localização proposto no capítulo anterior. Inicialmente é feita uma breve explicação sobre a construção da plataforma robótica utilizada nos experimentos, como também das ferramentas computacionais utilizadas nos mesmos. Em seguida, cada etapa do sistema é apresentada de forma separada, obedecendo a sequência de implementação. Finalmente, são descritos os resultados finais, onde todas estas etapas são unificadas, de forma a apresentar o sistema proposto completo.

5.1 Plataforma Robótica Utilizada

Para a implementação do sistema de localização proposto, foi utilizado um robô construído a partir de um *kit* da linha LEGO *MINDSTORMS* (Min, 2003). A utilização deste *kit* foi motivada pelo baixo custo do mesmo com relação às demais plataformas robóticas comerciais, além da sua disponibilidade imediata para a utilização neste trabalho. No seu projeto original, esses *kits* eram destinados ao entretenimento educativo de crianças e adolescentes. No entanto, hoje em dia eles são bastante utilizados

em ambientes de pesquisa e no ensino superior. A motivação para isso, foi o surgimento de ferramentas de *software*, como o LegOS (Leg, 2003), que permitem um melhor aproveitamento das potencialidades da unidade de controle do robô. Isso é obtido através da utilização de um *firmware* específico e da sua programação em uma linguagem de alto nível, ao contrário do proposto originalmente pelo fabricante, onde toda a programação era feita em uma linguagem visual bastante limitada.

O LegOS foi a ferramenta de *software* para a interação com o robô lego utilizada neste trabalho. Através do LegOS, o usuário pode escrever programas na linguagem de programação C e compilar esses programas para o *firmware* LegOS. Ambos, o programa do usuário e o *firmware*, são transferidos para o robô através de uma comunicação via infravermelho com um computador pessoal, para que possam ser executados. Dessa forma, foi possível realizar toda a programação do robô como também a implementação das funções de processamento de imagens, utilizando a linguagem de programação C e o sistema operacional Linux.

A construção da plataforma robótica foi realizada utilizando exclusivamente peças LEGO, como rodas, engrenagens, hastes e demais peças de encaixe, além de dois motores, dois sensores de rotação e uma unidade microcontrolada denominada RCX (*Robotic Control eXplorer*). O RCX é a unidade de controle do robô, onde são armazenados o *firmware* e os programas de usuário. O RCX possui um *display* de cristal líquido LCD (*Liquid Crystal Display*), quatro botões de controle e seis portas de E/S (Entrada/Saída). Três dessas portas são de saída, onde podem ser conectados os atuadores do robô (motores CC). Já as outras três portas são de entrada, onde podem ser conectados os sensores do robô (sensores de luz, toque e rotação). Neste trabalho, utilizamos apenas duas portas de saída e duas portas de entrada do RCX,

onde foram conectados respectivamente os dois motores e os dois sensores de rotação utilizados.

O protótipo do robô construído é mostrado na figura 5.2. Esse protótipo é baseado no modelo de plataforma com acionamento diferencial, com duas rodas tracionadas dispostas em cada lado do robô, sendo cada uma delas acionada por motores independentes e dotadas de *encoders*. Existe ainda uma terceira roda não tracionada na parte de trás, que serve apenas de apoio para o robô. A localização desse robô é obtida no ponto (x, y) de interseção entre o eixo das rodas tracionadas e o eixo de simetria, como mostrado na figura 5.1. Já o ângulo θ de orientação é o ângulo medido entre o eixo de simetria e o eixo X_w de um referencial fixo no espaço de trabalho.

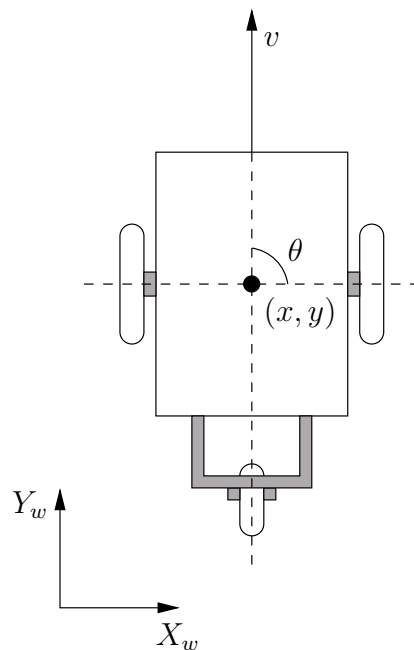


Figura 5.1: Ponto de localização do robô utilizado.

Os parâmetros utilizados no cálculo da localização relativa do robô por meio de odometria são mostrados na tabela 5.1. Os valores dos raios das rodas são valores

nominais, fornecidos pelo fabricante, enquanto que o valor do comprimento do eixo foi medido manualmente. Os sensores de rotação utilizados possuem uma resolução de 16 pulsos, ou seja, em uma volta completa do eixo desse sensor são lidos 16 pulsos. Para cada um dos sensores, foi realizado um acoplamento entre seus eixos e os eixos dos motores e das rodas correspondentes, de tal forma que o número de pulsos lidos em cada sensor de rotação em uma volta completa de uma roda do robô (N_{res}) é de 65.

Tabela 5.1: Parâmetros do modelo de odometria do robô.

b	r_d	r_e	N_{res}
13.3cm	4.08cm	4.08cm	65

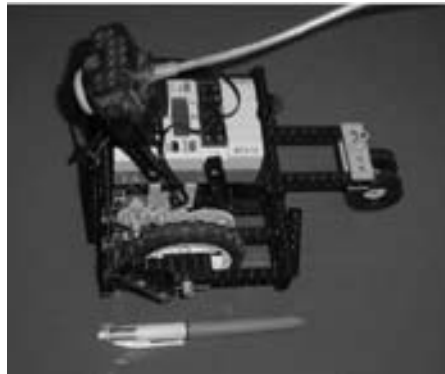


Figura 5.2: Plataforma robótica utilizada nos experimentos.

A câmera utilizada é de fabricação da Logitech para a LEGO, utilizando a mesma tecnologia de encaixe de peças, permitindo que a mesma seja acoplada ao robô com facilidade. Essa câmera é do tipo USB (*Universal Serial Bus*) e é conectada diretamente a um computador, responsável pela aquisição das imagens e pelo seu processamento. O uso desse tipo de câmera reduz em muito o custo do sistema, porém ocorre uma perda na qualidade e na taxa de captura das imagens se compararmos ao uso de uma placa de captura de vídeo.

De acordo com o proposto no capítulo 4, para realizar o processamento das imagens é necessário o conhecimento das leituras dos sensores de rotação, que provêm a localização relativa do robô. Para a obtenção desses dados pelo computador, foi utilizado, juntamente com o LegOS, um protocolo denominado LNP (LNP, 2003) (*LegOS Network Protocol*), o qual permite a comunicação entre um RCX, utilizando o *firmware* LegOS, e um computador. Essa comunicação é feita através da troca de mensagens usando a torre de transmissão, conectada ao computador. Essa torre é um dispositivo disponibilizado junto com o kit da LEGO, que possibilita a transmissão e recepção de dados, codificados em um sinal de luz infravermelho, entre o RCX e o computador. A figura 5.3, ilustra o funcionamento da comunicação entre o computador e o robô (RCX). Toda essa comunicação foi realizada a uma taxa de cerca de 100ms.

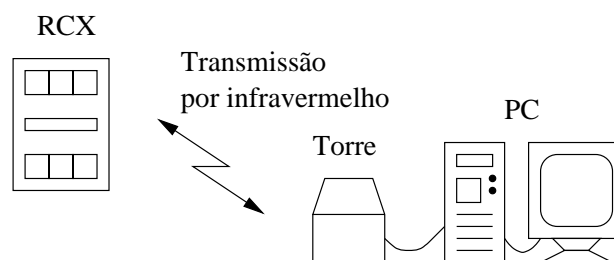


Figura 5.3: Esquema de comunicação do LNP.

5.2 Localização Padrão do Robô

Para a validação dos resultados obtidos pelo sistema de localização proposto, foi utilizado um sistema de visão global, que fornece uma localização absoluta do robô. Essa localização absoluta é assumida como sendo a localização padrão do robô e é utilizada como comparador da localização obtida pelo sistema proposto. Como não

é um dos objetivos deste trabalho o desenvolvimento de tal sistema de visão global, foi utilizado um sistema já desenvolvido e disponível para essa finalidade.

O sistema de visão global utilizado, cuja estrutura física é ilustrada na figura 5.4(a), possui uma câmera CCD (*Charged Coupled Device*) KODO KC-512NT, fixa a uma estrutura metálica e posicionada a uma altura de 250cm do piso do ambiente e com seu eixo ótico perpendicular ao mesmo. Essa câmera é conectada a um PC com uma placa de aquisição de imagens PCI/PXI 1411 da *National Instruments*, fornecendo imagens coloridas no formato RGB com 360×360 pixels a uma taxa de 30fps. A área do espaço de trabalho capturada pela imagem possui dimensão de 104×104 cm, que, desconsiderando a distorção da lente, nos fornece uma precisão de cerca de 0.28cm no cálculo da posição do robô. A localização padrão do robô é obtida com base na detecção, através do processamento das imagens, de um rótulo colocado acima do mesmo. O rótulo utilizado é composto de duas circunferências uma azul e outra vermelha. Após a detecção desse rótulo é calculada a posição (x, y) e orientação (θ) padrão do robô, com base na média dos centróides das circunferências. Os detalhes sobre como é feita a detecção dos rótulos e o cálculo dos centróides são descritos por Dias (2002). O rótulo foi colocado no robô de tal forma a coincidir com o ponto médio entre os centróides, onde é obtida a localização padrão, com o ponto de cálculo da localização pela odometria mostrado na figura 5.1.

5.3 Sistema de Odometria

O sistema de odometria foi implementado com base na obtenção das leituras dos *encoders*, acoplados às rodas do robô, através de uma comunicação via LNP. O cálculo de cada localização do robô, foi realizado a uma taxa de cerca de 100.0 milisegundos,

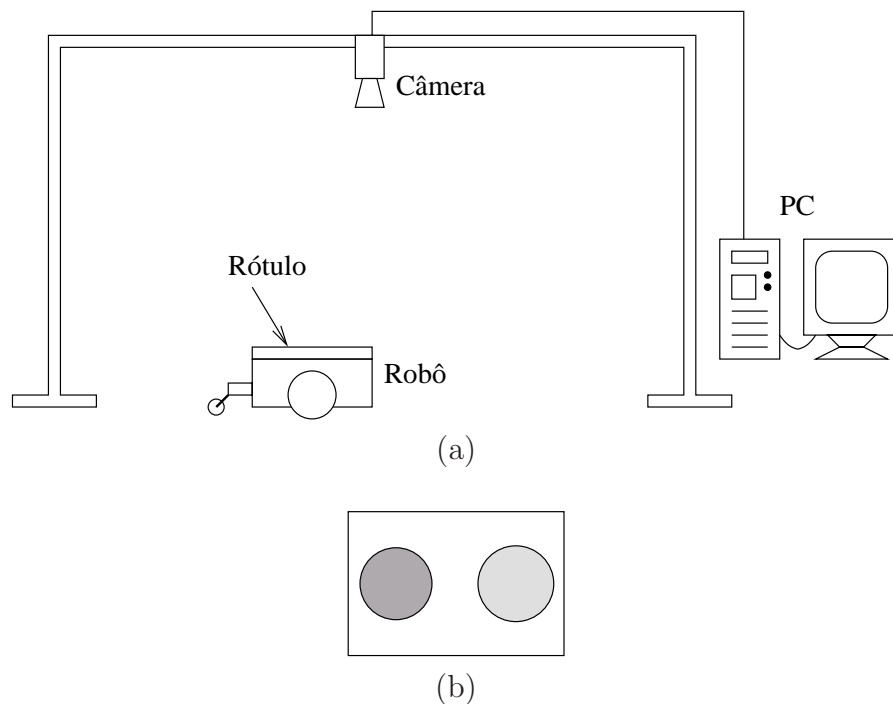


Figura 5.4: (a) Estrutura do sistema de visão global e (b) rótulo utilizado por esse sistema.

que foi o tempo necessário para a transmissão dos dados do RCX para o PC. De acordo com o mencionado anteriormente, os sensores de rotação foram arranjados de tal forma a proporcionar uma resolução de 65 pulsos por rotação. Usando essa resolução e rodas com raio de 4.08cm , obtemos uma precisão de cerca de 0.39cm , que é o menor deslocamento mensurável pelos *encoders* (distância percorrida entre dois pulsos do *encoder*).

Inicialmente foram realizados alguns testes utilizando somente o sistema de odometria. Estes testes, tinham como principal objetivo mostrar a precisão do método de odometria para pequenas trajetórias e a sua inadequação para distâncias maiores. O primeiro teste consistiu na execução de um caminho retilíneo de cerca de 0.5m , o qual foi repetido por dez vezes. A figura 5.5 mostra os caminhos real e calculado pela

odometria para um dos experimentos executados, como também a evolução temporal das variáveis x , y e θ do robô durante este mesmo caminho.

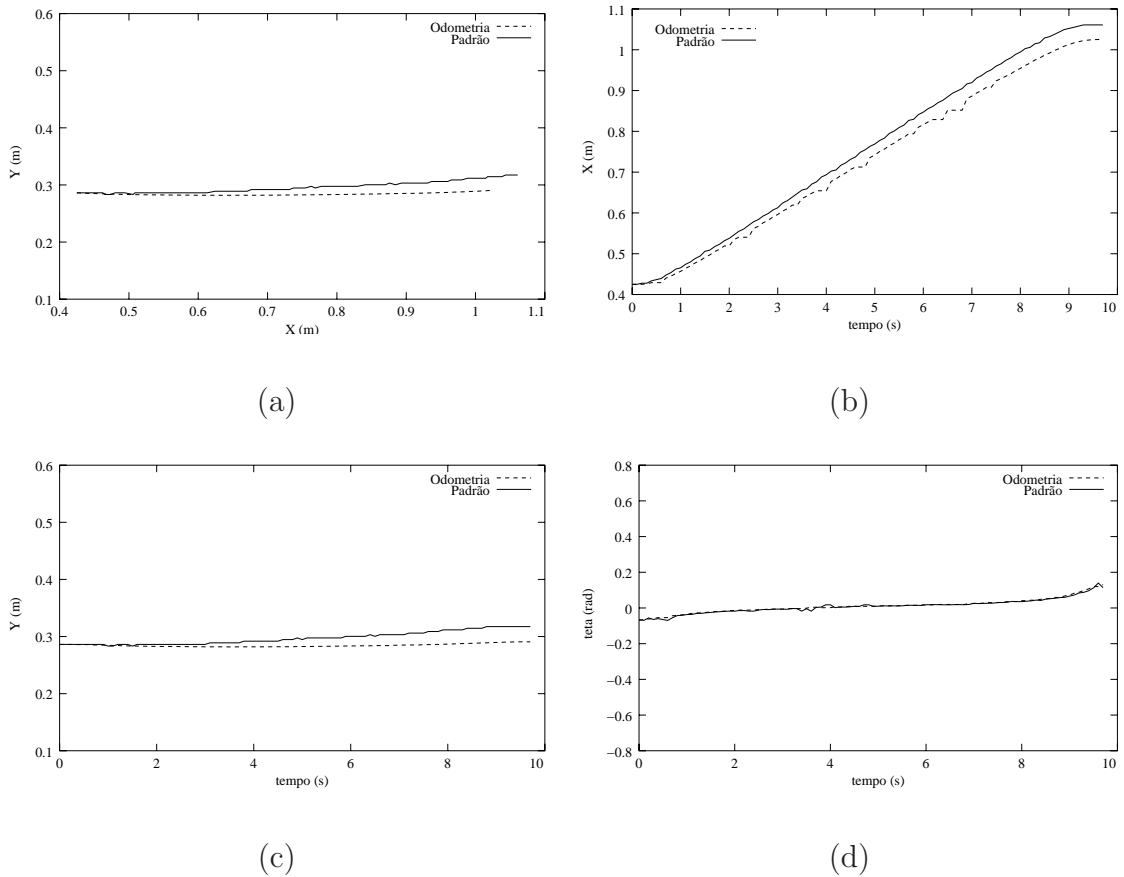


Figura 5.5: Comparação entre a localização calculada pela odometria (linha pontilhada) e a localização padrão do robô (linha cheia): (a) caminho retilíneo percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.

A tabela 5.2 mostra os erros calculados para x , y e θ , simbolizados por ϵ_x , ϵ_y e ϵ_θ respectivamente, para todos os testes. Esses erros foram calculados com base apenas na localização final do robô, através das equações:

$$\epsilon_x = x - \hat{x}$$

$$\epsilon_y = y - \hat{y}$$

$$\epsilon_\theta = \theta - \hat{\theta}$$

onde (x, y, θ) é a localização final padrão do robô obtida pelo sistema de visão global e $(\hat{x}, \hat{y}, \hat{\theta})$ é a localização final relativa, calculada pela odometria. Como pode ser observado, o erro máximo obtido em x foi de $0.038m$, com média de $0.035m$ e desvio padrão de $0.002m$. Com relação a y o erro máximo cometido foi de $0.018m$, com média $0.016m$ e desvio padrão de $0.0022m$. Já o erro máximo cometido no cálculo da orientação do robô foi de 0.0158 radianos, com média 0.0151 radianos e desvio padrão de 0.0008 radianos.

Tabela 5.2: Erros de odometria obtidos para um caminho retilíneo.

Teste	$x(m)$	$\hat{x}(m)$	$\epsilon_x(m)$	$y(m)$	$\hat{y}(m)$	$\epsilon_y(m)$	$\theta(rad)$	$\hat{\theta}(rad)$	$\epsilon_\theta(rad)$
1	1.066	1.031	0.035	0.300	0.285	0.015	0.0523	0.0367	0.0156
2	0.971	0.939	0.031	0.286	0.269	0.017	0.0523	0.0379	0.0144
3	1.060	1.022	0.038	0.317	0.301	0.016	0.1396	0.1257	0.0139
4	1.078	1.041	0.036	0.297	0.279	0.018	0.0872	0.0730	0.0142
5	1.066	1.030	0.035	0.311	0.294	0.017	0.4537	0.4387	0.0150
6	1.070	1.036	0.034	0.320	0.308	0.012	0.1260	0.1102	0.0158
7	1.065	1.032	0.032	0.304	0.286	0.018	0.1980	0.1825	0.0155
8	0.999	0.964	0.035	0.308	0.294	0.014	0.2082	0.1920	0.0162
9	1.012	0.974	0.037	0.315	0.303	0.012	0.1644	0.1496	0.0148
10	1.025	0.986	0.038	0.314	0.298	0.016	0.0564	0.0406	0.0158
Média	—	—	0.035	—	—	0.016	—	—	0.0151
Desvio padrão	—	—	0.0023	—	—	0.0022	—	—	0.0008

Como pode ser observado, o erro obtido para y e θ foi muito pequeno, o que já era esperado, pois o robô praticamente não alterou a sua orientação e não se deslocou

muito no eixo Y . Já com relação ao erro em x , observa-se que o erro cresce com o passar do tempo, de forma integrativa, no entanto sem atingir um valor muito alto. Porém, estes resultados não retratam adequadamente o fenômeno da integração dos erros de odometria, já que a distância percorrida pelo robô foi pequena. Além disso, o robô não alterou significativamente a sua orientação, impossibilitando uma análise mais precisa desse tipo de erro.

Dessa forma, tornou-se necessário a realização de uma nova série de testes, nos quais o robô percorreu uma distância maior e mudou significativamente a sua orientação. Estes novos testes foram realizados, usando uma seqüência de três caminhos retilíneos, cada um com cerca de $0.5m$ como o anterior, intercalados por duas mudanças de orientação de cerca de 90° no sentido anti-horário, gerando um caminho em forma de “U”. Da mesma forma que nos testes anteriores, esse caminho foi executado dez vezes e seus erros finais foram calculados, como mostrado na tabela 5.3. A figura 5.6, mostra a comparação entre a localização padrão e a obtida pela odometria para um desses testes.

Neste segundo experimento, a partir da análise dos resultados obtidos, observa-se uma disparidade maior entre os valores padrão e calculados. No caso do erro em x , o valor máximo obtido foi de $0.036m$, apresentando um valor médio de $0.032m$ e desvio padrão de $0.0024m$. Observa-se ainda que após a segunda rotação do robô esse erro diminuiu. Isso ocorre devido aos erros de odometria crescerem em uma única direção. No terceiro trajeto retilíneo o robô executou o caminho em x inverso ao executado no primeiro trajeto. Dessa forma, o erro foi sendo acumulado no terceiro trajeto com o sinal inverso ao acumulado no primeiro trajeto, de forma a se anularem. Já o erro em y tornou-se mais evidente neste experimento do que no experimento anterior,

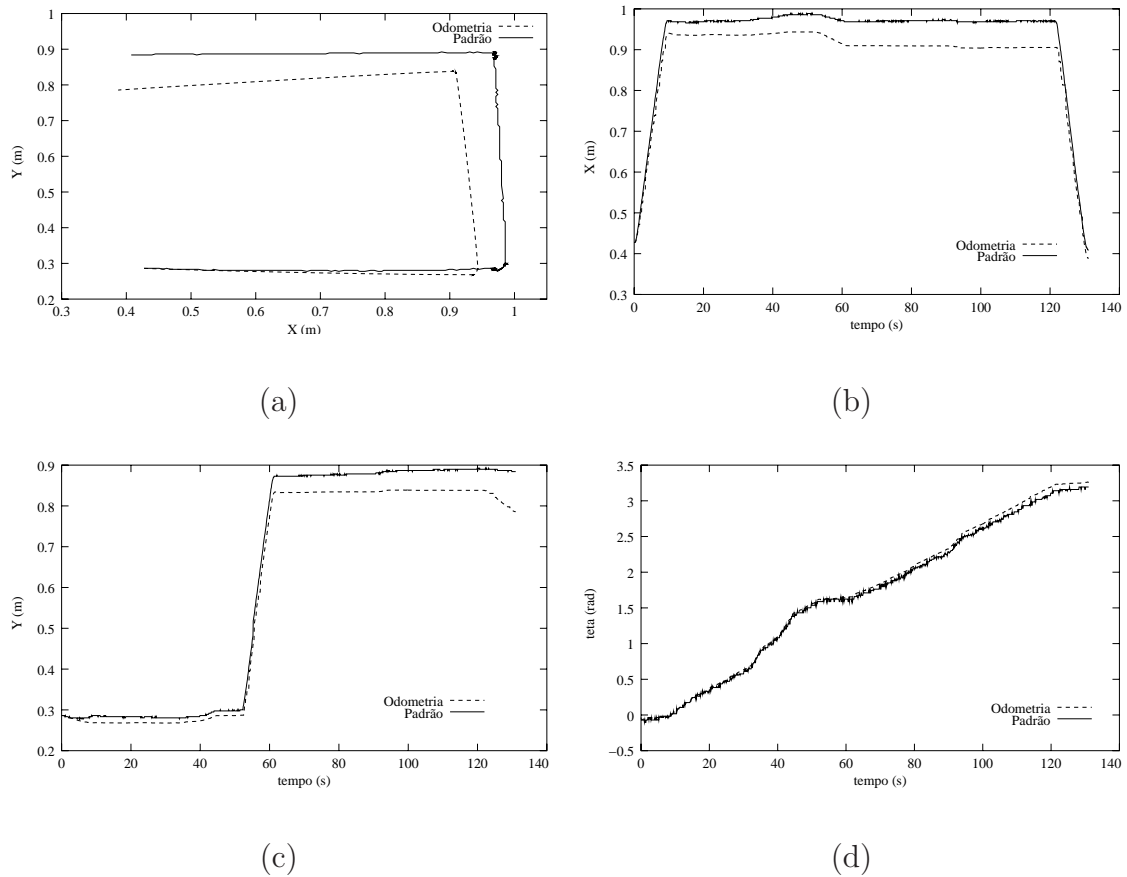


Figura 5.6: Comparação entre a localização calculada pela odometria (linha pontilhada) e a localização padrão do robô (linha cheia): (a) caminho em forma de “U” percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.

apresentando um valor máximo de $0.113m$, média de $0.108m$ e desvio padrão de $0.053m$. Da mesma forma podemos citar o erro em θ , cujo módulo do seu valor máximo foi de 0.1602 radianos. Além disso, a média desse erro foi de -0.1443 radianos e desvio padrão de 0.0104 radianos. Com isso, torna-se evidente a inviabilidade de se utilizar isoladamente a odometria, que fornece uma localização imprecisa, e a necessidade de um método de localização absoluta para corrigir esse erro acumulado.

Tabela 5.3: Erros de odometria obtidos para um caminho em forma de “U”.

Teste	$x(m)$	$\hat{x}(m)$	$\epsilon_x(m)$	$y(m)$	$\hat{y}(m)$	$\epsilon_y(m)$	$\theta(rad)$	$\hat{\theta}(rad)$	$\epsilon_\theta(rad)$
1	0.407	0.377	0.030	0.884	0.784	0.100	3.1939	3.3360	-0.1421
2	0.531	0.498	0.033	0.901	0.791	0.110	3.0543	3.1953	-0.1410
3	0.465	0.435	0.030	0.875	0.765	0.110	3.1590	3.2965	-0.1375
4	0.488	0.457	0.031	0.847	0.738	0.109	3.1939	3.3261	-0.1322
5	0.543	0.509	0.034	0.867	0.754	0.113	3.0717	3.2319	-0.1602
6	0.536	0.506	0.030	0.843	0.731	0.112	3.1273	3.2591	-0.1318
7	0.513	0.479	0.034	0.861	0.751	0.110	3.0345	3.1909	-0.1564
8	0.488	0.458	0.030	0.904	0.799	0.105	3.0654	3.2237	-0.1583
9	0.502	0.466	0.036	0.888	0.775	0.113	3.1595	3.3003	-0.1408
10	0.489	0.457	0.032	0.882	0.784	0.098	3.1902	3.3332	-0.1430
Média	—	—	0.032	—	—	0.108	—	—	-0.1443
Desvio padrão	—	—	0.0024	—	—	0.053	—	—	0.0104

5.4 Cálculo da Estimativa do Erro

Para o cálculo da estimativa do erro máximo de odometria, foi necessário uma calibração prévia, a fim de se obter os parâmetros (K_{ll} , $K_{l\theta}$, $K_{\theta\theta}$) utilizados no cálculo da matriz de covariância do erro (\mathbf{Q}). Essa calibração foi realizada através do método UMBmark (*University of Michigan Benchmark*) (Borenstein e Feng, 1994; Borenstein e Feng, 1996). Esse método, bastante utilizado para a medição de erros sistemáticos e não-sistemáticos em um robô móvel, consiste em submeter o robô a um caminho quadrangular em ambos os sentidos (horário e anti-horário) repetidas vezes, como mostrado na figura 5.7. Para simular os erros não-sistemáticos é utilizada uma série de pequenos obstáculos no caminho, de forma a gerar irregularidades no terreno. A localização final do robô calculada por odometria é comparada com uma localização absoluta, assumida como padrão.

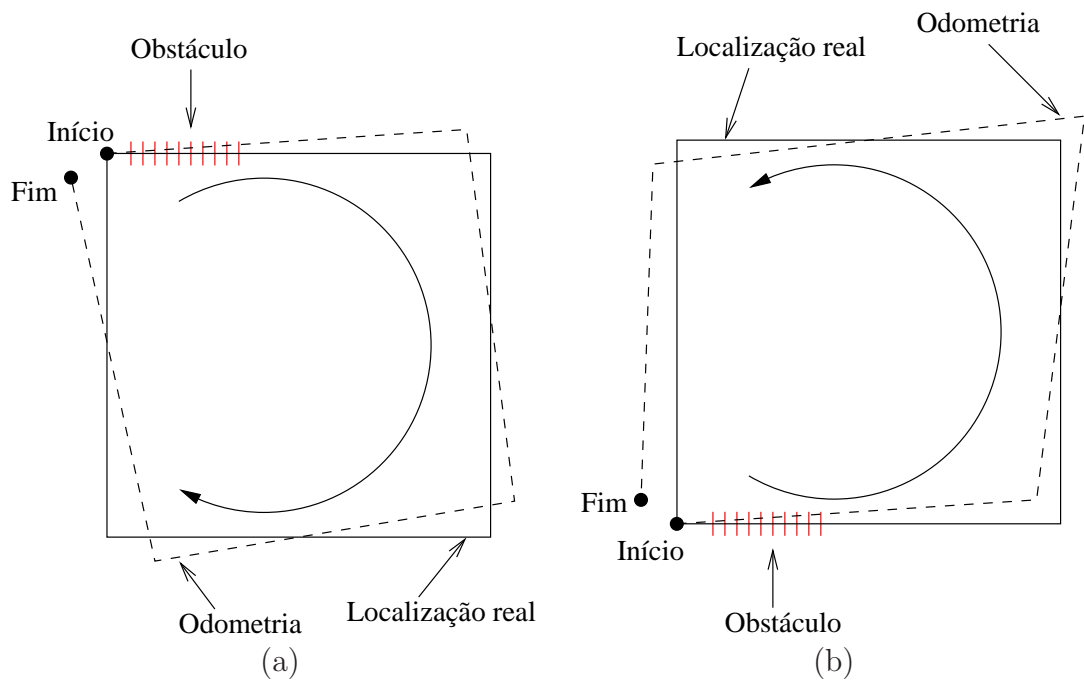


Figura 5.7: Método UMBmark: (a) sentido horário e (b) sentido anti-horário.

Utilizamos um caminho quadrangular com cerca de $0.5m \times 0.5m$, o qual foi executado dez vezes pelo robô (cinco em cada sentido). Feito isto, calculamos os erros da localização final (ϵ_x , ϵ_y e ϵ_θ) para cada um dos experimentos. A partir dos valores obtidos de ϵ_x e ϵ_y , calculamos também o erro de deslocamento, ϵ_l para cada um dos testes utilizando a equação:

$$\epsilon_l = \sqrt{\epsilon_x^2 + \epsilon_y^2}$$

Os erros calculados são mostrados na tabela 5.4, como também os deslocamentos linear e angular acumulados pelo robô em cada um dos testes, simbolizados por Δl e $\Delta\theta$ respectivamente. Os testes foram agrupados de acordo com o sentido do deslocamento (horário ou anti-horário), para que então fossem calculadas a média e a variância dos valores obtidos em cada sentido.

Tabela 5.4: Resultados do UMBmark: (a) sentido horário e (b) sentido anti-horário.

Sentido horário				
Teste	$\epsilon_l(m)$	$\epsilon_\theta(rad)$	$\Delta l(m)$	$\Delta\theta(rad)$
1	0.1250	0.0911	4.0396	27.8554
2	0.1322	0.0829	4.0273	27.0946
3	0.1263	0.1418	3.9957	26.1204
4	0.1278	0.0589	4.1018	25.9879
5	0.1302	0.1500	4.0915	27.6634
Média	0.1283	0.1049	4.0512	26.9443
Variância	8.49×10^{-6}	1.54×10^{-3}	—	—

(a)

Sentido anti-horário				
Teste	$\epsilon_l(m)$	$\epsilon_\theta(rad)$	$\Delta l(m)$	$\Delta\theta(rad)$
6	0.1264	-0.1783	4.2041	26.9673
7	0.1046	-0.1450	3.9952	25.7416
8	0.1257	-0.1940	4.2221	27.3112
9	0.1183	-0.1993	3.9846	25.4100
10	0.1298	-0.1137	4.0537	27.0577
Média	0.1210	-0.1661	4.0919	26.4975
Variância	1.012×10^{-4}	1.304×10^{-3}	—	—

(b)

A partir dos valores obtidos na tabela 5.4, calculamos os parâmetros da matriz \mathbf{Q} utilizando as equações:

$$K_{ll} = \frac{\text{variância}(\epsilon_l)}{\text{média}(\Delta l)}$$

$$K_{l\theta} = \frac{\text{variância}(\epsilon_\theta)}{\text{média}(\Delta l)}$$

$$K_{\theta\theta} = \frac{\text{variância}(\epsilon_\theta)}{\text{média}(\Delta\theta)}$$

Foram calculados dois valores para cada um desses parâmetros, um para cada

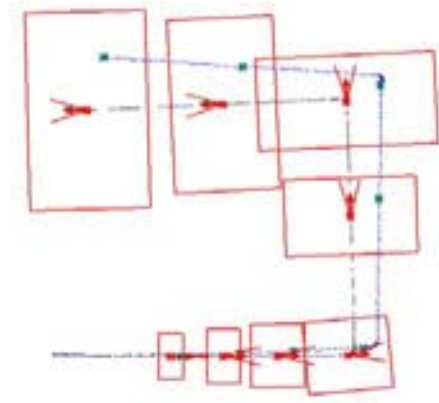
sentido, sendo que foi adotado para o restante dos experimentos o maior desses valores. Os valores da variância do erro na medição dos parâmetros do robô ($\sigma_{r_e}^2, \sigma_{r_d}^2, \sigma_b^2$), utilizados na matriz $\mathbf{J}_{u,m}$ dependem da precisão na medição desses parâmetros. Foi estimado um desvio padrão de $1.0mm$ para cada um deles. Além disso, foi utilizada no cálculo da região de incerteza uma distância de *Mahalanobis* (D) igual a 3.0. Utilizando esse valor temos uma probabilidade de 98.9% de que o robô esteja dentro da elipse. Todos os parâmetros calculados e estimados são mostrados na tabela 5.5.

Tabela 5.5: Parâmetros utilizados no cálculo da estimativa do erro.

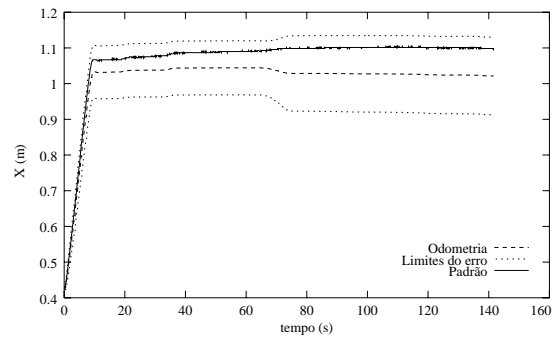
Parâmetro	Valor utilizado
K_{ll}	$2.4731 \times 10^{-5} m^2/m$
$K_{l\theta}$	$3.8013 \times 10^{-4} rad^2/m$
$K_{\theta\theta}$	$5.7154 \times 10^{-5} rad^2/rad$
σ_{r_e}	$1.0mm^2$
σ_{r_d}	$1.0mm^2$
σ_b	$1.0mm^2$
D	3.0

Para a validação do modelo preditivo apresentado, foram realizados outros dez testes. Em cada um deles, o robô foi novamente submetido ao caminho em forma de “U” e paralelamente ao cálculo da sua localização por odometria, foi calculada a região de incerteza da sua posição e o cone de incerteza da sua orientação. A figura 5.8 mostra os resultados obtidos em um dos testes realizados. Na figura 5.8(a) é ilustrado o caminho realizado pelo robô e a região de incerteza calculada ao redor da posição obtida pela odometria. Para facilitar a visualização, essa região só é ilustrada em alguns instantes, porém ela é calculada à mesma taxa de amostragem da odometria. Note que em todos os instantes apresentados, a posição real do robô está sempre contida na região de incerteza apresentada. Da mesma forma ocorre para a orientação do robô, já que a orientação real está sempre contida no cone de erro mostrado. Para

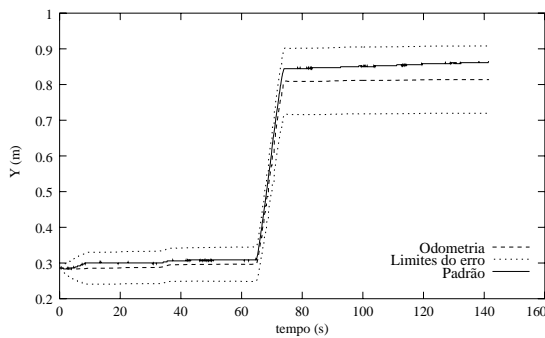
uma melhor análise dos resultados, as figuras 5.8(b), (c) e (d) ilustram a evolução temporal das variáveis de configuração (x, y, θ) ao longo do tempo, comparando-as com os limites inferior e superior do erro para cada uma delas. Podemos observar que os valores reais sempre se encontram dentro dos limites estipulados. Os resultados obtidos para os demais testes foram semelhantes ao apresentado, de forma que o modelo evolutivo utilizado foi considerado confiável aos requisitos do sistema.



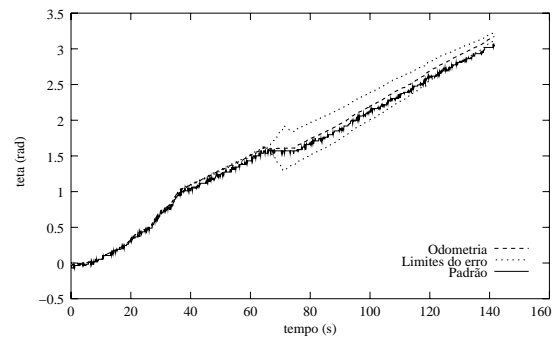
(a)



(b)



(c)



(d)

Figura 5.8: Resultados obtidos na estimação do erro de odometria. (a) caminho em forma de "U" percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.

5.5 Processamento das Imagens

As imagens são capturadas em escala de cinza, utilizando a resolução máxima da câmera, que é de 352 *pixels* de largura por 288 *pixels* de altura. Cada *pixel* é armazenado no computador em um *byte*, podendo assumir um valor entre 0 e 255, correspondente ao nível de cinza do *pixel*. A taxa média obtida de captura das imagens ficou em torno de 7fps. Uma característica da câmera utilizada é um mecanismo interno de ajuste automático de brilho, que faz com que as bordas da imagem fiquem mais escuras do que a sua parte central, dificultando ainda mais o processo de limiarização.

5.5.1 Segmentação das Imagens

O processo de segmentação é iniciado realizando-se uma varredura na região de interesse da imagem, a fim de se obter o histograma dessa região. A partir deste histograma são calculados, os valores da sua média e desvio padrão, de forma a se encontrar o valor limiar para a binarização da região. Feito isto é realizada uma nova varredura na região, onde é efetivamente realizada a sua binarização.

O objetivo dessa binarização é distinguir, aproximadamente, na imagem obtida do piso do ambiente, os marcos do restante da imagem. Como discutido anteriormente, os marcos são formados por retas provenientes das divisórias entre as lajotas que formam o piso do ambiente. Essas divisórias possuem uma tonalidade próxima ao preto, de forma que a mesma se distingue bem do piso propriamente dito, que possui uma tonalidade mais clara. No entanto, o piso possui uma textura composta por uma série de pequenos pigmentos, que muitas vezes possuem uma tonalidade próxima a das retas, se tornando um complicador para a binarização das imagens.

A distinção entre os *pixels* que formam as retas do piso, do restante dos *pixels* da imagem é obtida fazendo com que os *pixels* que estejam abaixo do valor limiar se tornem pretos e o restante dos *pixels* se tornem brancos. Dessa forma teremos regiões de *pixels* pretos, que serão processados pela transformada de Hough, nas quais estarão os *pixels* que formam as retas. Para ilustrar esse processo, considere a binarização da imagem mostrada na figura 5.9(a). Essa mesma imagem após o processo de binarização é mostrada na figura 5.9(b), que como pode ser observado possui, além das regiões onde se encontram as retas, uma série de pequenas regiões provenientes da textura do piso.

Além disso, essa imagem binarizada possui uma grande região de *pixels* pretos localizados nas bordas, o que é decorrente da não uniformidade do brilho da imagem, ajustado automaticamente pela câmera. Porém, o problema da não uniformidade do brilho das imagens é eliminado, na medida em que apenas uma pequena região da imagem é analisada. Isso é ilustrado na figura 5.9(b) e (c). Como os *pixels* analisados se encontram menos dispersos espacialmente, ocorre uma maior uniformidade do brilho nessa região, fazendo com que os *pixels* que pertencem ao chão se encontrem em níveis de cinza mais próximos, facilitando a limiarização. A utilização da região de interesse, também propicia uma redução quantitativa nos *pixels* que serão analisados pela transformada de Hough, de forma a possibilitar a redução no tempo de processamento dessa imagem.

5.5.2 Transformada de Hough

Realizada a binarização da região de interesse, iniciamos o processo de detecção dos marcos. Na aplicação da transformada de Hough a este problema, estamos interessados em calcular não apenas uma única reta e sim duas, que formam um marco

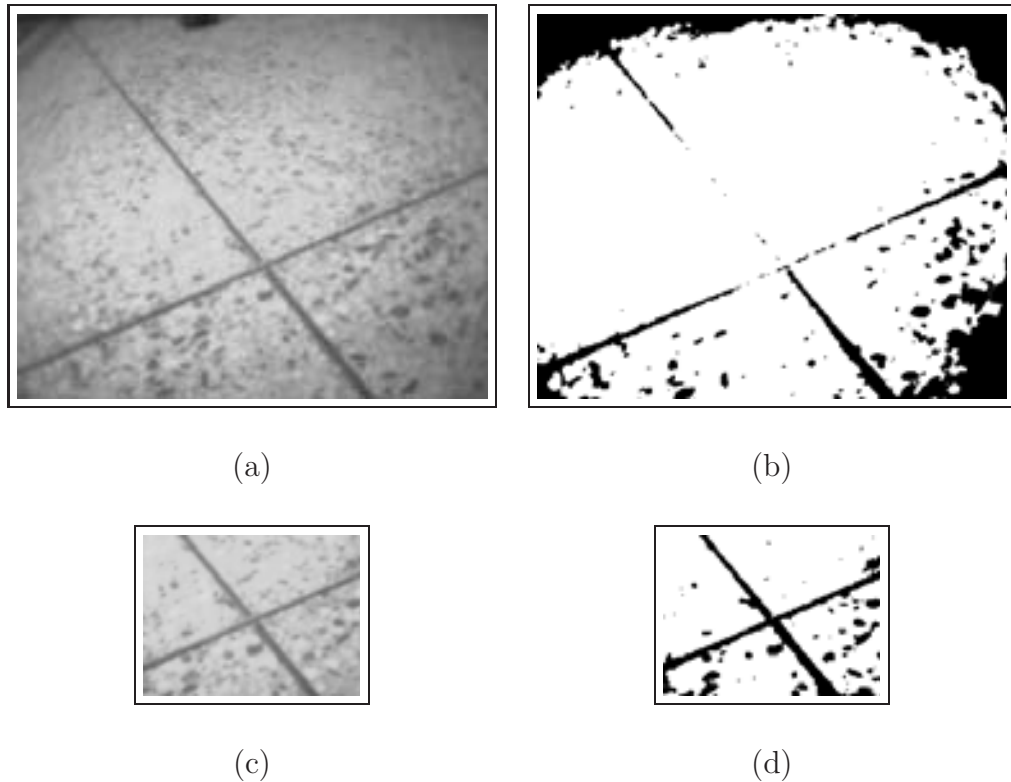


Figura 5.9: Comparação entre a segmentação de: (a) e (b) uma imagem completa e (c) e (d) apenas uma região.

analisado. Desta forma, devemos então encontrar não apenas um ponto de intercessão entre as senóides, e sim dois destes pontos. Devemos porém tomar um certo cuidado na hora de classificar tais pontos de intercessão. O ponto de intercessão com o maior número de senóides, indica a reta que melhor representa o conjunto de pontos analisados. Se determinarmos os dois melhores pontos de intercessão, poderemos estar escolhendo retas muito parecidas (próximas no espaço de parâmetros), conforme é ilustrado na figura 5.10.

As duas retas detectadas passam pelo mesmo aglomerado de *pixels* pretos, que são provenientes de uma única reta do piso. Isto ocorre, porque ambas as retas detectadas comportam mais pontos do que qualquer reta passando pelo outro aglomerado de

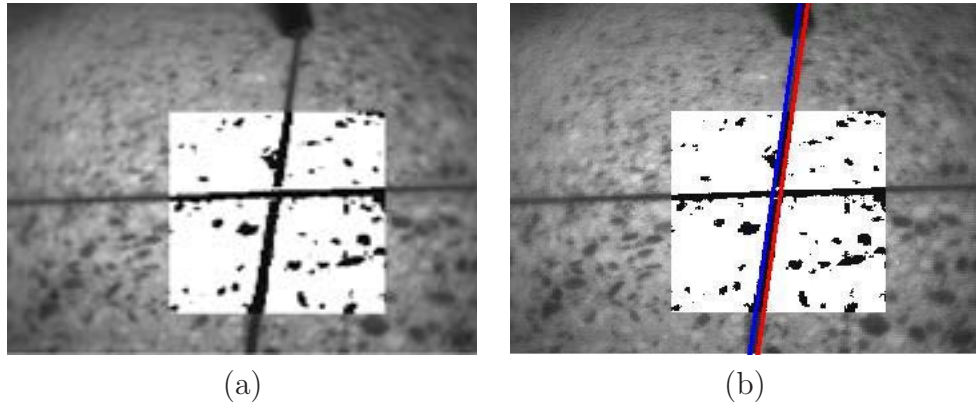


Figura 5.10: (a) Uma imagem contendo duas retas e (b) o resultado incorreto da detecção das retas.

pixels pretos, onde passa a outra reta procurada. Isto nos indica que as retas determinadas pela transformada de Hough devem ser suficientemente diferentes umas das outras para que possam representar o marco corretamente. Dessa forma, ao determinar os pontos máximos de intercessão no espaço de parâmetros, estes devem ter uma certa distância entre si, garantindo assim a distinção entre as retas determinadas.

Para solucionar tal problema, foi utilizado a estimativa do erro máximo de orientação cometido δ_θ , calculado no processo de estimação do erro de odometria. Dessa forma, a varredura do espaço de parâmetros, tanto no cálculo das retas quanto na busca dos máximos, foi realizada em apenas dois subintervalos de ϕ , conforme mostra a figura 5.11. Esses intervalos são definidos como:

$$\delta_\theta + \hat{\phi}_v \geq \hat{\phi}_v \geq \hat{\phi}_h - \delta_\theta$$

$$\delta_\theta + \hat{\phi}_h \geq \hat{\phi}_h \geq \hat{\phi}_v - \delta_\theta$$

Os valores $\hat{\phi}_v$ e $\hat{\phi}_h$, são as estimativas dos ângulos das duas retas procuradas na imagem, obtidos através das equações 4.17 que projetam as retas do piso na imagem,

utilizando a estimativa da localização atual $(\hat{x}, \hat{y}, \hat{\theta})$, ou seja,

$$\tan \hat{\phi}_v = \frac{(k_y c_\alpha - \hat{y} c_\alpha + h s_{\hat{\theta}} s_\alpha)}{h c_{\hat{\theta}}}$$

$$\tan \hat{\phi}_h = \frac{(-k_x c_\alpha + \hat{x} c_\alpha - h c_{\hat{\theta}} s_\alpha)}{h s_{\hat{\theta}}}$$

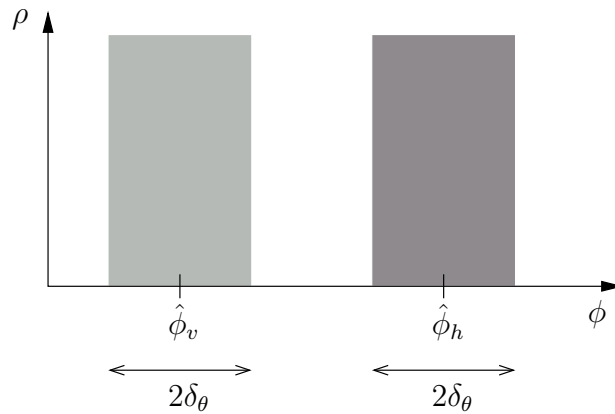


Figura 5.11: Intervalos de cálculo das retas no espaço de parâmetros.

Foi realizada uma série de experimentos, a fim de verificar a eficiência do método utilizado na redução da quantidade de dados processados e do tempo de processamento da transformada de Hough. Nesses experimentos o robô executou um certo deslocamento no ambiente, calculando a região de incerteza, e em seguida foi capturada uma imagem de um marco, realizando o mapeamento do erro estimado na região de interesse da imagem. Após isso, foi realizado todo o processamento na região de interesse para a detecção das retas, como também na imagem completa, a fim de comparação. A quantização do espaço de parâmetros utilizada foi de $\Delta\phi = 1$ e $\Delta\rho = 1$. A tabela 5.6 mostra os resultados obtidos nestes experimentos. Como pode ser observado, a redução na quantidade de *pixels* processados foi de pelo menos de 72% (obtida no teste 9), da mesma forma que também foi obtida uma redução de

pelo menos 80% no tempo de processamento da imagem (obtida no teste 2).

Tabela 5.6: Resultados obtidos com o mapeamento do erro de odometria na imagem.

Teste	distância percorrida		imagem completa		região de interesse	
	$\Delta x(m)$	$\Delta y(m)$	dimensões	tempo (ms)	dimensões	tempo (ms)
1	0.508	0.398	352×288	529.23	181×139	100.01
2	0.458	0.487	352×288	510.44	155×146	102.59
3	0.420	0.506	352×288	538.64	143×171	106.70
4	0.464	0.380	352×288	470.51	157×133	87.09
5	0.392	0.410	352×288	488.26	144×142	82.86
6	0.480	0.390	352×288	469.74	172×136	81.70
7	0.382	0.502	352×288	524.81	141×170	93.96
8	0.424	0.415	352×288	443.37	154×144	82.67
9	0.516	0.443	352×288	525.25	183×152	101.47
10	0.394	0.380	352×288	458.02	165×133	84.02

5.6 Resultados Finais

Finalmente, para a validação do sistema de localização completo, foi realizada uma série de experimentos. Inicialmente foi verificada a autonomia do sistema de localização, ou seja, a distância máxima que o robô pode percorrer sem que ocorra o risco da perda do marco na imagem. Isso pode ocorrer quando a região de interesse aumenta de tal forma a se tornar maior que a imagem capturada. O experimento consistiu na execução por parte do robô de um caminho em diagonal, de tal forma que o robô alterasse significativamente a sua posição tanto em x quanto em y . Ao percorrer esse caminho, a região de interesse da imagem, com relação ao seu ponto central, foi aumentando em ambas as direções até o momento em que essa região ultrapassou os limites da imagem, quando foi calculada a distância percorrida pelo robô até o momento, que foi de $1.2m$. Essa distância foi assumida como a distância máxima que o robô pode percorrer sem a utilização da correção absoluta.

No experimento seguinte, o robô foi posicionado em uma determinada localização, sem se movimentar, para que a sua localização global $(\tilde{x}, \tilde{y}, \tilde{\theta})$ pudesse ser calculada pelo sistema de visão. Para a realização desse experimento, o ângulo de inclinação da câmera α foi considerado fixo, como também a sua altura com relação ao chão h , cujos valores são mostrados na tabela 5.7. O cálculo da localização absoluta é realizado solucionando o sistema de equações 4.17, em função de x , y , θ e λ pelo método de Newton (Pires, 2003).

Tabela 5.7: Parâmetros utilizados pelo sistema de visão.

α	h
30°	14.5cm

Este experimento foi repetido dez vezes. Em cada uma delas o robô foi colocado em uma localização diferente. A partir da localização absoluta obtida, realizamos uma comparação com a localização padrão do robô, de forma a verificar a precisão do método. Para o erro em x , o valor máximo obtido foi de $0.011m$, com média $0.0082m$ e desvio padrão de $0.0021m$. Com relação ao erro em y , foi obtido um valor máximo de $0.010m$, média de $0.0073m$ e desvio padrão de $0.0025m$. Finalmente quanto ao erro de orientação, foi obtido um valor máximo de 0.0171 radianos, média de 0.0150 radianos e desvio padrão de 0.0011 radianos. Notadamente, se compararmos esses erros aos valores obtidos no experimento do caminho em forma de “U” utilizando somente a odometria (tabela 5.3), verificamos uma considerável redução nestes valores, mostrando a eficácia do cálculo da localização absoluta do robô, usando o sistema de visão local.

Finalmente o sistema de localização foi testado utilizando todas as suas etapas em conjunto. O robô foi novamente submetido a uma trajetória em forma de “U”. A

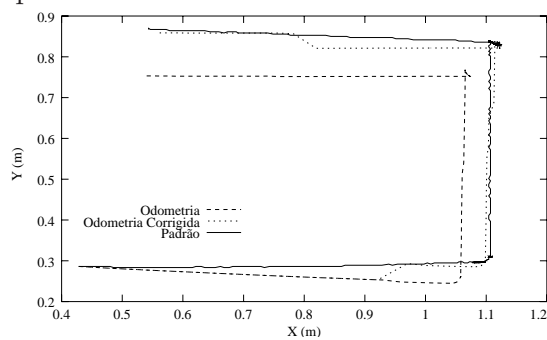
Tabela 5.8: Resultados obtidos utilizando o método absoluto.

Teste	$x(m)$	$\tilde{x}(m)$	$\epsilon_x(m)$	$y(m)$	$\tilde{y}(m)$	$\epsilon_y(m)$	$\theta(rad)$	$\tilde{\theta}(rad)$	$\epsilon_\theta(rad)$
1	0.431	0.421	0.010	0.242	0.234	0.009	0.101	0.086	0.0150
2	0.510	0.502	0.008	0.455	0.447	0.008	2.146	2.129	0.0162
3	0.313	0.302	0.011	0.397	0.388	0.009	1.514	1.499	0.0146
4	0.148	0.140	0.008	0.091	0.081	0.010	3.948	3.932	0.0158
5	0.445	0.438	0.007	0.367	0.365	0.002	2.827	2.811	0.0152
6	0.355	0.346	0.009	0.270	0.264	0.006	3.471	3.456	0.0146
7	0.480	0.470	0.010	0.302	0.293	0.009	4.502	4.488	0.0138
8	0.115	0.106	0.009	0.284	0.276	0.008	4.163	4.149	0.0134
9	0.134	0.128	0.006	0.196	0.192	0.004	5.572	5.557	0.0144
10	0.329	0.325	0.004	0.102	0.094	0.008	2.320	2.302	0.0171
Média	—	—	0.0082	—	—	0.0073	—	—	0.0150
Desvio padrão	—	—	0.0021	—	—	0.0025	—	—	0.0011

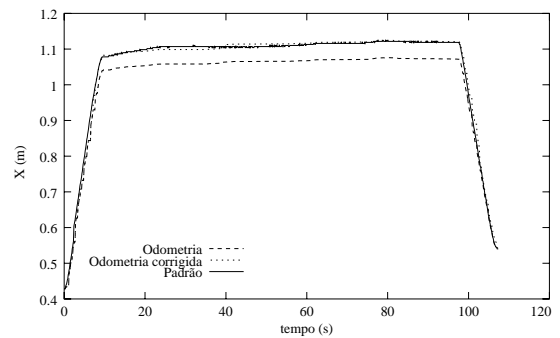
detecção dos marcos no ambiente foi realizada toda vez que o robô deslocava-se de uma distância de $0.5m$ no espaço de trabalho, reduzindo assim a possibilidade de perda do marco. Durante a execução do caminho, o sistema de visão local realizou a detecção de três marcos, reduzindo o erro de odometria nestes instantes. Essa correção foi feita apenas atualizando a localização da odometria pela localização absoluta obtida pelo sistema de visão local, no instante em que a imagem é capturada.

Como pode ser visto pela análise do gráfico, a utilização do método corretivo proporcionou uma maior confiabilidade na localização obtida pela odometria. Obviamente não é possível tornar o erro de localização completamente nulo, porém foi possível reduzi-lo significativamente. Porém, o fenômeno da integração dos erros de odometria não é definitivamente solucionado pelo método, o que faz com que sejam necessárias correções periódicas, utilizando o mesmo, à medida que o robô se desloca.

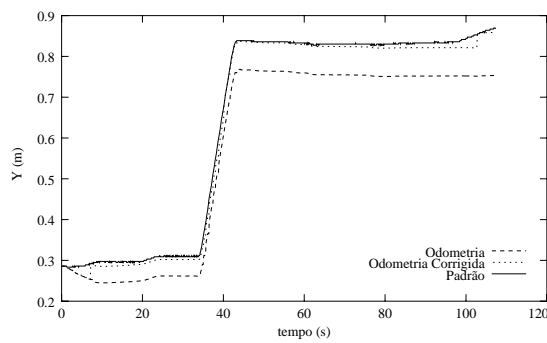
Figura 5.12: Comparação entre a localização calculada pela odometria sem correção (linha pontilhada), com correção absoluta (linha tracejada) e a localização padrão do robô (linha cheia): (a) caminho retilíneo percorrido e a evolução de (b) x , (c) y e (d) θ para esse mesmo caminho.



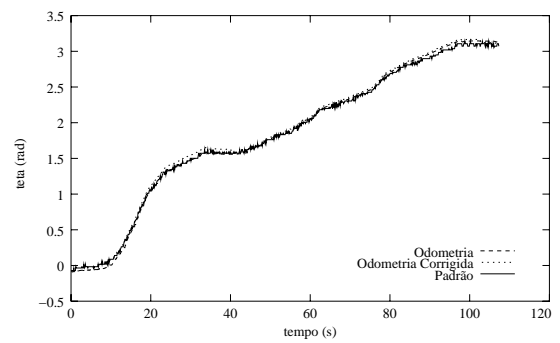
(a)



(b)



(c)



(d)

Capítulo 6

Conclusões Gerais

Este trabalho apresentou uma proposta para a determinação da localização de um robô móvel com rodas, o qual foi motivado pela importância que o processo de localização tem na navegação de um robô móvel. Durante o processo de navegação, é necessário que o robô conheça a sua posição e orientação constantemente, possibilitando dessa forma controlar a trajetória seguida pelo mesmo. Essa é uma tarefa aparentemente simples, porém possui algumas restrições temporais, já que necessita ser rápida o suficiente para atender à demanda do controlador. A proposta apresentada utiliza dois métodos de localização, o primeiro deles é a odometria, que é em um método rápido, mas que fornece uma informação que se torna cada vez mais imprecisa à medida que o robô se desloca no ambiente. O segundo método baseia-se na detecção de marcos visuais no ambiente que, apesar de fornecer uma informação mais precisa que a odometria, necessita de um tempo de processamento bem maior que o primeiro método. Um dos maiores desafios do projeto do sistema é a necessidade de que o mesmo atue em tempo real. Dessa forma, todas as tarefas executadas pelo mesmo possuem restrições temporais estritas a serem obedecidas.

O sistema de odometria implementado se mostrou capaz de fornecer a localização do robô com um certo erro, tolerável para pequenas distâncias. Para distâncias maiores, verificou-se que por diversas vezes a localização calculada pela odometria divergia em muito da localização real do robô. Esse erro ocorreu principalmente devido à imprecisão do modelo geométrico do robô, através da utilização de parâmetros no modelo de odometria diferentes dos parâmetros reais, além de folgas entre as partes mecânicas do robô e a resolução limitada dos *encoders*. Essa imprecisão da odometria já era esperada, o que motivou a utilização de um método evolutivo para estimar o erro máximo cometido no cálculo da localização do robô.

O modelo utilizado para a estimar o erro máximo cometido pela odometria se mostrou bastante eficaz. Em todos os testes realizados, o método conseguiu calcular com sucesso a região de incerteza da localização obtida pela odometria. No entanto, ainda é necessária uma análise comparativa, a partir de uma gama maior de situações, entre o método utilizado e outros métodos existentes na literatura que se propõem a modelar a propagação dos erros de odometria em diversos tipos de trajetória. Com isso, será possível evidenciar quais as vantagens e desvantagens de cada método, permitindo determinar qual deles se adequa melhor ao sistema, ou ainda, propor uma nova abordagem mais eficiente para a solução do problema.

Quanto ao método de localização absoluta utilizado, detecção de marcos naturais no ambiente, destacaram-se os aspectos inerentes ao processamento de imagens e sua adequação a um robô móvel equipado com uma câmera digital. A transformada de Hough foi a técnica utilizada para a detecção dos marcos. Essa técnica vem sendo bastante utilizada em trabalhos que envolvem processamento de imagens digitais, nas mais diversas áreas. A transformada de Hough se mostrou suficientemente capaz de

detectar as retas que formavam os marcos. Além disso, podemos destacar a robustez da transformada no que concerne ao processamento de imagens ruidosas, como as utilizadas neste trabalho.

Ainda com relação à transformada de Hough, verificou-se que se trata de um método com uma complexidade computacional alta, o que torna muitas vezes o método lento. Em particular, em robótica móvel, devido às restrições temporais impostas pelas aplicações, muitas vezes é necessário o uso de alguma técnica em conjunto com a transformada de Hough, para que se possa reduzir o tempo de processamento da mesma. Este trabalho oferece uma contribuição relevante no que diz respeito ao processamento da transformada de Hough, associando a mesma a algumas heurísticas que possibilitam a sua aplicação em tempo real. A redução do tempo de processamento da transformada foi obtida, através da redução da quantidade de dados de entrada utilizados pela mesma. Para isso, foi utilizada a informação da localização do robô, estimada pela odometria, bem como o erro máximo cometido por esse método, estimado através do modelo de evolução do erro utilizado. Os resultados obtidos utilizando essa técnica foram bastante satisfatórios, através dos quais obtemos uma redução de pelo menos 72% na quantidade de dados analisados e de pelo menos 80% no tempo necessário para a detecção das retas.

É necessário ainda um estudo mais detalhado sobre o número de níveis de quantização usados na transformada de Hough, que seja mais adequado para a aplicação, de forma a aliar precisão e velocidade no processo de detecção das retas. Uma outra possível forma de agilizar esse processamento, pode ser obtida através de uma busca aleatória dos *pixels* a serem processados pela transformada na região de interesse utilizada na imagem.

Para a implementação do sistema proposto, foram utilizados um sistema de odometria e um sistema de visão qualitativamente bastante limitados. Apesar disso, conforme mostram os resultados experimentais, o uso em conjunto desses dois sistemas, associados ao método proposto, permitiram uma localização adequada do robô a um baixo custo. Com relação à correção do erro de odometria, a mesma foi realizada de maneira bastante simples, apenas atualizando a localização atual da odometria com a localização absoluta calculada. Através de uma análise mais detalhada, talvez seja possível a obtenção de melhores resultados utilizando outro método para essa finalidade. Por exemplo, uma ponderação entre as duas localizações obtidas pelo sistema, através de um filtro de Kalman.

Finalmente, podemos concluir que o desenvolvimento bem sucedido do sistema aqui apresentado é uma prova conceitual da aplicabilidade do mesmo em robôs móveis autônomos com rodas em ambientes internos semi-estruturados. Além disso, a concepção do sistema consistiu numa oportunidade de trabalho original multidisciplinar, envolvendo as áreas de visão computacional, robótica móvel e processamento digital de imagens. Os resultados obtidos e apresentados no capítulo anterior foram satisfatórios, atendendo às expectativas iniciais do projeto. Tendo em vista esses resultados, mostrou-se que é possível desenvolver um sistema de localização eficiente para a sua utilização em tempo real. Apesar dos problemas encontrados, este trabalho trouxe inovações científicas para a solução do problema de localização de robôs móveis, as quais espera-se que sejam de grande utilidade em outros projetos de pesquisa.

Referências Bibliográficas

- Adams, M., Tschichold, N., Müller, R., Neogy, S., Ruf, L., Vestli, S. e von Flüe, D. (1994). Control and localisation of a post distributed mobile robot, *International Conference on Intelligent Robots and Systems*, pp. 150–156.
- Alsina, P. J., Gonçalves, L. M. G., Medeiros, A. A. D., Pedrosa, D. P. F. e Vieira, F. C. (2002). Navegação e controle de robôs móveis, Mini Curso do XIV Congresso Brasileiro de Automática.
- Bandeira, R. P. (2002). *Planejamento estratégico cooperativo: Aplicação a um ambiente de futebol de robôs*, Master's thesis, Universidade Federal do Rio Grande do Norte, Departamento de Engenharia Elétrica.
- Betke, M. e Gurdits, L. (1994). Mobile robot localization using landmarks, *International Conference on Intelligent Robots and Systems*, pp. 135–142.
- Borenstein, J. (1995). Internal correction of dead-reckoning errors with a dual-drive compliant linkage mobile robot, *Journal of Robotic Systems* **12**(4): 257–273.
- Borenstein, J., Everett, H. R. e Feng, L. (1996). “where am i?” sensors and methods for mobile robot positioning, University of Michigan.
- Borenstein, J., Everett, H. R., Feng, L. e Wehe, D. (1996). Mobile robot positioning: Sensors and techniques, *journal of Robotic Systems* **4**(14): 231–249.
- Borenstein, J. e Feng, L. (1994). Umbmark - a method for measuring, comparing and correcting dead-reckoning errors in mobile robots, Technical Report.

- Borenstein, J. e Feng, L. (1996). Measurement and correction of systematic odometry erros in mobile robots, *IEEE Transactions on Robotics and Automation*, Vol. 12, pp. 869–880.
- Branca, A., Stella, E. e Distante, A. (2000). Matching of line segments for indoor navigation using projective invariants, *International Conference on Intelligent Autonomous Systems*.
- Buffa, M., Faugeras, O. e Zhang, Z. (1993). A stereo vision-based navigation system for a mobile robot, Technical Report RR-1895, INRIA-Institut National de Recherche en Informatique et en Automatic.
- Chen, G.-Y. e Tsai, W.-H. (1998). An incremental-learning-by-navigation approach to vision-based autonomous land vehicle guidance in indoor environments using vertical line information and multiweighted generalized hough transform technique, *IEEE Transaction on Systems, Man, and Cybernetics*, pp. 740–748.
- Chenavier, F. e Crowley, J. L. (1992). Position estimation for a mobile robot using vision and odometry, *International Conference on Robotics and Automation*.
- Chong, K. S. e Kleeman, L. (97). Accurate odometry and error modeling for a mobile robot, *International Conference on Robotics and Automation*.
- Clerentin, A., Delahoche, L., Brassart, E. e Cauchois, C. (2002). Mobile robot localization based on multi target tracking, *International Conference on Robotics and Automation*, pp. 13–18.
- Coelho, L. S. e Campos, M. F. M. (1998). Navegação de dirigíveis autônomos baseados em visão computacional. Disponível em <http://www.dcc.ufmg.br/pos/html/spg98/anais/omni/omni.html>.
- Craig, J. J. (1986). *Introduction to Robotics: Mechanics and Control*, Addison-Wesley Publishing Company.

- Dias, A. M. (2002). *Posicionamento de um manipulador redundante utilizando realimentação visual*, Master's thesis, Universidade Federal do Rio Grande do Norte, Departamento de Engenharia Elétrica.
- Duda, R. e Hart, P. (1972). Use of the hough transformation to detect lines and curves in pictures, *Communications of the ACM* **15**(1): 11–15.
- Fairfield, N. (2002). Simple landmark localization on a three-layer mobile robot.
- Faugeras, O., Laveau, S., Robert, L., Csurka, G. e Zeller, C. (1995). 3d reconstruction of urban scenes from sequences of images, Technical Report RR-2572, INRIA-Institut National de Recherche en Informatique et en Automatique.
- Franchin, M. N. (2003a). Elementos de robótica.
URL: <http://www.dee.bauru.unesp.br/~marcelo/robotica/conteudo.html>
- Franchin, M. N. (2003b). Introdução à robótica.
URL: www.feb.unesp.br/~franchin
- Gonzalez, R. C. e Woods, R. E. (2000). *Processamento de Imagens Digitais*, Editora Edgar Blücher.
- Grisetti, G., Iocchi, L. e Nardi, D. (2000). Identificação visual para navegação de robôs móveis utilizando um controlador fuzzy sintonizado por otimização numérica, *XIII Congresso Brasileiro de Automática*.
- Grisetti, G., Iocchi, L. e Nardi, D. (2002). Global hough localization for mobile robots in polygonal environments, *International Conference on Robotics and Automation*.
- Hanahara, K., Maruyama, T. e Uchiyama, T. (1988). A real-time processor for the hough transform, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **10**: 121–125.

- Haykin, S. S. (2002). *Adaptive Filter Theory*, Prentice Hall.
- Hough, P. V. C. (1962). Methods and means for recognizing complex patterns, *U. S. Patent 3069654*.
- Howard, A. e Kitchen, L. (1997). Vision-based navigation using natural landmarks, *International Conference on Field and Service Robotics*.
- Irie, M., Nagatani, K. e Gofuku, A. (2003). Path evaluation for a mobile robot based on a risk of collision, *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 485–490.
- Jang, G., Kim, S., Lee, W. e Kweon, I. (2002). Color landmark based self-localization for indoor mobile robots, *International Conference on Robotics and Automation*, pp. 1037–1042.
- Jarvis, R. A. (1983). A perspective on range finding techniques for computer vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 122–139.
- Kelly, A. (2000). Some useful results for the propagation of error in vehicle odometry. Technical Report CMU-RU-TR-00-20.
- Kelly, A. (2001). Linearized systematic error propagation in vehicle odometry, *International Conference on Intelligent Robotics and Systems*.
- Kiryati, N., Kälviäinen, H. e Alaoutinen, S. (2000). Randomized or probabilistic hough transform: Unified performance evaluation, *Pattern Recognition*.
- Kiryati, N., Y., E. e Bruckstein, A. M. (1991). A probabilistic hough transform, *Pattern Recognition* **24**: 303–316.
- Komoriya, K. e Oyama, E. (1994). Position estimation of a mobile robot using optical fiber gyroscope, *International Conference on Intelligent Robots and Systems*, pp. 143–149.

- Lages, W. F. (1998). *Controle e estimação de Posição e Orientação de Robôs Móveis*, PhD thesis, Instituto Tecnológico de Aeronáutica.
- Latombe, J.-C. (1998). *Robot Motion Planning*, Kluwer Academic Publishers.
- Launay, F., Ohya, A. e Yuta, S. (2001). Autonomous indoor mobile robot navigation by detecting fluorescent tubes, *International Conference on Robotics and Automation*.
- Leg (2003). LegOS - a open source operating system and provides a c and c++ programming environment for the lego mindstorms robotics kits. Disponível em <http://legos.sourceforge.net>.
- Little, J. J., Lu, J. e Murray, D. R. (1998). Selecting stable image features for robot localization using stereo, *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 1037–1042.
- LNP (2003). LegOS Network Protocol. Disponível em <http://legos.sourceforge.net/HOWTO/x405.html>.
- Lora, F. A. S., Hemerly, E. M. e Lages, W. F. (1997). Estimação em tempo real de posição e orientação de robôs móveis utilizando sensores com diferentes taxas de amostragem, *Anais do 3º Simpósio Brasileiro de Automação Inteligente*, pp. 453–458.
- Lora, F. A. S., Hemerly, E. M. e Lages, W. F. (1998). Sistema para navegação e guiagem de robôs móveis autônomos, *SBA Controle & Automação*, Vol. 9, pp. 107–118.
- Mächler, P. (1998). *Robot Positioning by Supervised and Unsupervised Odometry Correction*, PhD thesis, École Polytechnique Fédérale de Lausanne, Département D'Informatique.

- Medeiros, A. A. D. (1998a). Introdução à robótica, *XVII Encontro Nacional de Automática* pp. 56–65.
- Medeiros, A. A. D. (1998b). A survey of control architectures for autonomous mobile robots, *JBCS - Journal of the Brazilian Computer Society, special issue on Robotics* 4.
- Menezes, P. J. C. (1999). *Estudos em navegação de robôs móveis*, Master's thesis, Universidade de Coimbra, Departamento de Engenharia Electrotécnica.
- Min (2003). Lego Mindstorms. Disponível em <http://mindstorms.lego.com>.
- Murphy, R. R. (2000). *Introduction to AI Robotics*, MIT Press.
- Pacheco, R. N. e Costa, A. H. R. (2002). Navegação de robôs móveis utilizando campos de potenciais, *Workshop de Computacao* pp. 125–130.
- Pedrosa, D. P. F. (2001). *Sistema de navegação para robôs móveis autônomos*, Master's thesis, Universidade Federal do Rio Grande do Norte, Departamento de Engenharia Elétrica.
- Pires, P. S. M. (2003). Métodos computacionais, <http://www.dca.ufrn.br/~pmotta>.
- Ribeiro, C. H. C., Costa, A. H. R. e Romero, R. A. F. (2001). Robôs móveis inteligentes: Princípios e técnicas, Divisão de Ciência da Computação, Instituto Tecnológico de Aeronáutica; Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica - Universidade de São Paulo; Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo.
- Sarcinelli-filho, M., Bastos-filho, T. F. e Freitas, R. A. C. (2002). Navegação de robôs móveis por reconhecimento de referências à base de sensoriamento ultra-sônico e visão monocular, *XIV Congresso Brasileiro de Automática*, pp. 2397–2402.

- Schilling, R. J. (1990). *Fundamentals of Robotics: Analysis and Control*, Prentice Hall.
- Schmidt, R. M. (2000). Identificação visual para navegação de robôs móveis utilizando um controlador fuzzy sintonizado por otimização numérica, *Anais do XIII Congresso Brasileira de Automática*, pp. 2245–2250.
- Se, S., Lowe, D. e Little, J. (2001). Local and global localization for mobile robots using visual landmarks, *International Conference on Intelligent Robotics and Systems*.
- Silva, L. R. (2001). *Análise e programação de robôs móveis autônomo da plataforma eyebot*, Master's thesis, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica.
- Tonouchi, Y., Tsubouchi, T. e Arimoto, S. (1994). Fusion of dead-reckoning positions with a workspace model for a mobile robot by bayesian inference, *International Conference on Intelligent Robots and Systems*, pp. 1347–1354.
- Valgas-filho, J. H. A. (2002). *Uma metodologia de correção dinâmica de erros de odometria em robôs móveis*, Master's thesis, Universidade Federal de Minas Gerais, Departamento de Ciências da Computação.
- Vassallo, R. F. (1998). *Uma estratégia de navegação para robôs móveis em ambientes interiores*, Master's thesis, Universidade Federal do Espírito Santo, Departamento de Engenharia Elétrica.
- Veen, T. M. V. e Groen, F. C. A. (1981). Discretization errors in the hough transform, *Pattern Recognition* **14**: 137–145.
- Vlassis, N., Motomura, Y., Hara, I. e Asoh, H. (2001). Edge-cased features from omnidirectional images for robot localization, *Proceedings of International Conference on Robotics and Automation*.

Wangenheim, A. V. (2003). Seminário introdução à visão computacional, Disponível em www.inf.ufsc.br/~visao.

Xu, L., Oja, E. e Kultanen, P. (1990). A new curve detection method: Randomized hough transform (rht), *Pattern Recognition* **11**: 331–338.