

Universidade Federal do Rio Grande do Norte

Centro de Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica

**Posicionamento e Movimentação de um Robô
Humanóide Utilizando Imagens de uma Câmera
Móvel Externa**

Marcelo Borges Nogueira

Natal, dezembro de 2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

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

**POSICIONAMENTO E MOVIMENTAÇÃO DE UM
ROBÔ HUMANÓIDE UTILIZANDO IMAGENS DE UMA
CÂMERA MÓVEL EXTERNA**

Marcelo Borges Nogueira

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. Adelardo Adelino Dantas de Medeiros

Natal, dezembro de 2005

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

Aprovada em 20 de dezembro de 2005 pela comissão examinadora,
formada pelos seguintes membros:

Prof. Dr. Adelardo Adelino Dantas de Medeiros (Orientador)
Universidade Federal do Rio Grande do Norte

Prof. Dr. Pablo Javier Alsina (Examinador Interno)
Universidade Federal do Rio Grande do Norte

Profa. Dra. Sílvia Silva da Costa Botelho (Examinador Externo)
Fundação Universidade Federal do Rio Grande

*“Se o conhecimento pode gerar
problemas, não é através da
ignorância que podemos
solucioná-los.”*

Isaac Asimov

Resumo

Este trabalho propõe um método para localizar um robô humanóide simples, sem sensores embarcados, utilizando imagens obtidas por uma câmera externa e técnicas de processamento de imagens. Localizando o robô em relação à câmera, e supondo conhecida a posição da câmera em relação ao mundo, podemos determinar a posição do robô humanóide em relação ao mundo. Para que a posição da câmera não seja fixa, utilizamos um outro robô móvel com rodas, dotado de um sistema de localização preciso, sobre o qual será colocada a câmera.

Uma vez que o humanóide seja localizado no ambiente de trabalho, podemos tomar as ações necessárias para realizar a sua movimentação. Simultaneamente, movimentamos o robô que contém a câmera, de forma que este tenha uma boa visada do humanóide.

As principais contribuições deste trabalho são:

- a idéia de utilizar um segundo robô móvel para auxiliar a movimentação de um robô humanóide sem eletrônica embarcada avançada;
- a escolha de métodos de calibração dos parâmetros intrínsecos e extrínsecos da câmera apropriados para a aplicação em questão, especialmente na parte em tempo real; e
- o algoritmo colaborativo de movimentação simultânea dos dois robôs.

Abstract

This work proposes a method to localize a simple humanoid robot, without embedded sensors, using images taken from an external camera and image processing techniques. Once the robot is localized relative to the camera, supposing we know the position of the camera relative to the world, we can compute the position of the robot relative to the world. To make the camera move in the work space, we will use another mobile robot with wheels, which has a precise locating system, and will place the camera on it.

Once the humanoid is localized in the work space, we can take the necessary actions to move it. Simultaneously, we will move the camera robot, so it will take good images of the humanoid.

The main contributions of this work are:

- the idea of using another mobile robot to aid the navigation of a humanoid robot without advanced embedded electronics;
- choosing of the intrinsic and extrinsic calibration methods appropriated to the task, especially in the real time part; and
- the collaborative algorithm of simultaneous navigation of the robots.

Agradecimentos

Gostaria de agradecer a todos que me ajudaram a realizar este trabalho. Especialmente aos meus pais, Roussel e Ângela, pelo incentivo e apoio; à minha irmã Ethel, por me iniciar no “mundo da computação” e à Meika, por estar presente e ajudando em todos os momentos.

Ao Prof. Adelardo, pela ótima orientação; ao Prof. Pablo, pelas dicas valiosas e à Bruno, pela contribuição ao trabalho.

À galera do mal, Ana C., Cynthia, Lantônio (Marconi), Sávio e Valnaide, pela amizade incondicional.

Aos amigos Arthur, Isaac, Jamerson, Mário, Pablo, Plínio e Ton, ao pessoal dos Laboratórios de Sistemas Inteligentes e de Pós-Graduação Anderson, Giovani, Goy, Henrique, João Paulo, Ricardo e Vidão (Danilo) e ainda à Belanha (Bella Reia), pelos momentos de descontração.

À Seu Abelardo e Dona Akemi, por terem me acolhido em sua casa.

À CAPES, pela ajuda financeira; ao Departametno de Engenharia de Computação e Automação e ao Programa de Pós-Graduação em Engenharia Elétrica, pela oportunidade.

Sumário

1	Introdução	1
1.1	Objetivos	3
1.2	Metodologia	3
2	Robôs Humanóides	5
2.1	Tipos de Robôs Bípedes	5
2.2	Tipos de Marcha	6
2.2.1	Marcha Estática	7
2.2.2	Marcha Dinâmica	7
2.3	Cérebro Remoto	8
3	Estimação de pose por Imagens	11
3.1	Calibração de Câmera	12
3.2	Notações	12
3.3	Parâmetros Intrínsecos	14
3.3.1	Método de Zhang	16
3.4	Parâmetros Extrínsecos	21
3.4.1	Método P4P	22
3.5	Processamento Digital de Imagens	26
3.5.1	Extração dos pontos para a estimação dos parâmetros intrínsecos	26
3.5.2	Extração dos pontos para a estimação dos parâmetros extrínsecos	30
4	Sistema Proposto	31
4.1	Robô Mestre	31
4.2	Robô Escravo	33

4.3	Funcionamento do Sistema	34
4.4	Localização do Robô Mestre	36
4.4.1	Determinação dos Parâmetros Intrínsecos	36
4.4.2	Determinação dos Parâmetros Extrínsecos	39
4.5	Movimentação do Robô Mestre	43
4.6	Movimentação do Robô Escravo	44
5	Experimentos e Resultados	47
5.1	Calibração dos Parâmetros Intrínsecos	47
5.1.1	Detecção dos Pontos Notáveis	47
5.1.2	Método de Zhang	48
5.2	Calibração Externa	49
5.2.1	Detecção dos Vértices do Losango	49
5.2.2	Método P4P	49
5.3	Localização e Movimentação do Robô Mestre	54
5.4	Movimentação do Robô Escravo	59
5.4.1	Determinação da Distância d e do Ângulo Crítico θ_c	60
5.4.2	Posicionamento do Robô Escravo	64
6	Conclusões e Perspectivas	67
	Referências Bibliográficas	70

Lista de Figuras

1.1	Sistema proposto	4
2.1	Robô puramente estático	6
2.2	Robô dinâmico	8
2.3	Esquema de Cérebro Remoto	9
3.1	Problema da projeção 2D	13
3.2	Transformação entre o referencial da câmera e o referencial da imagem	14
3.3	Padrão plano de pontos	27
3.4	Ponto fornecido pela interseção entre duas retas	28
3.5	Pontos fornecidos pela interseção entre várias retas	28
3.6	Parâmetros de uma representação normal de uma reta r	29
3.7	(a) Pontos em espaço cartesiano; (b) Transformada de Hough utilizando apresentação normal para os pontos A, B e C e sua interseção .	30
4.1	Robosapien	32
4.2	Controle remoto infra-vermelho	32
4.3	Robô escravo	34
4.4	Marca utilizada para estimação de pose do robô humanóide	35
4.5	Padrão utilizado na calibração interna da câmera	36
4.6	Transformada de Hough para determinação de duas retas sem distanciamento	37
4.7	Transformada de Hough para determinação de duas retas com distanciamento	38
4.8	Losango utilizado na calibração externa	40
4.9	Deteção de pontos pertencentes às arestas do losango	42

4.10	Movimentação do humanóide	44
4.11	Circuito transmissor infra-vermelho	44
4.12	Situação de ângulo crítico na movimentação da câmera	45
5.1	Resultado da aplicação do algoritmo de Hough a quatro imagens do padrão utilizado na calibração interna da câmera	48
5.2	(a)Imagem original distorcida (b) Imagem com distorção removida a partir dos coeficientes calculados	49
5.3	Resultado da aplicação do algoritmo de detecção dos vértices do losango aplicado a seis imagens diferentes	50
5.4	Imagem dos losangos utilizada no primeiro experimento de estimação do erro da calibração de câmera	52
5.5	Imagem dos losangos utilizada no segundo experimento de estimação do erro da calibração de câmera	52
5.6	Orientação do referencial do robô humanóide e seus ângulos de rotação de acordo com os ângulos Roll, Pitch e Yaw	53
5.7	Gráfico da posição x do robô mestre em função do quadro	56
5.8	Gráfico da posição z do robô mestre em função do quadro	56
5.9	Gráfico do ângulo Pitch do robô mestre em função do quadro	57
5.10	Posição z para os dois experimentos	57
5.11	Posição x para os dois experimentos	57
5.12	Ângulo Pitch para os dois experimentos	58
5.13	Posição z do robô mestre em relação à câmera	59
5.14	Posição z da câmera em relação ao mundo	59
5.15	Posição z do robô mestre em relação ao mundo	60
5.16	Losango muito próximo da câmera causando movimentos bruscos e embaçamento da imagem	61
5.17	Desvio padrão na estimação da posição z em função da razão r	61
5.18	Desvio padrão na estimação do ângulo Pitch em função da razão r	62
5.19	Relação entre a distância do robô mestre para a câmera e a razão entre a área do losango e a área da imagem	62
5.20	Relação entre o ângulo Pitch e o desvio padrão na estimação da pose no ângulo Pitch	63

5.21	Relação entre o ângulo Pitch e o desvio padrão na estimação da pose no eixo z	63
5.22	Posição do robô mestre e do robô escravo no eixo Z	64
5.23	Posição do robô mestre e do robô escravo no eixo X	65
5.24	Posição onde situação de ângulo crítico foi detectada	65

Lista de Tabelas

4.1	Comandos IR básicos do Robosapien	33
4.2	Detecção dos pontos pertencentes às arestas do losango	41
4.3	Detecção das arestas do losango	42
5.1	Vértices dos losangos no referencial R	51
5.2	Erros da calibração de câmera para a figura 5.4	54
5.3	Erros da calibração de câmera para a figura 5.5	54

Capítulo 1

Introdução

Os robôs estão sendo cada vez mais empregados para auxiliar ou substituir o homem em diversas tarefas, tais como o transporte de carga, limpeza e manutenção de ambientes, segurança, diversão, o patrulhamento e a exploração de ambientes perigosos ou nocivos ao ser humano, em tarefas repetitivas e em tarefas de precisão [Sil01].

Podemos dividir os robôs basicamente em dois grupos: os robôs manipuladores e robôs móveis [Cra86]. Estes últimos, como o nome sugere, possuem a capacidade de se locomover por seu espaço de trabalho, oferecendo assim uma maior gama de aplicações. Os robôs móveis que utilizam duas pernas (bípedes) são a base para construção de robôs humanóides, os quais possuem semelhança física com o corpo humano, ou seja, possuem pernas e braços.

Provavelmente, o melhor *design* de um robô para uma determinada tarefa não seria o de um robô humanóide. Estes, atualmente, ainda têm uma área de aplicação restrita, sendo utilizados principalmente na indústria do entretenimento [Tan03]. Porém, em um futuro próximo, a sociedade irá enfrentar uma crescente necessidade por robôs que auxiliem ou substituam tarefas realizadas por humanos em ambientes tais como escritórios, casas, hospitais e espaços abertos. Tais robôs devem realizar tarefas nestes ambientes "humanos". Por esse motivo, será necessário um robô com uma forma similar à forma humana. [HLNT01].

A respeito dos robôs humanóides, podemos citar algumas das vantagens do seu

design [Tan03]:

- O comportamento dos robôs humanóides produz sentimentos que facilitam a comunicação entre máquina e homem;
- A forma humana é uma das melhores para robôs controlados remotamente;
- Existem muitos casos em que se faz necessário uma máquina semelhante ao homem para poder substituí-lo em seu ambiente de trabalho.

Podemos dizer, então, que para realizar um conjunto de tarefas não especificadas previamente em um ambiente construído para humanos, o robô humanóide se torna uma boa escolha. Isto acontece porque os ambientes para humanos são construídos de forma a possibilitar a interação destes com o ambiente a partir das características do seu corpo. Dessa forma, o robô humanóide tem uma maior possibilidade de interagir com tais ambientes, já que este possui várias características do corpo humano. Por exemplo, um robô humanóide pode subir escadas, abrir portas, pegar um objeto, etc. Dentre as aplicações que um robô humanóide poderá desenvolver, no futuro, temos manutenção de plantas industriais, trabalhos cooperativos em espaço aberto, segurança caseira, tele-operação de máquinas de construção e robôs de assistência aos humanos.

Dentre os problemas da robótica móvel, podemos destacar o da navegação. 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, 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.

No caso de robôs com pernas, essa tarefa se torna ainda mais complexa, pois ele pode perder o equilíbrio e cair. Por este motivo, vários pesquisadores têm estudado

tal processo. Para desempenhar totalmente sua função, o robô deveria ser capaz de se locomover em vários ambientes, em diferentes tipos de piso, e com vários tipos de obstáculos. Também seria uma qualidade importante consumir pouca energia, além de andar o mais rápido possível. Tais requisitos são encontrados na natureza. Então, uma solução é copiá-la, construindo robôs com formas encontradas na natureza, como, por exemplo, a forma de um dos bípedes mais bem sucedidos: o homem.

1.1 Objetivos

A construção de um robô humanóide em tamanho natural se mostra uma tarefa difícil, demorada e cara. Por esse motivo, muitos dos robôs humanóides desenvolvidos na atualidade são de pequeno porte. Porém, o tamanho reduzido destes robôs apresenta um problema: a dificuldade de inserção de componentes embarcados na plataforma, como sensores (câmeras) e processadores. Com isso, se mostra útil utilizar uma técnica de sensoriamento a distância, assim como um processamento remoto de informações, enviando comandos ao robô logo após tal processamento. Neste trabalho propomos efetuar a localização e movimentação de um robô humanóide, sem uma eletrônica embarcada avançada, de forma autônoma, em seu ambiente de trabalho. Isto será realizado através de imagens obtidas do mesmo a partir de uma câmera montada sobre um segundo robô móvel com rodas.

1.2 Metodologia

Primeiramente, iremos adquirir um robô humanóide de pequeno porte, barato, operado remotamente e sem sensores visuais ou processadores embarcados. Para realizar a localização e movimentação do robô humanóide, iremos utilizar um robô móvel, chamado robô escravo, com um sensor externoceptivo com relação ao robô humanóide (robô mestre). O robô escravo, equipado com uma câmera, irá obter

imagens do robô mestre. A partir destas imagens, o robô escravo será capaz de determinar a posição e orientação relativa do robô mestre, podendo assim gerar uma trajetória e enviar os devidos comandos para o robô mestre, de forma a realizar sua navegação pelo ambiente. Na medida em que o robô mestre se movimenta pelo ambiente o robô escravo também deve se deslocar acompanhando o robô mestre. O sistema proposto, que será mais bem explicado no capítulo 4, é ilustrado na figura 1.1.

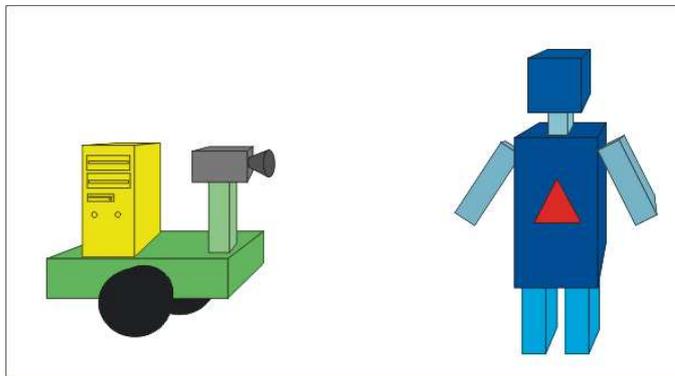


Figura 1.1: Sistema proposto

A vantagem deste método é que o robô escravo, por ser um veículo com rodas, pode utilizar métodos de localização de eletrônica mais simples e barata, como odometria ou marcos naturais.

Este trabalho está organizado em seis capítulos. Neste capítulo 1, foi feita uma breve introdução sobre robótica, além de apresentar o objetivo do trabalho. Informações sobre robôs humanóides, focando principalmente sua locomoção, serão discutidas no capítulo 2. Os métodos de visão robótica utilizados para a estimação da pose de um objeto a partir de imagens serão revisados no capítulo 3. No capítulo 4 serão mostrados os métodos utilizados na implementação do sistema. Experimentos e resultados serão apresentados no capítulo 5. Finalmente, no capítulo 6, os resultados serão discutidos e analisados.

Capítulo 2

Robôs Humanóides

A grande dificuldade em robôs que apresentam pernas é justamente a sua locomoção. No caso de robôs móveis com rodas, essa primitiva é bastante simples, uma vez que com o simples acionamento dos motores podemos fazer com que este se mova. Já no caso de robôs com pernas, devemos movimentá-las de forma sincronizada para realizar um deslocamento, com a preocupação adicional de não fazê-lo perder o equilíbrio e cair. Por este motivo, o deslocamento de robôs desse tipo é um grande desafio. A principal preocupação quanto ao controle de robôs humanóides é a falta de estabilidade do mesmo. Portanto, andar com estabilidade é um pré-requisito fundamental no desenvolvimento de um robô de tal categoria. A forma de controle das pernas de um robô bípede pode variar de acordo com o tipo de construção deste, assim como com o tipo de marcha que se deseja que o mesmo desenvolva. A seguir mostraremos tais possibilidades.

2.1 Tipos de Robôs Bípedes

A classificação dos robôs bípedes [Kul] é feita em duas categorias:

- Robôs puramente estáticos: o centro de massa do robô sempre se encontra no interior do perímetro do pé de apoio deste. Dessa forma, o robô nunca perde o equilíbrio, independente do movimento realizado pelas pernas. Um robô desse

tipo pode ser visto na figura 2.1.



Figura 2.1: Robô puramente estático

- Robôs dinâmicos: dependendo do movimento realizado pelas suas pernas, o centro de massa pode vir a se encontrar fora do perímetro do pé de apoio do mesmo.

No caso dos robôs puramente estáticos, o controle é bastante simples, já que não devemos nos preocupar com a estabilidade do mesmo. Porém, este tipo de robô possui um *design* que impossibilita várias aplicações, já que a forma humana de se locomover não é estática. Portanto, os robôs dinâmicos são mais interessantes por terem uma maior gama de aplicações.

2.2 Tipos de Marcha

Existem basicamente duas técnicas de controle para realizar a locomoção de robôs bípedes dinâmicos: a estática e a dinâmica.

2.2.1 Marcha Estática

Neste tipo de marcha, o robô deve realizar movimentos de tal forma que o seu centro de massa permaneça sempre interior ao perímetro do seu pé de apoio. Dessa forma, o robô estará sempre estável. Contudo, este tipo de locomoção restringe os movimentos do robô a movimentos com velocidades constantes e lentas, para não introduzir componentes dinâmicas. Assim, a velocidade de movimento do robô fica limitada [KTII99].

2.2.2 Marcha Dinâmica

Em oposição à marcha estática, a marcha dinâmica não exige que o robô esteja sempre em equilíbrio. Neste caso, a exigência é com respeito ao ZMP, ou *Zero Moment Point*. Este tem que estar sempre no interior do perímetro do pé de apoio. A definição do ZMP vem do princípio de D’Alambert: ele reduz o movimento de n pontos de massa, influenciados por forças externas, a um problema estático. Ele diz o seguinte: assuma um robô com n pontos de massa m_i . Seja $r_i = [x_i, y_i, z_i]$ a posição de m_i , $P = [X_p, Y_p, 0]$ um ponto arbitrário, $G = [g_x, g_y, g_z]$ a aceleração gravitacional e $T = [T_x, T_y, T_z]$ o torque total aplicado no ponto P ; a equação do movimento é:

$$\sum m_i (r_i - P) \times (\ddot{r}_i + G) + T = 0 \quad (2.1)$$

Fazendo $T = 0$, iremos descobrir o ponto no qual não há torque aplicado sobre o robô, ou seja, o ZMP:

$$x = \frac{\sum m_i (\dot{z}_i + g_z) x_i - \sum m_i (\dot{x}_i + g_x) z_i}{\sum m_i (\dot{z}_i + g_z)} \quad (2.2)$$

$$y = \frac{\sum m_i (\dot{z}_i + g_z) y_i - \sum m_i (\dot{y}_i + g_y) z_i}{\sum m_i (\dot{z}_i + g_z)} \quad (2.3)$$

A vantagem desse tipo de marcha é a velocidade dos movimentos que o robô pode realizar, proporcionando uma maior agilidade deste. Um robô puramente estático

(figura 2.1) é capaz de realizar apenas a marcha estática. Porém, um robô dinâmico (figura 2.2) é capaz de realizar tanto a marcha estática como a marcha dinâmica.



Figura 2.2: Robô dinâmico

2.3 Cérebro Remoto

As dificuldades em visão robótica para humanóides se dão devido à limitação do *hardware*. Devido a esta dificuldade de se desenvolver um sistema de visão avançado em um hardware limitado, Inaba *et al* [IKKI95] sugeriram um sistema de cérebro remoto. Como o nome sugere, o cérebro não se encontra no robô, e sim no que ele chamou de "ambiente mãe". O corpo e o cérebro seriam conectados por *links* sem fio. Dessa forma, a construção do robô se torna mais fácil, podendo ser mais leve e ter mais graus de liberdade. A figura 2.3 ilustra o sistema proposto.

Assim, é possível se construir um robô pequeno e leve com um cérebro grande e pesado. Cada robô desenvolvido tem sua característica física moldada de acordo com a aplicação deste, porém, desde que este seja construído de acordo com o sistema de cérebro remoto, o cérebro utilizado pode ser o mesmo, compartilhando assim um

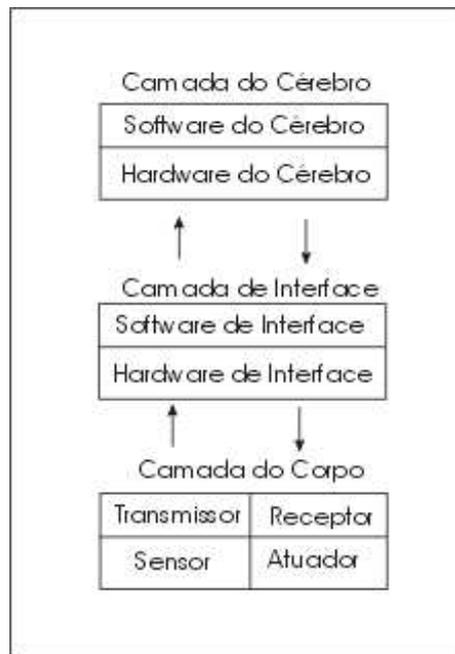


Figura 2.3: Esquema de Cérebro Remoto

único cérebro com novas gerações de robôs. Além disso, a evolução do cérebro se torna mais fácil e mais barata, devido a sua não miniaturização.

O objetivo deste trabalho não é estudar a marcha desenvolvida por um robô humanóide, e sim realizar sua movimentação pelo ambiente de trabalho. Para isso devemos saber a posição do robô em cada instante. Isto é feito utilizando-se algum tipo de sensoriamento, como por exemplo a odometria, um método bastante utilizado por ser simples e barato. Contudo a odometria se mostra um método inapropriado para robôs humanóides, independente do tipo de marcha que este possua, devido a sua passada ser de comprimento variável e seu deslizamento ser muito grande quando comparado a robôs com rodas.

Este trabalho sugere outro método de sensoriamento utilizando imagens obtidas por uma câmera. Por ser indiferente ao tipo de marcha, o robô utilizado neste trabalho já possui um algoritmo de marcha embutido (marcha estática). Além disso, o humanóide adquirido é de pequeno porte e possui uma eletrônica embarcada barata, sem grande capacidade de processamento. Porém, por utilizar imagens para

o sensoriamento, o robô deve possuir uma grande capacidade de processamento. Faz-se assim necessário a utilização de um cérebro remoto.

Capítulo 3

Estimação de pose por Imagens

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. Para se localizar, o robô faz uso de sensores para obter informações do ambiente. Dentre os sensores mais utilizados podemos citar encoders, sonares, câmeras, bússolas, lasers, etc. Um dos métodos mais simples, baratos e populares é a odometria [AC04]. Esta consiste na determinação da posição e orientação do robô através da integração dos deslocamentos incrementais, medidos a partir de um referencial fixo. No caso de robôs móveis com rodas, basta medir a rotação desta e, sendo o raio conhecido, pode-se calcular o deslocamento realizado. Porém, no caso de robôs com pernas, a tarefa não é tão simples, pois estes não têm um passo de tamanho fixo, além de o deslizamento ser, normalmente, bem maior do que em robôs com rodas. Por este motivo, a localização de robôs humanóides sugere outros métodos. Alguns destes são muito caros, como o laser e o GPS. Já outros fornecem informações escassas[AC04], como o sonar. Dessa forma, um sistema de navegação visual se torna atrativo devido ao seu baixo custo e ao alto poder de processamento dos computadores atuais.

A utilização da visão computacional em robótica sempre foi uma grande área de pesquisa. Durante a década de 70, a visão foi usada largamente para reconhecimento de padrões. Hoje em dia, devido aos avanços da computação, oferecendo um maior

poder de processamento por um custo reduzido, as áreas de aplicação vão desde localização e inspeção até visão estéreo, tudo isso sendo realizado em tempo real. Dentre essas novas aplicações, um dos problemas principais é o da calibração de câmera.

3.1 Calibração de Câmera

A calibração de câmera consiste em determinar um conjunto de valores relativos à mesma que possibilitem interpretar imagens obtidas por esta. “*Em visão computacional 3D, a calibração de câmera é um passo necessário para se obter informações métricas a partir de uma imagem 2D*” [Zha00]. A calibração de câmera é um assunto que ainda é bastante discutido na literatura atualmente [RVSS05, KGK05, HSH05, RSL05, JCC⁺05].

A calibração de câmera pode ser dividida em duas etapas:

- Determinação dos seus parâmetros intrínsecos

Consiste em determinar as propriedades intrínsecas da câmera, como: distância focal, centro de imagem, coeficiente de distorção das lentes e fator de escala.

- Determinação dos seus parâmetros extrínsecos

Os parâmetros extrínsecos indicam a posição e a orientação da câmera com relação ao sistema de coordenadas do mundo.

3.2 Notações

Suponha $\{R\}$ o sistema de coordenadas 3D, em centímetros, do mundo, $\{C\}$ o sistema de coordenadas 3D, em centímetros, da câmera, $\{I\}$ o sistema de coordenadas 2D, em centímetros, da imagem, e $\{F\}$ o sistema de coordenadas 2D, em píxeis, da imagem.

Suponha um ponto P em relação a $\{R\}$, isto é ${}^R P = [R_x \ R_y \ R_z]^T$. Este mesmo

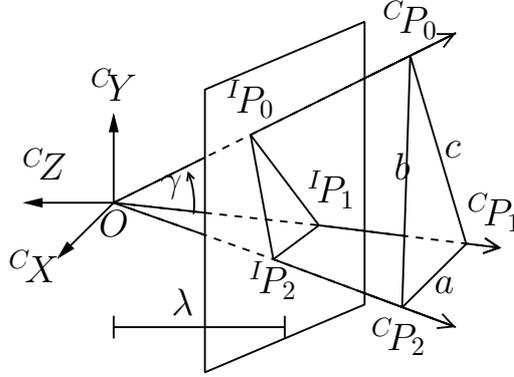


Figura 3.1: Problema da projeção 2D

ponto pode ser expresso em relação a $\{C\}$ como mostrado na equação 3.1, onde ${}^C P = [c_x \ c_y \ c_z]^T$ é um ponto em relação ao referencial da câmera e ${}^C T_R$ é a matriz de transformação entre $\{C\}$ e $\{R\}$.

$$[{}^C P \ 1]^T = {}^C T_R \cdot [{}^R P \ 1]^T \quad (3.1)$$

Cada ponto no referencial do mundo tem uma representação no plano de projeção da imagem, como mostrado na figura 3.1. Utilizando o modelo de câmera *pinhole*, a relação entre as coordenadas da imagem ${}^I P = [I_x \ I_y]^T$ e as coordenadas da câmera ${}^C P$ é mostrada nas equações 3.2 e 3.3, onde λ é a distância focal da câmera em centímetros (figura 3.2).

$$I_x = \frac{\lambda}{c_z} c_x \quad (3.2)$$

$$I_y = \frac{\lambda}{c_z} c_y \quad (3.3)$$

A matriz D é a matriz de conversão de centímetros para píxeis, como mostrado na equação 3.4, sendo ${}^F P = [F_x \ F_y]^T$ um ponto no referencial $\{F\}$.

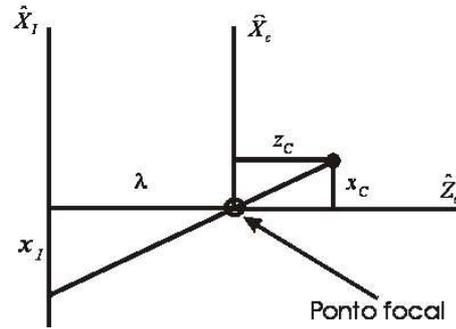


Figura 3.2: Transformação entre o referencial da câmera e o referencial da imagem

$$\begin{bmatrix} FP & 1 \end{bmatrix}^T = D \cdot \begin{bmatrix} IP & 1 \end{bmatrix}^T \quad (3.4)$$

A matriz D possui o seguinte formato:

$$D = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.5)$$

3.3 Parâmetros Intrínsecos

Existem vários métodos na literatura sobre a calibração dos parâmetros intrínsecos. Tais métodos podem ser divididos, basicamente, em duas categorias: fotogrametria e auto-calibração.

Na fotogrametria a calibração da câmera é realizada observando-se uma ou mais imagens de objetos cuja geometria no espaço é conhecida com precisão. Este método é bastante eficiente [Fau93]. Normalmente, o objeto de calibração consiste em dois ou três planos ortogonais entre si. Fukui [Fuk81] utilizou um objeto em forma de diamante. Nele, o centro óptico da câmera e do objeto devem estar na mesma altura. Magee e Aggarwal [MA84] utilizaram uma esfera com dois grandes círculos perpendiculares e, antes da imagem ser adquirida, o centro óptico da câmera deve passar pelo centro da esfera. Tasi [Tsa87] sugere um método que utiliza uma translação precisamente conhecida.

O principal problema dos métodos utilizados na fotogrametria é que eles exigem uma alta precisão nas medições dos objetos ou deslocamentos realizados pela câmera utilizados na calibração.

Na auto-calibração, não são usados objetos conhecidos previamente [GS03, MF92, FM89, DB93]. A calibração é realizada apenas capturando-se imagens com uma mesma câmera de uma cena estática. Em seguida é feita a correspondência entre pontos presentes nestas imagens. Faugeras [FLM92] sugere um método de auto-calibração onde 3 imagens diferentes de uma cena são necessárias para calibrar a câmera. Inicialmente, ele calcula os epipólos das imagens e em seguida utiliza as equações de Kruppa [LH81] para concluir a calibração.

Os métodos de auto-calibração se mostram interessantes quando não é possível realizar a calibração da câmera em um ambiente controlado, observando objetos conhecidos. Além disso eles oferecem a capacidade de calibrar a câmera dinamicamente durante o funcionamento do sistema em que ela é utilizada. Porém, eles exigem um maior processamento para realizar a correspondência entre os pontos das imagens, além de normalmente serem mais complicados e de fornecerem resultados menos precisos.

Existem ainda outras técnicas utilizando desde ponto de fuga [WT91, YZ05] até mesmo a gravidade [SQ00], entre outros [NSNI92, GS03].

Neste trabalho, desejamos calibrar os parâmetros intrínsecos da câmera apenas uma vez, antes do início do processo. Isto pode ser realizado utilizando um objeto de dimensões conhecidas, como nos métodos de fotogrametria. Porém, não dispomos de aparelhos precisos para medir as dimensões do objeto nem para medir deslocamentos realizados pela câmera entre imagens obtidas do objeto. Por este motivo, resolvemos utilizar o método proposto por Zhang [Zha00], o qual se encontra entre a fotogrametria e a auto-calibração. As vantagens deste método são flexibilidade, robustez e baixo custo de processamento.

3.3.1 Método de Zhang

Neste método, é necessário que a câmera observe um padrão plano de, no mínimo, duas posições diferentes. Porém, não é necessário que se tenha conhecimento sobre o deslocamento realizado pela câmera entre uma observação e outra. O padrão plano deve ser fixado sobre uma superfície razoavelmente plana.

Utilizando as equações 3.1, 3.2, 3.3 e 3.4, teremos:

$$\begin{bmatrix} \frac{c_z}{\lambda} & 0 & 0 \\ 0 & \frac{c_z}{\lambda} & 0 \\ 0 & 0 & c_z \\ 0 & 0 & 1 \end{bmatrix} \cdot D^{-1} \cdot \begin{bmatrix} FP & 1 \end{bmatrix}^T = {}^C T_R \cdot \begin{bmatrix} RP & 1 \end{bmatrix}^T \quad (3.6)$$

$$\begin{bmatrix} \frac{c_z}{\lambda} & 0 & 0 \\ 0 & \frac{c_z}{\lambda} & 0 \\ 0 & 0 & c_z \\ 0 & 0 & 1 \end{bmatrix} \cdot D^{-1} \cdot \begin{bmatrix} FP \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix}. \quad (3.7)$$

Se

$$D^{-1} = \begin{bmatrix} a' & b' & c' \\ 0 & d' & e' \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.8)$$

então podemos obter:

$$\begin{bmatrix} C_z \frac{a'}{\lambda} & C_z \frac{b'}{\lambda} & C_z \frac{c'}{\lambda} \\ 0 & C_z \frac{d'}{\lambda} & C_z \frac{e'}{\lambda} \\ 0 & 0 & C_z \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} FP \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix}. \quad (3.9)$$

Já que a equação fornecida pela última linha das matrizes é inútil, tais linhas podem ser removidas:

$$\begin{bmatrix} C_z \frac{a'}{\lambda} & C_z \frac{b'}{\lambda} & C_z \frac{c'}{\lambda} \\ 0 & C_z \frac{d'}{\lambda} & C_z \frac{e'}{\lambda} \\ 0 & 0 & C_z \end{bmatrix} \cdot \begin{bmatrix} FP \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix} \quad (3.10)$$

$$C_z \begin{bmatrix} \frac{a'}{\lambda} & \frac{b'}{\lambda} & \frac{c'}{\lambda} \\ 0 & \frac{d'}{\lambda} & \frac{e'}{\lambda} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} FP \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix}. \quad (3.11)$$

Fazendo

$$A^{-1} = \begin{bmatrix} \frac{a'}{\lambda} & \frac{b'}{\lambda} & \frac{c'}{\lambda} \\ 0 & \frac{d'}{\lambda} & \frac{e'}{\lambda} \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.12)$$

teremos que

$$C_z A^{-1} \cdot \begin{bmatrix} FP \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix} \quad (3.13)$$

$$C_z \begin{bmatrix} FP \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} r_1 & r_2 & r_3 & t \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ R_z \\ 1 \end{bmatrix}, \quad (3.14)$$

onde A é a matriz de parâmetros intrínsecos, dada por:

$$A = \begin{bmatrix} \alpha & \gamma & Fx_0 \\ 0 & \beta & Fy_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.15)$$

onde (F_x_0, F_y_0) são as coordenadas do centro de imagem, em píxeis, α e β são os fatores de escala nos eixos x e y da imagem, respectivamente, e γ é a distorção do ângulo formado pelos eixos.

Sem falta de generalidade, podemos assumir que $R_z = 0$, o que nos dá:

$$C_z \begin{bmatrix} FP \\ 1 \end{bmatrix} = A \cdot \begin{bmatrix} r_1 & r_2 & t \end{bmatrix} \cdot \begin{bmatrix} R_x \\ R_y \\ 1 \end{bmatrix}. \quad (3.16)$$

Podemos dizer, então, que um ponto ${}^R P' = [\ r_x \ r_y \ 1 \]^T$ e sua imagem ${}^F P$ se relacionam pela homografia H :

$$q \left[{}^F P \ 1 \right]^T = H \left[{}^R P' \ 1 \right]^T \quad \text{onde } H = A \cdot [\ r_1 \ r_2 \ t \] \text{ e } q = C_z . \quad (3.17)$$

A matriz H , portanto, pode ser calculada, a menos de um fator de escala, a partir da equação 3.17.

Sendo $H = [\ h_1 \ h_2 \ h_3 \]$, temos:

$$[\ h_1 \ h_2 \ h_3 \] = sA [\ r_1 \ r_2 \ t \] \quad (3.18)$$

onde s é um fator de escala arbitrário. Sabendo que r_1 e r_2 são ortonormais, e sendo A^{-T} a abreviação para $(A^{-1})^T$ ou $(A^T)^{-1}$, podemos deduzir que:

$$h_1^T A^{-T} A^{-1} h_2 = 0 \quad (3.19)$$

$$h_1^T A^{-T} A^{-1} h_1 = h_2^T A^{-T} A^{-1} h_2 \quad (3.20)$$

Seja $B = A^{-T} A^{-1}$ (note que ela é uma matriz simétrica) representada pelo vetor

$$b = [\ B_{11} \ B_{12} \ B_{22} \ B_{13} \ B_{23} \ B_{33} \]$$

e seja $h_i = [\ h_{i1} \ h_{i2} \ h_{i3} \]^T$ a i -ésima coluna de H , temos que

$$h_i^T B h_j = v_{ij}^T b \quad (3.21)$$

onde

$$v_{ij} = [\ h_{i1} h_{j1} \ h_{i1} h_{j2} + h_{i2} h_{j1} \ h_{i2} h_{j2} \ h_{i3} h_{j1} + h_{i1} h_{j3} \ h_{i3} h_{j2} + h_{i2} h_{j3} \ h_{i3} h_{j3} \]^T$$

Dessa forma, podemos reescrever as equações 3.19 e 3.20 na seguinte forma:

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (3.22)$$

Portanto, observando n imagens do padrão, obtemos n equações do tipo 3.22, o que nos dá:

$$Vb = 0 \quad (3.23)$$

onde V é uma matriz $2n \times 6$.

Se $n \geq 3$, b pode ser calculado unicamente, exceto por um fator de escala. Se $n = 2$, podemos forçar com que $\gamma = 0$, i.e., $[0 \ 1 \ 0 \ 0 \ 0 \ 0]b = 0$, acrescentando assim mais uma equação ao sistema 3.23. Para calcular a resposta do sistema 3.23, podemos utilizar a técnica de decomposição em valores singulares [Leo98] da matriz $V^T * V$. A resposta do sistema será, então, o vetor coluna direito associado com o menor valor singular. Uma vez que sabemos o valor de b , podemos calcular a matriz A , como mostrado no conjunto de equações 3.24, que nos dá os parâmetros intrínsecos da câmera.

$$\begin{aligned} Fx_0 &= \frac{B_{12}B_{13} - B_{11}B_{23}}{B_{11}B_{22} - B_{12}^2} \\ s &= \frac{B_{33} - [B_{13}^2 + v_0(B_{12}B_{13} - B_{11}B_{23})]}{B_{11}} \\ \alpha &= \sqrt{\frac{s}{B_{11}}} \\ \beta &= \sqrt{\frac{sB_{11}}{B_{11}B_{22} - B_{12}^2}} \\ \gamma &= \frac{-B_{12}\alpha^2\beta}{s} \\ Fy_0 &= \frac{\gamma v_0}{\beta} - \frac{B_{13}\alpha^2}{s} \end{aligned} \quad (3.24)$$

Distorção Radial

Até o momento, ao calcularmos os parâmetros intrínsecos da câmera, não consideramos a distorção causada pelas lentes. Contudo, câmeras pessoais possuem uma distorção radial significativa [Zha00]. Zhang sugere que a função de distorção é totalmente dominada pelas componentes radiais, e que basta considerarmos os dois primeiros termos da distorção, já que modelos mais elaborados não fornecem melhores resultados além de causar instabilidade numérica [Tsa87, WM94].

Seja $({}^F u, {}^F v)$ o ponto da imagem ideal, sem distorção, e o ponto $({}^{F\tilde{u}}, {}^{F\tilde{v}})$ as coordenadas da imagem realmente observadas (ambos em píxeis). Os pontos ideais são a projeção dos pontos do sistema de coordenadas do mundo de acordo com o modelo de câmera *pinhole*. Similarmente, os pontos $({}^I x, {}^I y)$ e $({}^{I\tilde{x}}, {}^{I\tilde{y}})$ (em centímetros) são respectivamente o ponto ideal (sem distorção) e o ponto real (com distorção) da imagem. Com a distorção, teremos:

$${}^{I\tilde{x}} = {}^I x + {}^I x[k_1({}^I x^2 + {}^I y^2) + k_2({}^I x^2 + {}^I y^2)^2] \quad (3.25)$$

$${}^{I\tilde{y}} = {}^I y + {}^I y[k_1({}^I x^2 + {}^I y^2) + k_2({}^I x^2 + {}^I y^2)^2] \quad (3.26)$$

onde k_1 e k_2 são os coeficientes da distorção radial. O centro da distorção é o mesmo que o centro de imagem $({}^F u_0, {}^F v_0)$. Então, se ${}^{F\tilde{u}} = {}^F u_0 + \alpha {}^{I\tilde{x}} + \gamma {}^{I\tilde{y}}$, ${}^F u = {}^F u_0 + \alpha {}^I x + \gamma {}^I y$, ${}^{F\tilde{v}} = {}^F v_0 + \beta {}^{I\tilde{y}}$ e ${}^F v = {}^F v_0 + \beta {}^I y$, teremos:

$${}^{F\tilde{u}} = {}^F u + ({}^F u - {}^F u_0)[k_1({}^I x^2 + {}^I y^2) + k_2({}^I x^2 + {}^I y^2)^2] \quad (3.27)$$

$${}^{F\tilde{v}} = {}^F v + ({}^F v - {}^F v_0)[k_1({}^I x^2 + {}^I y^2) + k_2({}^I x^2 + {}^I y^2)^2] \quad (3.28)$$

Zhang sugere resolver os parâmetros k_1 e k_2 estimando, inicialmente, a matriz de parâmetros intrínsecos A , considerando que não existe distorção radial. A partir dessa estimativa inicial de A , pode-se calcular os parâmetros k_1 e k_2 . De posse destes, pode-se calcular novamente a matriz A , agora considerando a distorção radial, que por sua vez nos fornece novos valores para k_1 e k_2 . Este processo se repete até a convergência. Porém, Zhang afirmou que tal processo se mostrou lento. Por este motivo, ele propôs utilizar uma otimização da seguinte função:

$$\sum_{i=1}^n \sum_{j=1}^m \| {}^F P_{ij} - f(A, k_1, k_2, R_j, t_j, {}^R P_{ij}) \|^2 \quad (3.29)$$

onde $f(A, k_1, k_2, R_j, t_j, {}^R P_{ij})$ é a projeção do i -ésimo ponto na j -ésima imagem de acordo com a equação 3.14, seguido pela distorção radial de acordo com as equações 3.27 e 3.28. Os parâmetros R_j e t_j representam os parâmetros extrínsecos da câmera

para a j -ésima imagem (a rotação e a translação respectivamente). A estimativa inicial da matriz A pode ser calculada como mostrado na seção 3.3.1, e a estimativa inicial de R_j e t_j pode ser calculada como mostrado na seção 3.4.

3.4 Parâmetros Extrínsecos

Originado do processo de calibração de câmera, o problema de estimação de pose consiste em determinar a posição e orientação de um determinado objeto a partir de imagens obtidas deste, supondo que os parâmetros intrínsecos da câmera são conhecidos. Existem várias abordagens para este problema utilizando pontos, linhas [DRLR88, Hor87] e até mesmo utilizando a Karhunen-Loeve Transform [QC00] ou filtros de Kalman [WWHB96]. Porém, os métodos mais estudados e utilizados atualmente são os *Perspective-n-Point*, ou PnP .

Os métodos PnP utilizam um conjunto de n pontos notáveis presentes tanto no objeto como em uma imagem do objeto. O método a ser utilizado depende da quantidade de pontos detectados na imagem.

Um dos primeiros a abordar este problema obtendo resultados satisfatórios foi Longuet-Higgins, o qual utilizava duas imagens de uma cena com um conjunto de oito pontos notáveis em cada [LH81]. Porém, o fato deste método utilizar duas imagens o torna pouco prático.

O mínimo de pontos que nos fornece soluções finitas são três (P3P). Rives *et al* [RBPD81] construíram um sistema de três equações quadráticas e três incógnitas, as quais representam as distâncias do centro óptico da câmera para cada um dos três pontos no mundo. Neste caso existem oito soluções. Fischler e Bolles [FB81] conseguiram reduzir o número de soluções para quatro. Porém, com apenas três pontos, o P3P tem como desvantagem sempre fornecer múltiplas soluções.

Dementhon e Davis [DD95] propuseram um método iterativo para quatro pontos (P4P) enquanto Lowe [Low91] e Yuan [Yua89] desenvolveram técnicas baseadas no

método de Newton-Raphson. Estes métodos apresentam os problemas de métodos iterativos: inicialização e convergência. Já Horaud [HCLL89] converteu o problema P4P em um problema de três linhas especial. Infelizmente tal conversão utiliza apenas informação parcial e mantém as mesmas equações polinomiais de quarto grau [QL99]. No caso de quatro pontos, se estes forem coplanares é possível achar uma única solução analítica [FB81, KETK92].

Outros algoritmos que requerem menos que oito pontos foram propostos na literatura [DZLF94, Zha98], porém estes possuem equações não-lineares para serem resolvidas.

Uma das primeiras aplicações práticas da fotogrametria em robótica implementada em tempo real será utilizada em naves espaciais [PPKA81]. Apesar de utilizar métodos iterativos, o algoritmo iterava uma vez a cada imagem, permitindo assim ao algoritmo ser em tempo real. Para acelerar ainda mais o processo, Hughes [Hug86] limitou o algoritmo a apenas algumas configurações específicas.

Por ser um processo que deve ser realizado a cada etapa de amostragem, o método utilizado deve exigir o mínimo de pontos possível além de ter uma solução analítica. Por este motivo resolvemos utilizar o método P4P de Kamata, o qual nos fornece uma única solução, analítica, com o menor número de pontos possível: quatro.

Utilizando a notação descrita na seção 3.2, calibrar os parâmetros extrínsecos da câmera significa calcular a matriz de transformação ${}^C T_R$ (equação 3.1).

3.4.1 Método P4P

O método P4P (*Perspective Four-Points problem*) que é proposto por Kamata *et al* [KETK92], tem como principal vantagem possuir uma única solução. Suponha conhecer a posição de quatro pontos coplanares não alinhados relativos ao referencial $\{R\}$: ${}^R P_0$ a ${}^R P_3$, e os pontos 2D correspondentes na imagem ${}^F P_0$ a ${}^F P_3$. Para calcular a matriz ${}^C T_R$, nós introduzimos dois sistemas de coordenadas intermediários, o sistema

$\{A\}$ e o $\{B\}$ e fazemos ${}^C T_R$ como mostrado na equação 3.30.

$${}^C T_R = {}^C T_B {}^B T_A {}^A T_R \quad (3.30)$$

Portanto, para determinar ${}^C T_R$ devemos calcular as três matrizes ${}^C T_B$, ${}^B T_A$ e ${}^A T_R$.

Cálculo da Matriz ${}^A T_R$

Nós escolhemos o sistema $\{A\}$ para que os pontos notáveis estejam no que chamamos de *standard position*: ${}^R P_0$ está na origem, ${}^R P_1$ está na parte positiva do eixo x e ${}^R P_2$ está no primeiro ou no segundo quadrante. Sejam os ângulos α , β e θ dados por:

$$\theta = \begin{cases} 0 & \text{se } \Delta y_{10} = \Delta x_{10} = 0 \\ \tan^{-1} \left(\frac{\Delta y_{10}}{\Delta x_{10}} \right) & \text{caso contrário} \end{cases} \quad (3.31)$$

$$\beta = \tan^{-1} \left(\frac{-\Delta z_{10}}{\Delta x_{10} c\theta + \Delta y_{10} s\theta} \right) \quad (3.32)$$

$$\alpha = \tan^{-1} \left[\frac{(\Delta x_{20} c\theta + \Delta y_{20} s\theta) s\beta + \Delta z_{20} c\beta}{-\Delta x_{20} s\theta + \Delta y_{20} c\theta} \right] \quad (3.33)$$

Os elementos a_{ij} da matriz ${}^A T_R$ são calculados de acordo com as seguintes equações, onde $s\odot$, $c\odot$ e $\Delta\ominus_{ij}$, $\odot \in (\alpha, \beta, \theta)$, $\ominus \in (x, y, z)$, significam $\sin(\odot)$, $\cos(\odot)$ e ${}^R \ominus_i - {}^R \ominus_j$, respectivamente:

$$a_{11} = c\beta c\theta \quad (3.34)$$

$$a_{12} = c\beta s\theta \quad (3.35)$$

$$a_{13} = -s\beta \quad (3.36)$$

$$a_{14} = -{}^R x_0 c\beta c\theta - {}^R y_0 c\beta s\theta + {}^R z_0 s\beta \quad (3.37)$$

$$a_{21} = -c\alpha s\theta + s\alpha s\beta c\theta \quad (3.38)$$

$$a_{22} = c\alpha c\theta + s\alpha s\beta s\theta \quad (3.39)$$

$$a_{23} = s\alpha c\beta \quad (3.40)$$

$$a_{24} = -{}^R x_0 (-c\alpha s\theta + s\alpha s\beta c\theta) - {}^R y_0 (c\alpha c\theta + s\alpha s\beta s\theta) - {}^R z_0 (s\alpha c\beta) \quad (3.41)$$

$$a_{31} = s\alpha s\theta + c\alpha s\beta c\theta \quad (3.42)$$

$$a_{32} = -s\alpha c\theta + c\alpha s\beta s\theta \quad (3.43)$$

$$a_{33} = c\alpha c\beta \quad (3.44)$$

$$a_{34} = -{}^R x_0(s\alpha s\theta + c\alpha s\beta c\theta) - {}^R y_0(-s\alpha c\theta + c\alpha s\beta s\theta) - {}^R z_0(c\alpha c\beta) \quad (3.45)$$

$$a_{41} = a_{42} = a_{43} = 0 \quad (3.46)$$

$$a_{44} = 1 \quad (3.47)$$

Cálculo da Matriz ${}^C T_B$

O sistema $\{B\}$ é escolhido de tal forma que se alinhe com uma câmera fictícia, a qual tem sua lente centrada na mesma posição que a câmera real, porém é orientada tal que ${}^J P_0$, a nova imagem de ${}^R P_0$, está na origem da imagem e ${}^J P_1$ está na parte positiva do eixo x . Dizemos que esta câmera fictícia se encontra em *ideal position*. Utilizando a notação $s\odot = \sin(\odot)$ e $c\odot = \cos(\odot)$, $\odot \in (\phi, \omega, \rho)$, a matriz ${}^C T_B$ é dada por:

$$\phi = \tan^{-1} \left(\frac{-{}^I x_0}{k} \right) \quad (3.48)$$

$$\omega = \tan^{-1} \left(\frac{{}^I y_0 c\phi}{k} \right) \quad (3.49)$$

$$\rho = \tan^{-1} \left(\frac{{}^I y_1 c\omega + {}^I x_1 s\phi s\omega - k c\phi s\omega}{{}^I x_1 c\phi + k s\phi} \right) \quad (3.50)$$

$${}^C T_B = \begin{bmatrix} c\phi c\rho + s\phi s\omega s\rho & -c\phi s\rho + s\phi s\omega c\rho & s\phi c\omega & 0 \\ c\omega s\rho & c\omega c\rho & -s\omega & 0 \\ -s\phi c\rho + c\phi s\omega s\rho & s\phi s\rho + c\phi s\omega c\rho & c\phi c\omega & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.51)$$

onde k é a distância focal em píxeis.

Cálculo da Matriz ${}^B T_A$

A matriz ${}^B T_A$ representa a solução para um problema no qual os pontos notáveis estão em *standard position* e são observados por uma câmera em *ideal position*. As coordenadas dos pontos notáveis em *standard position*, ${}^A P_i$, são calculadas aplicando-se a transformada ${}^A T_R$ aos pontos notáveis originais, ${}^R P_i$. As coordenadas destes pontos, na imagem, vistos por uma câmera em *ideal position*, ${}^J P_i$, são calculadas aplicando-se a transformada $({}^C T_B)^{-1} [= ({}^C T_B)^T]$ aos pontos 3D $({}^F x_i, {}^F y_i, k)$, obtendo-se os pontos $({}^B x_i, {}^B y_i, {}^B z_i)$, e depois fazendo ${}^J x_i = -k{}^B x_i / {}^B z_i$ e ${}^J y_i = -k{}^B y_i / {}^B z_i$.

Devido à maneira pela qual foram definidos os sistemas $\{A\}$ e $\{B\}$, os parâmetros c_{ij} da matriz ${}^B T_A$ podem ser determinados unicamente. Primeiramente, devemos calcular alguns parâmetros intermediários:

$$b_1 = k({}^A x_2 {}^A y_3 - {}^A x_3 {}^A y_2) \quad (3.52)$$

$$b_2 = {}^B x_2 {}^A x_2 {}^A y_3 - {}^B x_3 {}^A x_3 {}^A y_2 \quad (3.53)$$

$$b_3 = ({}^B x_2 - {}^B x_3) {}^A y_2 {}^A y_3 \quad (3.54)$$

$$b_4 = {}^B x_2 {}^A y_3 - {}^B x_3 {}^A y_2 \quad (3.55)$$

$$b_5 = {}^B y_2 {}^A x_2 {}^A y_3 - {}^B y_3 {}^A x_3 {}^A y_2 \quad (3.56)$$

$$b_6 = ({}^B y_2 - {}^B y_3) {}^A y_2 {}^A y_3 \quad (3.57)$$

$$b_7 = {}^B y_2 {}^A y_3 - {}^B y_3 {}^A y_2 \quad (3.58)$$

$$b_8 = k {}^A x_1 (b_4 b_6 - b_7 b_3) - {}^B x_1 b_1 b_6 \quad (3.59)$$

$$b_9 = {}^B x_1 [{}^A x_1 (b_4 b_6 - b_7 b_3) - (b_2 b_6 - b_5 b_3)] \quad (3.60)$$

Finalmente, os coeficientes são dados por:

$$c_{11} = \begin{cases} +\sqrt{1/[1 + (b_8/b_9)^2]} & \text{se } k/{}^B x_1 \leq b_8/b_9 \\ -\sqrt{1/[1 + (b_8/b_9)^2]} & \text{caso contrário} \end{cases} \quad (3.61)$$

$$c_{31} = -b_8 c_{11} / b_9 \quad c_{34} = -{}^A x_1 (k c_{11} / {}^B x_1 + c_{31}) \quad (3.62)$$

$$c_{12} = \frac{k {}^A x_2 c_{11} + {}^B x_2 {}^A x_2 c_{31} + {}^B x_2 {}^A y_2 c_{32} + {}^B x_2 c_{34}}{-k {}^A y_2} \quad (3.63)$$

$$c_{32} = \begin{cases} -(b_5 c_{31} + b_7 c_{34}) / b_6 & \text{se } b_6 \neq 0 \\ -(b_1 c_{11} + b_2 c_{31} + b_4 c_{34}) / b_3 & \text{caso contrário} \end{cases} \quad (3.64)$$

$$c_{22} = -{}^B y_2 ({}^A x_2 c_{31} + {}^A y_2 c_{32} + c_{34}) / (k {}^A y_2) \quad (3.65)$$

$$c_{13} = -c_{22} c_{31} \quad (3.66)$$

$$c_{23} = c_{12} c_{31} - c_{11} c_{32} \quad (3.67)$$

$$c_{14} = c_{21} = c_{24} = c_{41} = c_{42} = c_{43} = 0 \quad (3.68)$$

$$c_{33} = c_{11} c_{22} \quad (3.69)$$

$$c_{44} = 1 \quad (3.70)$$

3.5 Processamento Digital de Imagens

Para se extrair os pontos notáveis necessários para os algoritmos de calibração de câmera, se faz necessário um processamento digital de imagem. Esse processamento pode ser dividido em duas fases: a extração de pontos para a estimação dos parâmetros intrínsecos e a extração de pontos para a estimação dos parâmetros extrínsecos. A principal diferença entre o processamento digital exigido nos dois processos é com relação ao tempo. Enquanto a estimação dos parâmetros internos é um procedimento *off-line*, ou seja, não possui uma alta restrição de tempo, a estimação dos parâmetros externos deve ser feita em tempo real. Por este motivo, utilizamos diferentes estratégias em cada fase.

3.5.1 Extração dos pontos para a estimação dos parâmetros intrínsecos

Ambas as fases de calibração de câmera exigem que pontos notáveis de um objeto de dimensões conhecidas sejam localizados em uma imagem.

Para a determinação dos parâmetros intrínsecos, devemos localizar na imagem n pontos (são necessários no mínimo 6 pontos para podermos solucionar a equação 3.17, porém, para realizar a otimização, é aconselhado que se utilizem mais pontos). Poderíamos utilizar um padrão plano, onde apareceriam os n pontos de cores características, facilmente localizados na imagem, como mostrado na figura 3.3.

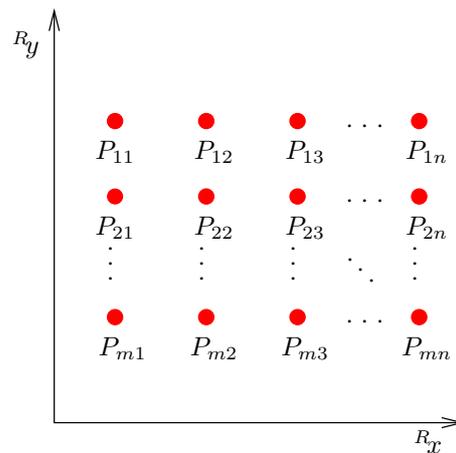


Figura 3.3: Padrão plano de pontos

Porém, este esquema apresenta problemas. Para tornar o sistema mais preciso, devemos imprimir os pontos com o menor raio possível. Isto causa dificuldades na localização do ponto na imagem. Por sua vez, um ponto com o raio grande, apesar de ser facilmente detectado, compromete a precisão. Sabendo que a distância da câmera para os pontos pode variar, fica difícil determinar um raio ótimo, que torne o sistema robusto e preciso. Por este motivo, decidimos utilizar um esquema que, ao invés de localizar diretamente os pontos, localiza retas, e que as interseções de tais retas forneceriam os pontos desejados. Um exemplo é mostrado na figura 3.4, onde a interseção das retas $r1$ e $r2$ fornece o ponto $P1$.

Dessa forma, utilizamos um padrão plano onde apareciam várias retas, de forma que a interseção destas nos daria os n pontos desejados. Um exemplo de tal padrão é mostrado na figura 3.5.

Portanto, para localizar os pontos na imagem, devemos inicialmente localizar as

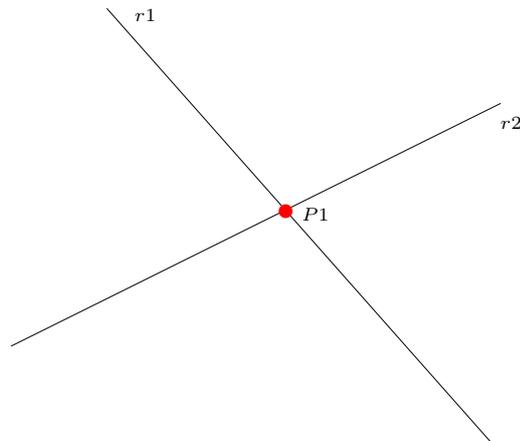


Figura 3.4: Ponto fornecido pela interseção entre duas retas

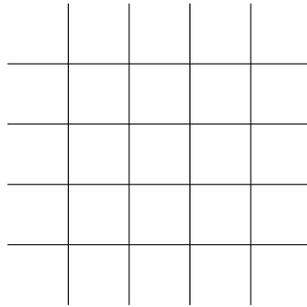


Figura 3.5: Pontos fornecidos pela interseção entre várias retas

retas presentes na imagem. Para tal, utilizaremos o algoritmo de Hough.

Algoritmo de Hough

A transformada de Hough é um método utilizado para detectar, em uma imagem digital, uma classe de formas geométricas conhecidas e que pode ser representada como uma curva paramétrica, como por exemplo, retas, círculos e elipses [Hou62]. Essa transformada mapeia pontos entre o espaço cartesiano e o espaço de parâmetros no qual a curva foi definida. Desta maneira, os pontos que compõem a curva desejada são concentrados no espaço de parâmetros de acordo com as características que unem tais pontos no espaço cartesiano.

Neste trabalho, utilizaremos a transformada de Hough com o objetivo de detectar retas. Assim, a acumulação dos pontos no espaço de parâmetros irá nos informar

os parâmetros que definem a reta que passa por tais pontos no espaço cartesiano. Uma forma de parametrizar uma reta no plano $X - Y$ é pela representação normal, mostrada na equação 3.71:

$$\rho = x \cos \theta + y \sin \theta \quad (3.71)$$

onde os parâmetros ρ e θ representam, respectivamente, o comprimento de um vetor que passa pela origem do plano cartesiano e corta a reta com um ângulo de 90° , e o ângulo formado entre tal vetor e o eixo X (mostrados na figura 3.6).

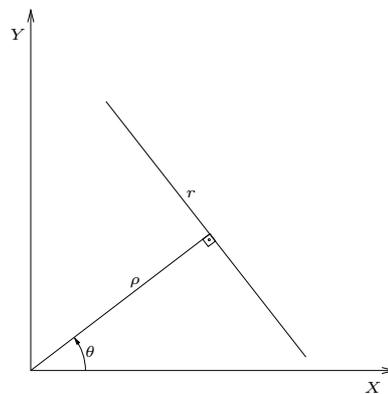


Figura 3.6: Parâmetros de uma representação normal de uma reta r

A transformada de Hough para retas consiste em, para cada ponto classificado como pertencente à reta, calcular os parâmetros de todas as possíveis retas que passam por tal ponto.

Na representação normal, cada ponto no espaço cartesiano irá gerar uma senóide no espaço de parâmetros. Dessa forma, a intercessão destas curvas senoidais irá nos fornecer os parâmetros que identificam uma reta que une os pontos no espaço cartesiano, como mostrado na figura 3.7.

No caso de uma imagem digital, os pontos classificados como pertencentes à reta poderão não estar colineares, devido a ruídos. Desta forma, não será possível encontrar um único ponto de intercessão entre todas as senóides no espaço de parâmetros. Devemos então encontrar o ponto que contém o maior número de senóides, achando

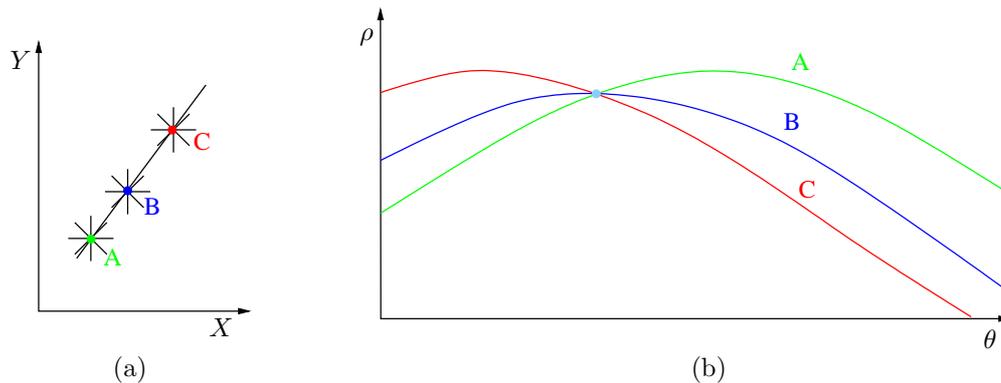


Figura 3.7: (a) Pontos em espaço cartesiano; (b) Transformada de Hough utilizando apresentação normal para os pontos A, B e C e sua interseção

assim a equação da reta que comporta o maior número possível de pontos no espaço cartesiano.

Para a implementação do algoritmo, o espaço de parâmetros é discretizado. Se na discretização poucos valores forem fornecidos, o algoritmo se torna mais rápido, porém também se torna menos preciso. Testes mostraram que para se ter uma precisão aceitável, o algoritmo se torna complexo, exigindo portanto um certo esforço computacional.

3.5.2 Extração dos pontos para a estimação dos parâmetros extrínsecos

Nesta fase faz-se necessário um algoritmo rápido, devido ao seu caráter de tempo real. Dessa forma o algoritmo de Hough se mostra inapropriado. Surgiu então a necessidade de desenvolver um algoritmo de extração de pontos de uma imagem que fosse robusto e nos garantisse uma resposta rápida. O algoritmo desenvolvido teve como idéia básica localizar as arestas de um losango e em seguida, a partir da intercessão de tais retas, calcular seus vértices. O algoritmo será detalhado no capítulo 4.

Capítulo 4

Sistema Proposto

Neste capítulo apresentaremos o sistema de localização do robô humanóide baseado em sensores visuais (câmera CCD). Iremos utilizar, além do robô humanóide (robô mestre), um outro robô móvel, chamado robô escravo (detalhado na seção 4.2), o qual irá processar imagens obtidas do robô mestre, enviando a este os comandos necessários para realizar a sua navegação. A medição da distância relativa entre os dois robôs utilizará as técnicas apresentadas no capítulo 3, enquanto que as técnicas de movimentação dos robôs serão apresentadas neste capítulo.

4.1 Robô Mestre

O robô humanóide adquirido foi o *Robosapien*[Anô05c], mostrado na figura 4.1, da empresa *Wow Wee*. O *Robosapien* foi projetado por Mark Tilden, um projetista robótico que já trabalhou para a NASA, a DARPA e a JPL pelo Los Alamos National Laboratory. Foi ele quem desenvolveu a técnica de robôs biomórficos [LS01] em 1988. O *Robosapien* é um dos primeiros robôs de baixo custo disponível comercialmente, baseado neste princípio, o qual tem como objetivo imitar mecanismos, sensores, estruturas computacionais e metodologias utilizados por animais.

O *Robosapien* possui quatro funções básicas em relação à sua movimentação: andar para frente, andar para trás, girar para a esquerda e girar para a direita. A sua forma de marcha é a marcha estática (seção 2.2.1). Além disso, ele possui



Figura 4.1: Robosapien

funções para levantar e abaixar ambos os braços e para rotacionar o antebraço, que forma um certo ângulo com o braço. Este movimento de rotação do antebraço também controla a abertura da garra. O Robosapien é equipado com um receptor infra-vermelho em sua cabeça, de forma que todos estes comandos, além de outras funções que combinam vários movimentos, são enviados para o robô humanóide por um controle remoto via infra-vermelho (figura 4.2).



Figura 4.2: Controle remoto infra-vermelho

As características do sinal infra-vermelho do Robosapien [Anô5b] são:

- Portadora do sinal: 39.2kHz
- Modulação do sinal com 8 bits
- Taxa de transmissão de 1200Hz

Os comandos básicos que o controle remoto envia são mostrados na tabela 4.1.

CÓDIGO	AÇÃO
\$80	virar para a direita
\$81	levantar braço direito
\$82	girar antebraço direito para fora
\$83	inclinar corpo para a direita
\$84	abaixar braço direito
\$85	girar antebraço direito para dentro
\$86	andar para frente
\$87	andar para trás
\$88	girar para a esquerda
\$89	levantar braço esquerdo
\$8A	girar antebraço esquerdo para fora
\$8B	inclinar corpo para a esquerda
\$8C	abaixar braço esquerdo
\$8D	girar braço direito para dentro
\$8E	parar

Tabela 4.1: Comandos IR básicos do Robosapien

O robô humanóide também é equipado com um conjunto de quatro sensores de toque (um em cada garra e um em cada pé), além de um microfone e um sensor auditivo.

4.2 Robô Escravo

O robô escravo está sendo desenvolvido no laboratório para testes de funcionalidades básicas de robôs móveis da UFRN (figura 4.3). Este irá comandar os movimentos diferenciais de suas rodas, fazendo assim sua locomoção. O robô escravo, além de possuir vários sensores (encoders, por exemplo), será equipado com uma câmera CCD, de forma a obter imagens que serão processadas em seu processador interno. Além disso, o robô escravo terá um sistema de localização absoluta de precisão, baseado em odometria, já que este utiliza rodas, e em imagens de marcos naturais [Bez04].

Na fase final do trabalho, iremos acoplar ao robô escravo um transmissor infravermelho para se comunicar com o robô mestre, de acordo com os códigos exibidos

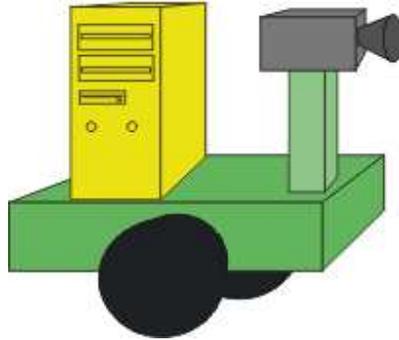


Figura 4.3: Robô escravo

na tabela 4.1.

4.3 Funcionamento do Sistema

Para realizar a navegação do robô mestre, devemos saber qual sua localização atual, de forma a planejar sua trajetória e corrigir eventuais erros. O responsável pela localização do robô mestre será o robô escravo. A partir de imagens obtidas do humanóide por uma câmera localizada no robô escravo, utilizando os métodos de estimação de pose discutidos no capítulo 3, poderemos calcular a posição do robô mestre em relação ao robô escravo. Supondo então conhecida a posição absoluta do robô escravo pode-se calcular a posição absoluta do robô mestre, o que constitui o objetivo deste trabalho.

Para obter os pontos notáveis necessários para se utilizar um dos métodos de estimação de pose, iremos fixar uma marca no robô humanóide de uma cor característica, como mostrado na figura 4.4. Esta marca, que irá possuir forma poligonal (um losango, por exemplo), irá fornecer os pontos necessários a partir de seus vértices (no caso do losango, quatro pontos seriam fornecidos, sugerindo assim o método P4P).

Partindo da hipótese de que a posição absoluta do robô escravo é conhecida, já que também sabemos a posição relativa do robô mestre em relação ao robô escravo,



Figura 4.4: Marca utilizada para estimação de pose do robô humanóide

podemos facilmente obter a posição absoluta do robô mestre. De posse dessa informação, o robô escravo poderá determinar quais os movimentos necessários para que o humanóide siga a trajetória desejada e, após convertê-los em códigos de acordo com a tabela 4.1, enviá-los, via infra-vermelho, para o robô mestre. Vale ressaltar que este trabalho não irá abordar a geração de trajetória para o robô mestre, ou seja, a posição desejada do robô humanóide não será tratada.

Para que tal sistema funcione, se faz necessário que o robô escravo acompanhe o robô mestre. Ele deve fazê-lo mantendo uma distância e uma orientação de forma que facilite a localização da marca contida na imagem e nos forneça informação suficiente para efetuar uma boa estimação da posição do robô mestre, já que algumas poses são melhores que outras, como provou Herve [Her96] matematicamente. Assim, os movimentos do robô escravo serão dependentes dos movimentos do robô mestre. O posicionamento do robô escravo, de forma a maximizar a imagem obtida, deverá ser estudado neste trabalho.

4.4 Localização do Robô Mestre

Para realizar a navegação do robô mestre, temos que ser capazes de localizá-lo no ambiente de trabalho. Portanto, nesta etapa, temos como objetivo determinar a posição do robô mestre em relação ao robô escravo a partir de imagens obtidas por uma câmera acoplada ao robô escravo. Para possibilitar extrair tal informação das imagens, devemos primeiro determinar os parâmetros intrínsecos da câmera que está sendo utilizada (seção 3.3).

4.4.1 Determinação dos Parâmetros Intrínsecos

Para a realização da calibração interna utilizamos o método de Zhang, descrito na seção 3.3.1. Para utilizar tal método, é necessário localizar um conjunto de pontos conhecidos em três ou mais imagens. Esses pontos notáveis foram representados por interseções entre retas. Podemos ver na figura 4.5 uma imagem do padrão utilizado, o qual possui 18 retas (nove retas verticais e nove retas horizontais) e um total de 81 pontos de interseção. As retas paralelas estão separadas por uma distância de 2 cm.

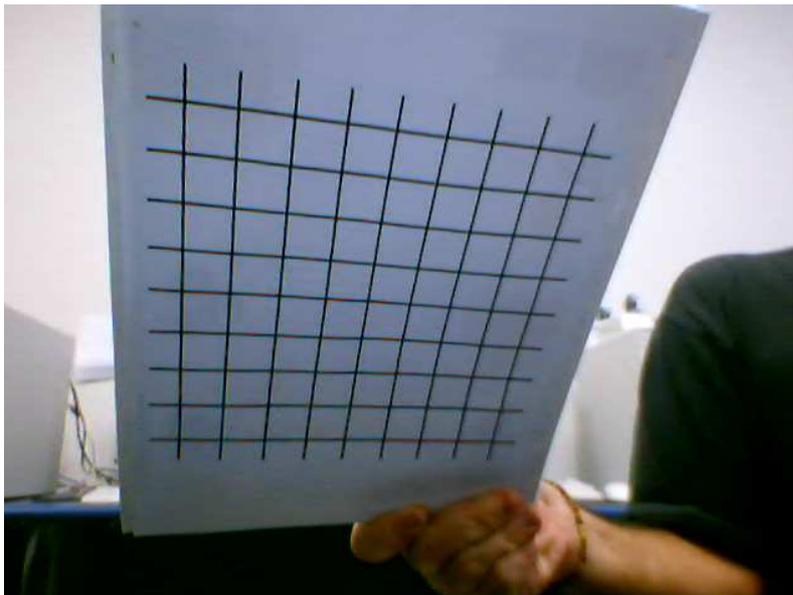


Figura 4.5: Padrão utilizado na calibração interna da câmera

A localização de tais pontos na imagem foi realizada utilizando a estratégia de determinar inicialmente o conjunto de retas presente na imagem e em seguida calcular os pontos de interseção entre as retas, como discutido na seção 3.5.1.

Porém, na aplicação da transformada de Hough a este problema, estamos interessados em calcular não apenas uma única reta, e sim um conjunto de retas. Desta forma, devemos então encontrar não apenas um ponto de interseção entre as senóides, e sim n destes pontos (no caso do padrão da figura 4.5, por exemplo, $n = 81$). Devemos porém tomar um certo cuidado na hora de classificar tais pontos de interseção. Por ser uma imagem digital, as senóides irão se interceptar em vários pontos. O ponto de interseção com o maior número de senóides seria, portanto, a reta que melhor representa o conjunto de pontos. Se determinarmos os n melhores pontos de interseção, poderemos estar escolhendo retas muito parecidas (próximas no espaço de parâmetros) que passam pelo mesmo objeto. Isto é mostrado na figura 4.6.

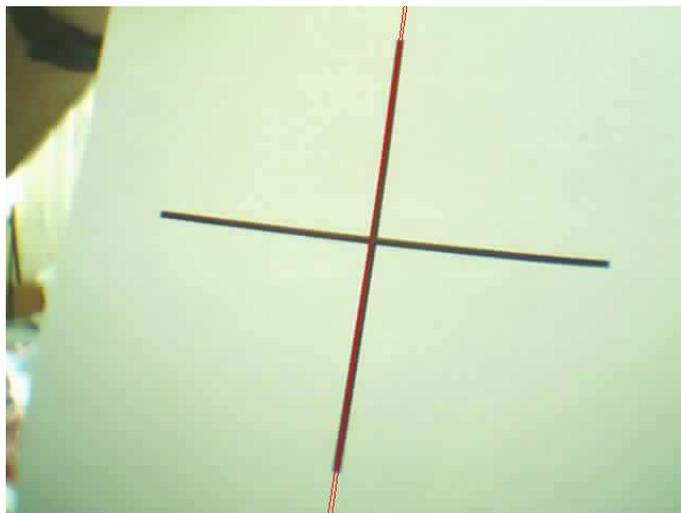


Figura 4.6: Transformada de Hough para determinação de duas retas sem distanciamento

As duas retas determinadas passam pela mesma linha (linha vertical). Isto acontece porque mesmo a segunda melhor reta passando pela linha vertical ainda comporta mais pontos do que a melhor reta passando pela linha horizontal. Isto nos indica que as retas determinadas pela transformada devem ser suficientemente diferentes umas das outras para que possam representar todos os objetos. Ou seja, ao determinar os pontos máximos de interseção no espaço de parâmetros, estes devem ter uma certa distância entre si, garantindo assim a diferença entre as retas determinadas. Com a utilização deste método, obtivemos o resultado da figura 4.7. Perceba que as duas retas foram determinadas corretamente.

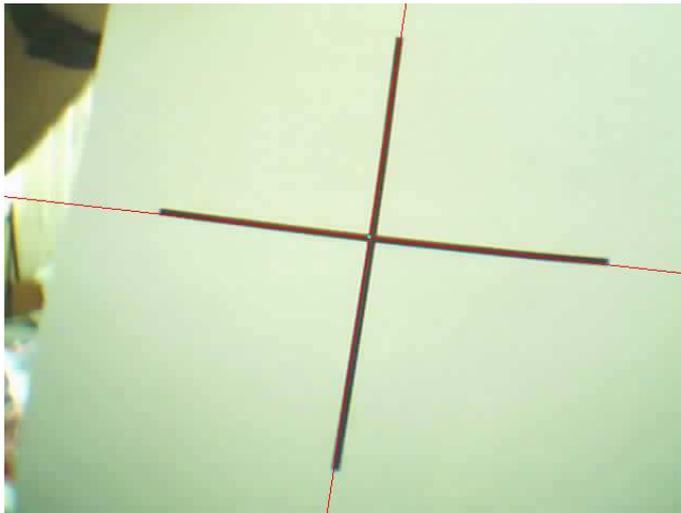


Figura 4.7: Transformada de Hough para determinação de duas retas com distanciamento

Aplicando a transformada de Hough, desta maneira, a várias imagens do padrão, mostrado na figura 4.5, em poses diferentes, podemos determinar as várias retas do padrão presentes em cada imagem. A partir das interseções de tais retas podemos obter os pontos notáveis em cada imagem, e assim aplicar o método descrito na seção 3.3.1. Os resultados obtidos por este método serão mostrados no capítulo 5.

4.4.2 Determinação dos Parâmetros Extrínsecos

Para realizar a determinação dos parâmetros extrínsecos, é necessário determinar na imagem um conjunto de pontos notáveis. A principal diferença da calibração de câmera interna é que neste caso isto deve ser realizado em tempo real e várias vezes por segundo.

Poderíamos ter utilizado o método P3P, o qual exige apenas três pontos notáveis, como sugerido por Nogueira [NMA04]. Porém, este método tem como desvantagem fornecer múltiplas soluções, como discutido no capítulo 3. Por isso, optamos por utilizar o método P4P de Kamata, que é indicado para aplicações em tempo real por ter solução única e não empregar métodos iterativos, sendo a solução dada por uma expressão analítica.

Assim como na determinação dos pontos para a calibração interna, utilizamos a estratégia de localizar a interseção entre retas para determinar um ponto. No método P4P são necessários 4 pontos, ou seja, a interseção entre quatro retas. Isto nos sugere utilizar as arestas de um quadrilátero (neste caso um losango), o qual foi afixado no robô humanóide, como mostrado na Figura 4.8. A interseção das retas que formam suas arestas nos fornecerá os vértices do mesmo. Poderíamos ter utilizado novamente a transformada de Hough para determinar as arestas; contudo, esta se mostra inapropriada devido ao custo computacional, já que a calibração externa é executada a cada amostragem e exige rapidez. Tivemos portanto que desenvolver um sistema de detecção de arestas mais rápido ainda não presente na literatura.

Determinação dos vértices de um losango

Supondo que o losango possua uma cor característica, podemos determinar se um dado ponto pertence ou não à região interna do losango. Para uma melhor detecção de tais pontos, colocamos o losango sobre um fundo de cor contrastante com a cor do losango, como pode ser observado na figura 4.8. Dessa maneira, é possível

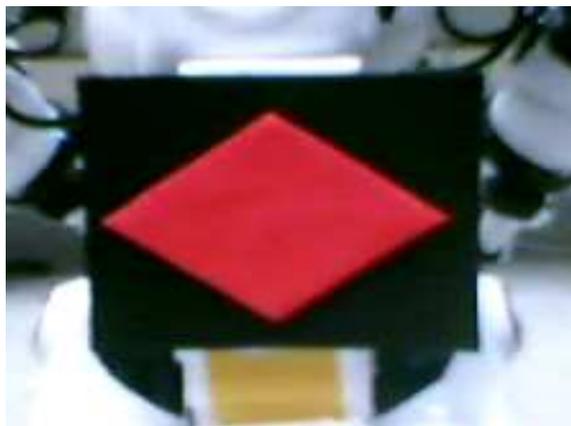


Figura 4.8: Losango utilizado na calibração externa

detectar pontos que estão sobre cada uma das arestas deste. Contudo, devido a vários fatores (iluminação, reflexão do material que compõe o losango, ruído, etc), os pontos determinados podem não estar exatamente sobre suas arestas, apresentando um erro. A idéia então é utilizar um conjunto de pontos para, a partir destes, determinar a melhor reta que representa cada aresta, minimizando assim o erro do processo.

Assim, primeiramente, a partir de um ponto pertencente ao losango (P_0), determinamos um conjunto de pontos pertencentes à suas arestas superiores (U_i) e inferiores (D_i). Tal procedimento é apresentado na Tabela 4.2 e ilustrado na figura 4.9.

Determinamos então duas retas a partir dos pontos U_i : L_L^U , que vai da esquerda para o centro; e L_R^U , que vai da direita para o centro. De maneira similar, determinamos as retas L_L^D e L_R^D a partir dos pontos D_i . Tais retas são calculadas de forma a minimizar o somatório do quadrado das distâncias entre a reta e os pontos a ela associados. A maneira como cada ponto é associado a uma das retas é descrita na Tabela 4.3. Calculando a interseção entre estas 4 retas, obtemos os vértices do losango. Uma vez que temos os vértices do losango calculados no passo atual, podemos calcular o seu centróide e utilizar tal ponto como o ponto inicial pertencente

ao losango (P_0) para o passo seguinte.

A determinação da cor (média e faixa de variação) do losango é feita dinamicamente, de forma a proporcionar uma maior imunidade a variações na iluminação do ambiente. Dado que no passo anterior conseguimos determinar com sucesso os vértices do losango, podemos amostrar um certo número de píxeis internos e calcular a média (E) e o desvio padrão (σ) de cada componente de cor no padrão RGB. Dessa forma, podemos calcular os novos valores limites de cada componente de cor. Seja V_i o limite inferior e V_s o limite superior de uma certa componente, que podem ser calculados pelas equações 4.1 e 4.2, onde $k \in \mathbf{N}$.

$$V_i = E - k\sigma \quad (4.1)$$

$$V_s = E + k\sigma \quad (4.2)$$

Tabela 4.2: Detecção dos pontos pertencentes às arestas do losango

<ol style="list-style-type: none"> 1. Procure um ponto (pixel) inicial P_0 na imagem que pertença ao losango // Busca a direita 2. $k \leftarrow 0$; $M \leftarrow -1$ 3. Enquanto o pixel P_k pertencer ao losango: <ol style="list-style-type: none"> (a) A partir de P_k, busque para cima e para baixo pelo último ponto pertencente ao losango (pontos U_k e D_k). (b) Se $\ U_k - D_k\ < \text{MIN_LEN}$ Pare Enquanto (c) Calcule $P_{k+1} \leftarrow \frac{U_k - D_k}{2} + [\Delta x \ 0]^T$ (d) $M \leftarrow k$; $k \leftarrow k + 1$ // Busca a esquerda 4. $k \leftarrow 0$; $N \leftarrow +1$ 5. Enquanto o pixel P_k pertencer ao losango: <ol style="list-style-type: none"> (a) A partir de P_k, busque U_k e D_k. (b) Se $\ U_k - D_k\ < \text{MIN_LEN}$ Pare Enquanto (c) Calcule $P_{k-1} \leftarrow \frac{U_k - D_k}{2} - [\Delta x \ 0]^T$ (d) $N \leftarrow k$; $k \leftarrow k - 1$
--

De posse dos pontos que representam os vértices do losango, podemos utilizar o método P4P de Kamata para obter a matriz de transformação ${}^C T_R$. Porém, a

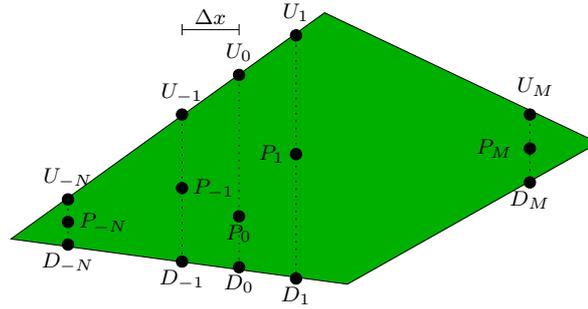


Figura 4.9: Detecção de pontos pertencentes às arestas do losango

Tabela 4.3: Detecção das arestas do losango

<pre>// Parte de cima 1. Inicie a reta L_L^U determinada pelos pontos U_{-N} e $U_{-(N-1)}$, e a reta L_R^U, a partir de U_M e U_{M-1}. 2. $i \leftarrow -(N-2)$ 3. Enquanto $i \leq M-2$: (a) Calcule as distâncias e_L^U e e_R^U, que são as distâncias entre o ponto U_i e L_L^U e L_R^U, respectivamente. (b) Se $e_L^U < e_R^U$, recalcule L_L^U, considerando que U_i pertence a ela; senão, recalcule L_R^U. (c) $i \leftarrow i+1$ // Parte de baixo 4. Realize um procedimento similar (passos 1 a 3) para determinar L_L^D e L_R^D a partir dos pontos D_i.</pre>

matriz de rotação calculada pelo P4P de Kamata pode não representar exatamente uma matriz de rotação, a qual deve ser ortonormal. Para conferir tal propriedade à matriz de rotação R , realizamos o seguinte procedimento:

- Decompomos a matriz em seus valores singulares [Leo98]:

$$R = UDV^T \quad (4.3)$$

- Forçamos os seus valores singulares a serem todos iguais a 1:

$$D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

- Recalculamos R a partir de sua decomposição modificada.

Alguns resultados da aplicação deste método serão apresentados no capítulo 5.

4.5 Movimentação do Robô Mestre

Devido às restrições de movimento e à dificuldade de um controle preciso do deslocamento do humanóide, decidimos utilizar uma técnica de movimentação heurística.

O primeiro passo a fazer seria, dado uma posição inicial e uma posição final do humanóide, gerar uma trajetória paramétrica para o humanóide. Neste trabalho, supomos que um nível hierárquico superior gerou esta trajetória, de tal forma que sabemos, a cada instante, a posição onde se deseja que o robô mestre esteja. Esta posição é conhecida como posição referência (P_F).

O algoritmo de controle do robô mestre seguirá os seguintes passos:

1. Calcula-se:
 - (a) o erro e entre a posição atual do robô mestre e a posição de referência P_F (distância euclidiana)
 - (b) o ângulo α entre a reta definida pela posição atual do humanóide e a posição de referência e o eixo z do robô mestre (z_R) (Figura 4.10).
2. Se α for maior que um certo valor v , comanda-se o robô mestre para parar e girar até que α esteja próximo de 0.
3. Se e for maior que um certo valor l , o humanóide é comandado para andar para a frente.

Os comandos são enviados para o robô mestre através de sinais infra-vermelhos. Isto é feito utilizando o programa WinLIRC [PB02], versão para o sistema operacional Windows do programa LIRC [SB99], o qual envia comandos infra-vermelhos através da porta serial. Tais comandos para o Robosapien são descritos em um arquivo de configuração [Anô5a]. Para transmitir os sinais enviados pela porta

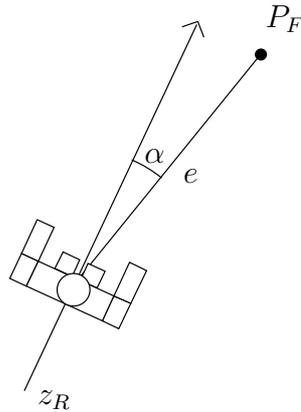


Figura 4.10: Movimentação do humanóide

serial, foi desenvolvido um circuito emissor de sinais infra-vermelhos, cujo esquema é mostrado na figura 4.11.

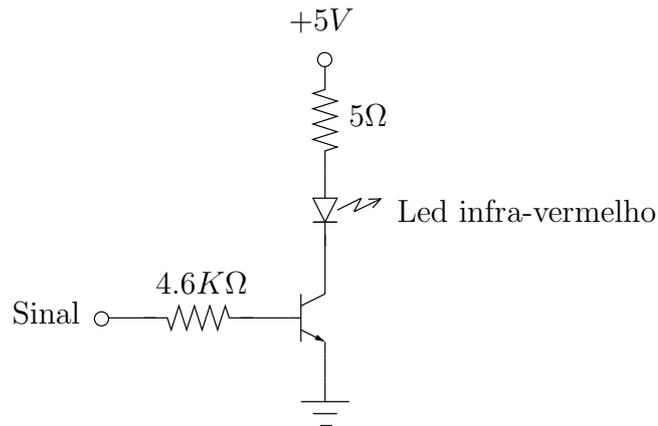


Figura 4.11: Circuito transmissor infra-vermelho

4.6 Movimentação do Robô Escravo

A movimentação da câmera depende totalmente da movimentação do humanóide, pois esta deve acompanhar o humanóide garantindo sempre a obtenção de uma boa imagem do mesmo. Para tal temos que, a cada passo da amostragem, comandar a câmera para uma posição que esteja a uma distância d do humanóide, e de forma que a câmera aponte em sua direção (o centróide do losango se localize no centro da imagem). Além disso, devemos garantir que a câmera capture imagens que nos

forneça informação suficiente para efetuar uma boa estimaco da posio do robo mestre. Para isto devemos monitorar o ângulo θ , formado entre o eixo x do robo mestre (x_R) e o eixo z da cmera (z_C) (Figura 4.12). Caso θ se torne menor que um ângulo crtico θ_c , o que poderia causar uma m deteco dos vrtices do losango, o robo humanide   comandado para parar, e a cmera   comandada para a posio a uma distncia d do humanide e de forma que θ seja 90° (posio P_D na Figura 4.12). Isto causa uma no continuidade no movimento, por m, se a trajet ria for suave o suficiente, o movimento ser cont nuo.

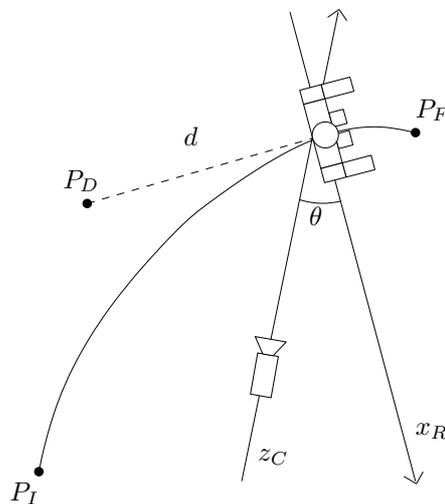


Figura 4.12: Situao de ângulo crtico na movimentaco da cmera

O algoritmo de movimentaco do robo escravo seguir os seguintes passos:

1. O robo escravo (e conseqentemente a cmera)   comandado para girar at  que o centr ide do losango esteja no centro da imagem
2. O robo escravo   comandado para andar para frente at  que esteja a uma distncia d do humanide
3. Calcula-se o ângulo θ
4. Se θ for menor que um valor crtico θ_c , o robo humanide   comandado para parar e o robo escravo   comandado para a posio P_D .

Um caso especial que deve ser observado   quando a posio destino calculada

para a câmera causa uma colisão desta com um objeto do ambiente. Neste caso devemos sacrificar temporariamente a visibilidade do humanóide e posicionar a câmera o mais próxima possível da posição destino sem causar colisão. O humanóide deve ser reposicionado de forma a eliminar o problema.

Para determinar a distância d a qual a câmera deve permanecer do humanóide e o ângulo crítico θ , iremos estudar a relação entre o erro das medições e a posição da câmera. Podemos afirmar que uma boa posição da câmera é uma que garanta que a razão r entre a área do losango e a área total da imagem seja sempre a máxima possível, fornecendo assim uma imagem de boa qualidade para a extração dos vértices do losango. Sabemos que quanto menor a distância d , maior será r . Desse forma essa distância deve ser a menor possível, porém sem prejudicar outros aspectos, os quais serão comentados na seção 5.4.1.

Capítulo 5

Experimentos e Resultados

A seguir iremos apresentar alguns experimentos realizados para verificar os métodos propostos e os resultados obtidos.

5.1 Calibração dos Parâmetros Intrínsecos

Avaliamos as duas etapas da calibração interna da câmera: a detecção dos pontos notáveis e a aplicação do método de Zhang. A câmera utilizada foi uma câmera *web* Creative modelo WebCam NX Ultra.

5.1.1 Detecção dos Pontos Notáveis

O padrão utilizado para a detecção dos pontos notáveis é o mesmo mostrado na figura 4.5: 18 retas, sendo 9 verticais e 9 horizontais, separadas por 2 cm de distância.

O resultado do algoritmo de Hough (seção 3.5.1) aplicado a quatro imagens obtidas pela câmera pode ser observado na figura 5.1.

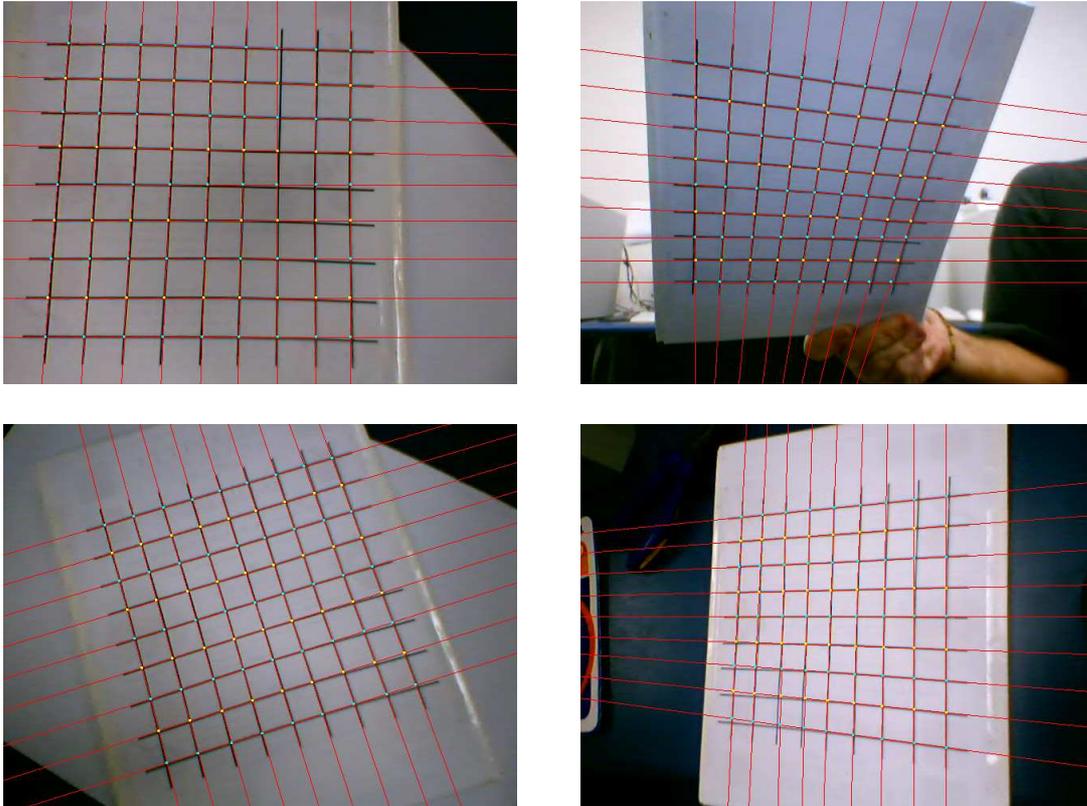


Figura 5.1: Resultado da aplicação do algoritmo de Hough a quatro imagens do padrão utilizado na calibração interna da câmera

5.1.2 Método de Zhang

O método de Zhang (seção 3.3.1) aplicado às quatro imagens vistas na figura 5.1 nos forneceu o seguinte resultado:

$$\text{Matriz de parâmetros intrínsecos: } A = \begin{bmatrix} 482.386 & -3.08362 & 325.729 \\ 0 & 482.314 & 216.476 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{Coeficientes de distorção: } \begin{aligned} k_1 &= 0.12802 \\ k_2 &= -0.2284 \end{aligned}$$

A validação dos resultados obtidos para a matriz de parâmetros intrínsecos será feita na seção 5.2.2.

O resultado dos coeficientes de distorção calculados pode ser observado na figura 5.2, onde removemos a distorção de uma imagem a partir dos coeficientes obtidos.

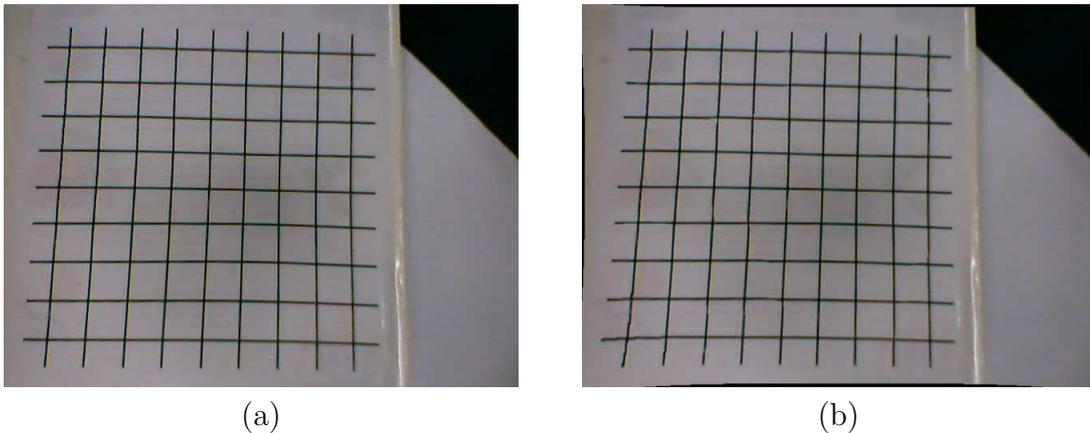


Figura 5.2: (a) Imagem original distorcida (b) Imagem com distorção removida a partir dos coeficientes calculados

5.2 Calibração Externa

Nesta etapa avaliamos o método proposto para a detecção dos vértices de um losango e o método P4P de Kamata.

5.2.1 Detecção dos Vértices do Losango

O resultado do algoritmo de detecção dos vértices do losango pode ser observado na figura 5.3, onde aplicamos o método a várias imagens do robô mestre em várias poses diferentes.

5.2.2 Método P4P

A seguir iremos mostrar os resultados obtidos pelo método P4P de Kamata e avaliar tanto este como os parâmetros intrínsecos obtidos pelo método de Zhang.

Uma maneira de validar a calibração interna e externa ao mesmo tempo seria colocar um determinado objeto em uma posição e orientação conhecidas com relação ao sistema de coordenadas da câmera e, a partir de uma imagem deste, calcular a sua pose. Comparando a pose calculada com a pose verdadeira, obteríamos o erro. Porém, este método de validação se torna difícil pois o centro de coordenadas do referencial da câmera se encontra em seu ponto focal, um ponto inatingível e

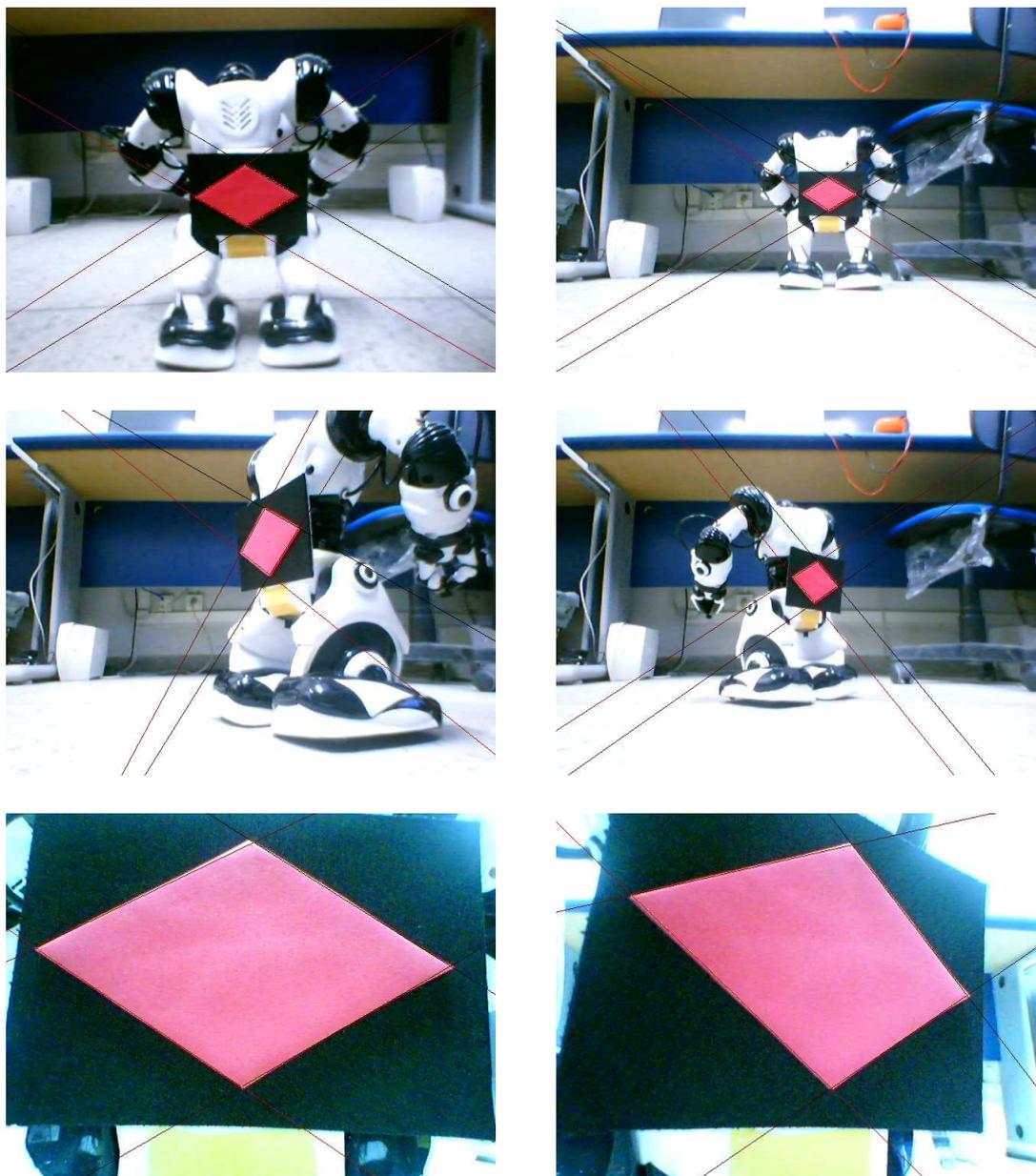


Figura 5.3: Resultado da aplicação do algoritmo de detecção dos vértices do losango aplicado a seis imagens diferentes

imensurável por estar no interior da câmera.

Por este motivo, resolvemos utilizar outra técnica. Esta consiste em adquirir uma imagem de vários objetos cujas posições são conhecidas com relação ao sistema de coordenadas do mundo R . Em seguida, fornecendo ao método P4P as coordenadas de cada objeto, calculamos a matriz de transformação ${}^C T_R$, entre o sistema de coordenadas da câmera e R . Perceba que, para cada objeto, a matriz de transformação deveria ser a mesma. Dessa forma, podemos calcular o erro obtido pelo método. Apesar de não nos garantir a corretude da matriz ${}^C T_R$, este processo nos garante a coerência entre as medições.

As imagens utilizadas neste experimento podem ser observadas nas figuras 5.4 e 5.5. O referencial do sistema de coordenadas R está no canto inferior esquerdo do padrão (figura 5.4), que contém quatro losangos, cujos vértices estão localizados nas posições $(x, y, z)^T$ mostradas na tabela 5.1.

	Vértice 1	Vértice 2	Vértice 3	Vértice 4
Losango 1	(10,7,0)	(5,4,0)	(10,1,0)	(15,4,0)
Losango 2	(14,15,0)	(9,12,0)	(14,9,0)	(19,12,0)
Losango 3	(5,20,0)	(0,17,0)	(5,14,0)	(10,17,0)
Losango 4	(13,25,0)	(8,22,0)	(13,19,0)	(18,22,0)

Tabela 5.1: Vértices dos losangos no referencial R

As matrizes ${}^C T_R$ obtidas para os losangos 1, 2, 3 e 4 da figura 5.4 podem ser observadas nas equações 5.1, 5.2, 5.3 e 5.4 respectivamente.

$${}^C T_R = \begin{bmatrix} 0,995 & 0,015 & 0,091 & -9,808 \\ 0,018 & -1 & -0,03 & 18,059 \\ -0,092 & -0,031 & 0,995 & 36,87 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

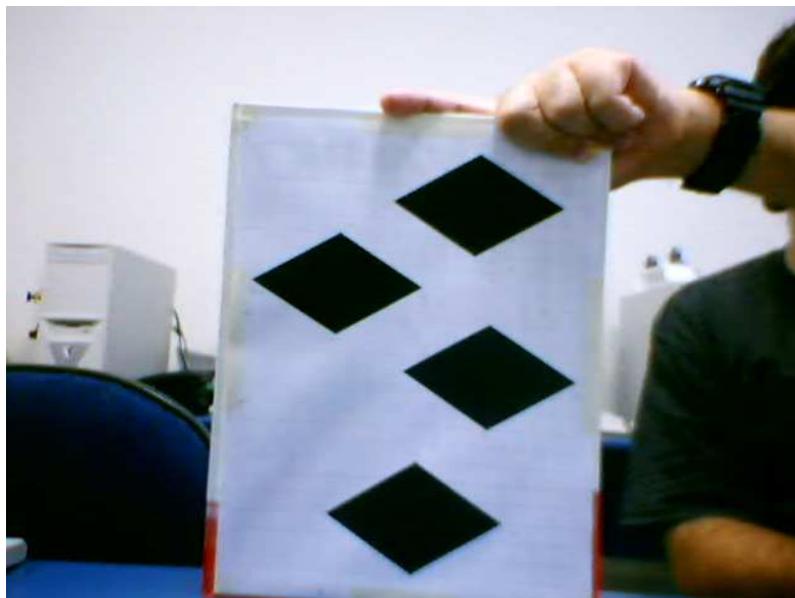


Figura 5.4: Imagem dos losangos utilizada no primeiro experimento de estimação do erro da calibração de câmera

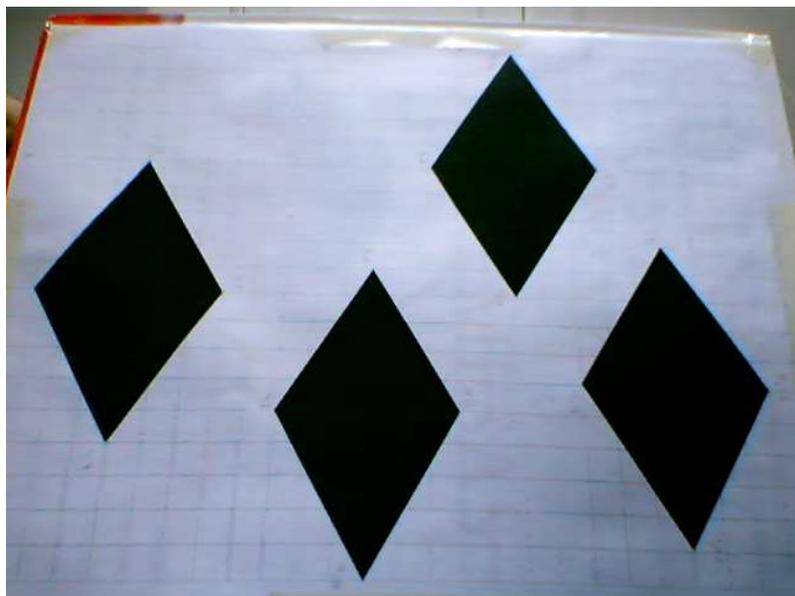


Figura 5.5: Imagem dos losangos utilizada no segundo experimento de estimação do erro da calibração de câmera

$${}^c T_R = \begin{bmatrix} 0,997 & 0,044 & 0,048 & -9,838 \\ 0,041 & -0,998 & 0,062 & 17,428 \\ -0,052 & 0,06 & 0,996 & 38,114 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

$${}^c T_R = \begin{bmatrix} 0,989 & 0,039 & -0,138 & -10,308 \\ 0,036 & -1 & -0,025 & 17,235 \\ 0,137 & -0,02 & 0,99 & 37,532 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.3)$$

$${}^c T_R = \begin{bmatrix} 0,995 & 0,043 & 0,078 & -9,93 \\ 0,046 & -0,999 & -0,027 & 16,962 \\ -0,078 & -0,03 & 0,996 & 39,515 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

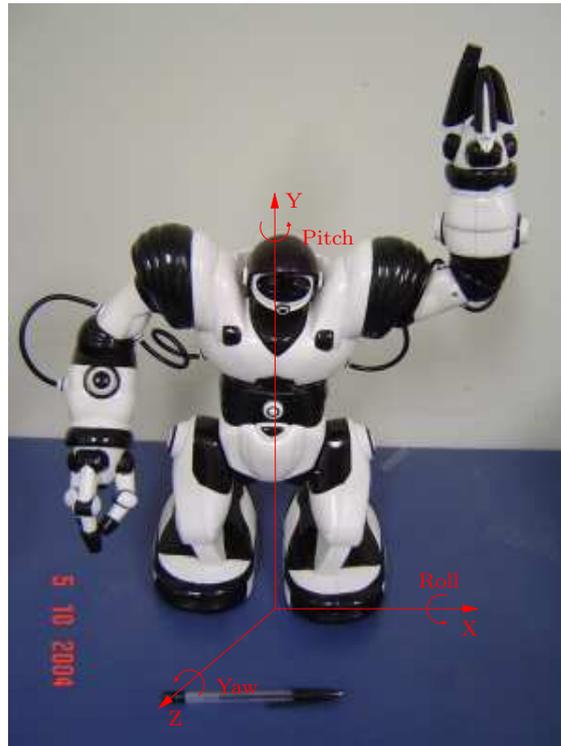


Figura 5.6: Orientação do referencial do robô humanóide e seus ângulos de rotação de acordo com os ângulos Roll, Pitch e Yaw

A tabela 5.2 mostra os valores calculados e os erros obtidos (em relação à média) para cada losango da figura 5.4, em cada um dos eixo, em centímetros, e em cada

ângulo de rotação (de acordo com os ângulos Roll, Pitch e Yaw, mostrados na figura 5.6), em graus.

	Losango 1	Losango 2	Losango 3	Losango 4	Erro médio
X	-9,808	-9,838	-10,308	-9,93	0,265
Y	18,059	17,428	17,235	16,962	0,580
Z	36,87	38,114	37,532	39,515	1,419
Roll	-1,682	3,583	-1,433	-1,5	2,643
Pitch	-5,253	-2,782	7,878	-4,524	6,855
Yaw	0,91	2,536	2,261	2,525	0,857

Tabela 5.2: Erros da calibração de câmera para a figura 5.4

Repetindo o experimento com a imagem da figura 5.5, obtivemos os valores e erros mostrados na tabela 5.3.

	Losango 1	Losango 2	Losango 3	Losango 4	Erro médio
X	10,03	10,04	10,16	10,22	0,11
Y	-8,76	-7,86	-7,25	-7,24	0,86
Z	24,61	25,83	23,5	24,48	1,18
Roll	-21,72	-23,59	-23,26	19,3	2,4
Pitch	3,53	3,69	1,89	1,35	1,45
Yaw	89,7	88,3	87,6	87,27	1,33

Tabela 5.3: Erros da calibração de câmera para a figura 5.5

Perceba a diminuição, em geral, dos erros obtidos entre os resultados da figura 5.4 e a figura 5.5. Isto acontece em razão de os losangos da figura 5.5 estarem mais perto da câmera e aparecerem maiores na imagem. Porém, mesmo para a figura 5.4 os erros obtidos são pequenos. Os erros nos eixos X e Y e no ângulo Roll são insignificantes. O erro no eixo Z e nos ângulos Pitch e Yaw são maiores, porém ainda são aceitáveis dado à natureza do problema proposto.

5.3 Localização e Movimentação do Robô Mestre

O movimento que o humanoíde em estudo realiza para se locomover não é um movimento suave. Por utilizar a marcha estática, o humanoíde deve pender de um

lado para o outro de forma a colocar seu centro de massa no pé de apoio a cada passada. Além disso, para girar sobre seu próprio eixo, o humanóide utiliza uma espécie de movimento de quadril. Por isso, a marca afixada no humanóide está sujeita a movimentos bruscos. Isto, associado a uma variação de iluminação no ambiente, causa um ruído no cálculo da localização do humanóide.

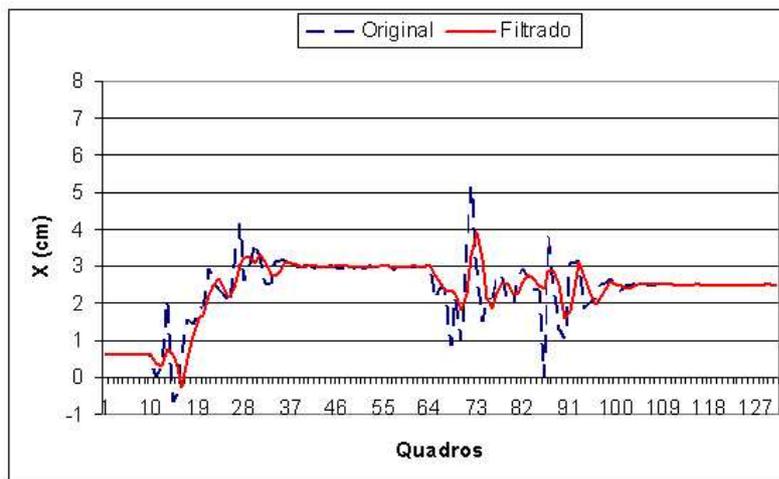
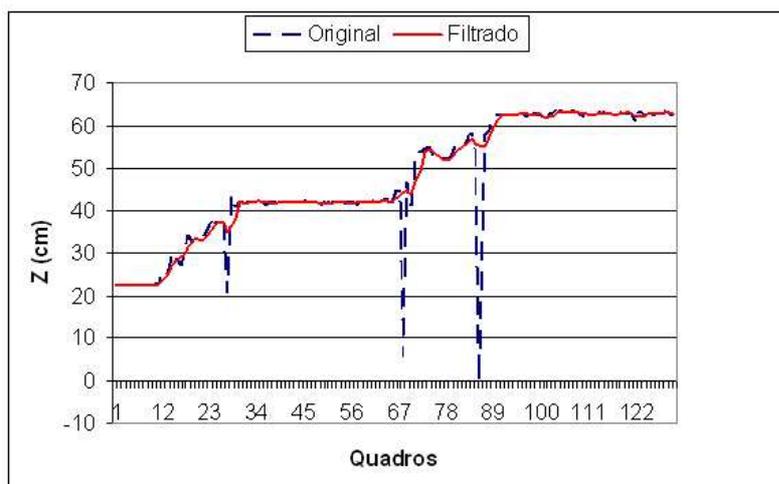
Para solucionar este problema, utilizamos um filtro passa baixa, o qual tem como objetivo retirar componentes de alta frequência, diminuindo assim o ruído. A fórmula do filtro utilizado, conhecido como média móvel [Hay98], é mostrada na equação 5.5, onde x_n é a n -ésima amostra do sinal original e y_n é a n -ésima amostra do sinal filtrado.

$$y_n = \sum_{k=-l}^0 \frac{1}{l} x_{n+k} \quad (5.5)$$

Além disso, acrescentamos ao filtro um algoritmo de detecção de erros no cálculo da pose do robô mestre. Caso a pose calculada no quadro atual seja muito diferente da pose calculada no quadro anterior, esta é descartada. Isso é possível em razão de os movimentos do humanóide serem lentos em relação à quantidade de quadros processados por segundo (em torno de 5 quadros por segundo).

Aplicando o filtro a um sinal que mostra a posição x, Z e o ângulo Pitch do robô mestre, em movimento, em função do quadro processado pelo algoritmo, e fazendo $l = 3$, obtivemos o resultado mostrado nas figuras 5.7, 5.8 e 5.9. Os gráficos em azul (linha tracejada) mostram o sinal sem nenhum tratamento, enquanto os gráficos em vermelho (linha sólida) mostra o sinal após ser filtrado por um filtro passa baixa.

Para mostrar a impossibilidade de realizar odometria no humanóide, realizamos o seguinte experimento: este foi comandado para andar para frente 10 passos. Este experimento foi repetido duas vezes, partindo da mesma posição e orientação inicial. A figuras 5.10, 5.11 e 5.12 mostram as posições x e z e o ângulo Pitch, respectivamente, para cada experimento realizado.

Figura 5.7: Gráfico da posição x do robô mestre em função do quadroFigura 5.8: Gráfico da posição z do robô mestre em função do quadro

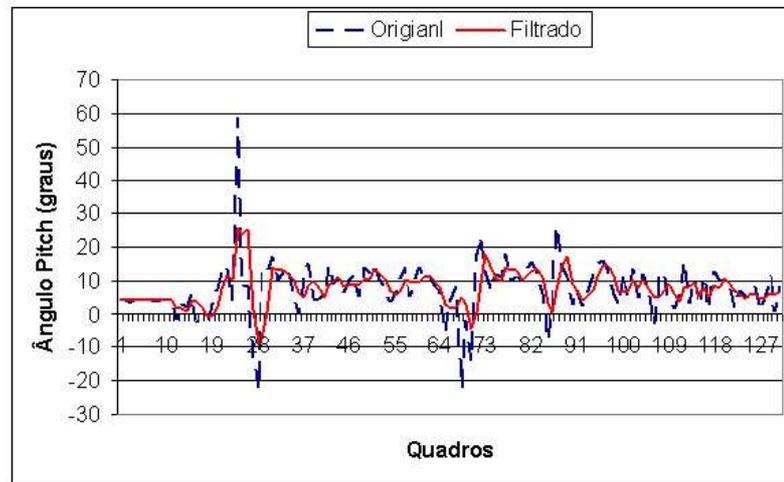


Figura 5.9: Gráfico do ângulo Pitch do robô mestre em função do quadro

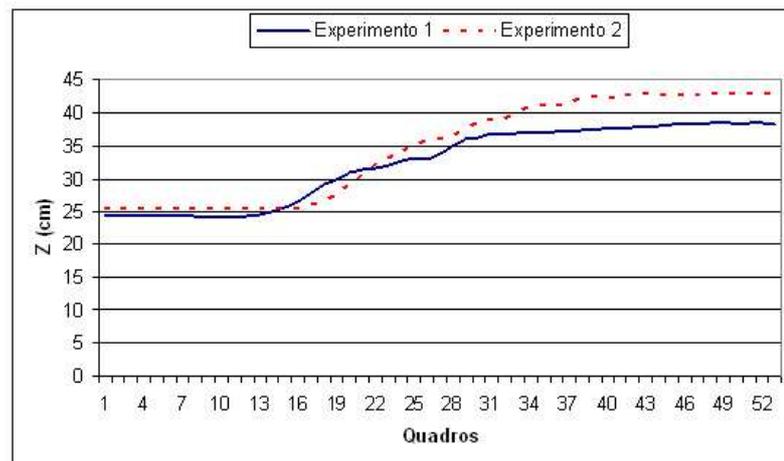


Figura 5.10: Posição z para os dois experimentos

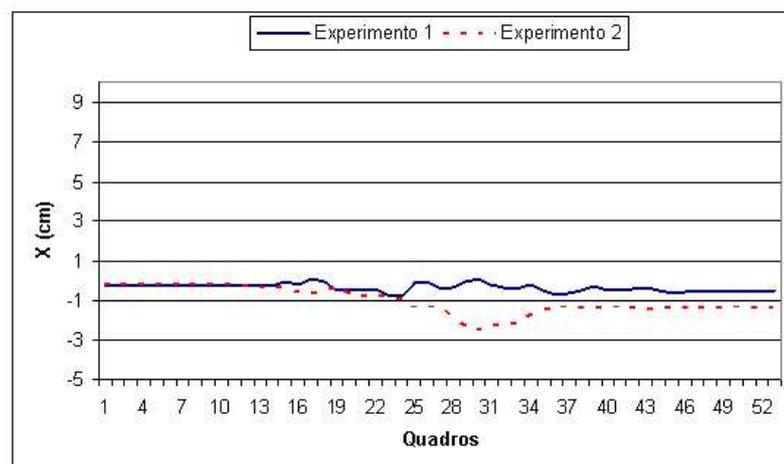


Figura 5.11: Posição x para os dois experimentos

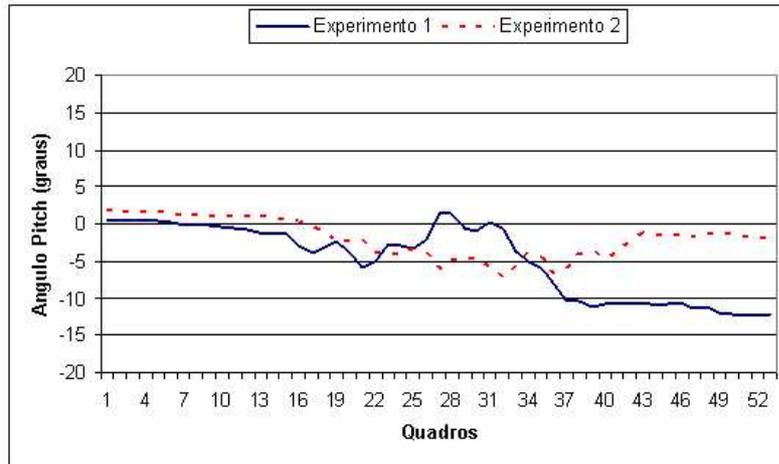


Figura 5.12: Ângulo Pitch para os dois experimentos

Observando as figuras 5.10, 5.11 e 5.12, podemos verificar a diferença na posição final entre os experimentos. No plano XZ a diferença foi de 1,3 cm, e no ângulo Pitch foi de 10,1 graus.

A seguir, mostramos o resultado do método heurístico de movimentação do robô mestre. Neste experimento, o humanóide foi comandado para a posição $(x, y, z) = (0, 4.7, 60)$ a partir da posição $(0, 4.7, 29)$ e Pitch=0. Já que o robô escravo se encontra em processo de construção, a movimentação da câmera teve de ser feita manualmente. Cada vez que a distância entre a câmera e o robô mestre superasse 40 cm, a câmera era comandada para uma posição a 25 cm do humanóide.

O resultado do experimento pode ser observado nas figuras 5.13, 5.14 e 5.15. A posição final atingida pelo robô humanóide foi $(3.3, 4.8, 58.2)$.

Vale salientar que, ao comandar o robô mestre para uma determinada posição P_F , consideramos que este atingiu tal posição se a diferença entre a sua posição atual e P_F for menor que um certo valor l (seção 4.5). Neste experimento $l = 4\text{cm}$. Este valor pode parecer alto, porém ele é justificado pelo tipo de movimento realizado pelo humanóide, já que o movimento mínimo realizado por ele é um passo (que tem aproximadamente 2 cm) ou, até mesmo, dois passos antes que este responda ao

comando infra-vermelho.



Figura 5.13: Posição z do robô mestre em relação à câmera



Figura 5.14: Posição z da câmera em relação ao mundo

5.4 Movimentação do Robô Escravo

Nesta seção, iremos estudar os valores ideais para a distância d , a qual o robô escravo deve ficar do robô mestre, e ângulo crítico θ_c .

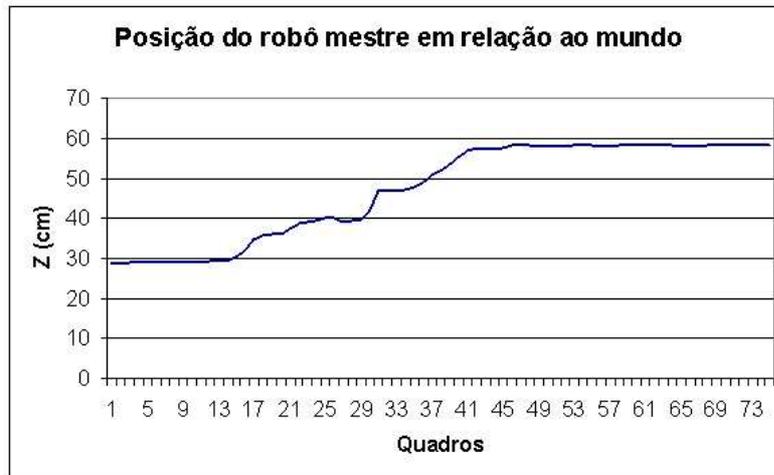


Figura 5.15: Posição z do robô mestre em relação ao mundo

5.4.1 Determinação da Distância d e do Ângulo Crítico θ_c

Como comprovado na seção 5.2.2, quanto maior o losango aparece na imagem, melhor será o resultado da estimação de pose do robô humanóide. Isto nos indica que o losango deve estar o mais próximo da câmera (de forma a ser visualizado com o maior tamanho possível na imagem). Porém, isto causa alguns problemas. Estar muito perto da câmera, aumenta a probabilidade de o losango (ou uma parte deste) sair da imagem, impossibilitando a detecção de seus vértices e, conseqüentemente, a estimação de sua pose. Além disso, a câmera deve permanecer a uma certa distância do robô mestre, de forma a permitir que este realize seus movimentos no ambiente de trabalho, sem bater na câmera. Podemos dizer ainda que, quanto mais perto da câmera, mais bruscos são os movimentos realizados pelo losango na imagem, causando um embaçamento (*blurring*) na imagem, como mostrado na figura 5.16. Por estes motivos, a distância d deve ser estudada de forma a minimizar tais erros.

Nas figuras 5.17 e 5.18, vemos o desvio padrão entre várias medições realizadas consecutivamente no cálculo da posição z e no cálculo do ângulo Pitch em função da razão r entre a área do losango e a área da figura, respectivamente. Podemos observar que o desvio padrão permanece quase inalterado quando r é maior que

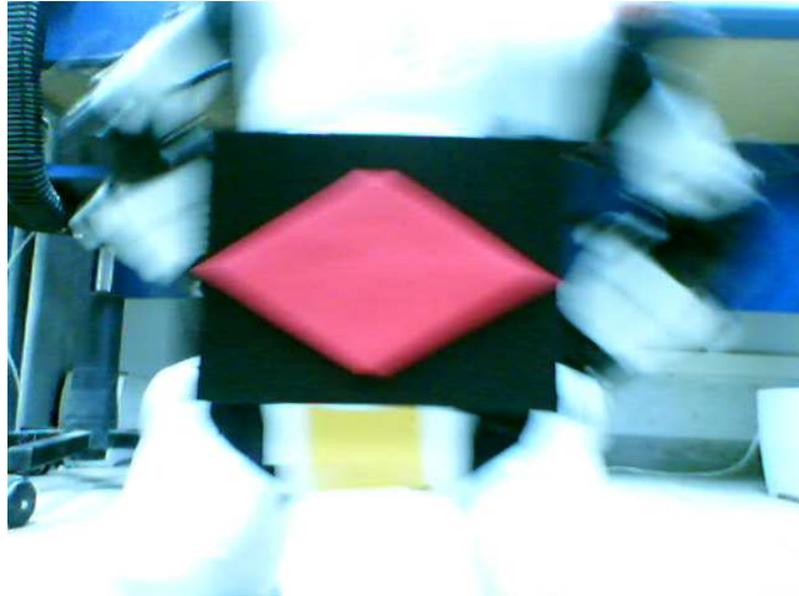


Figura 5.16: Losango muito próximo da câmera causando movimentos bruscos e embaçamento da imagem

15%, e sofre pequenas alterações quando r está entre 5% e 15%. Porém, quando r se torna menor que 5%, observamos um grande aumento no desvio padrão. Isto nos indica que a razão limite que não prejudica a estimação de pose é em torno de 5%.

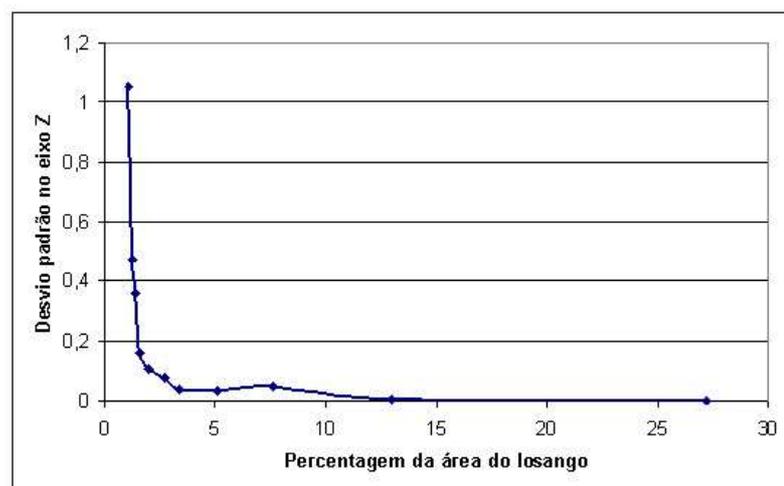


Figura 5.17: Desvio padrão na estimação da posição z em função da razão r

Observando a figura 5.19, que mostra a razão r em função da distância do robô mestre em relação à câmera (com Pitch=0), podemos concluir que a distância limite

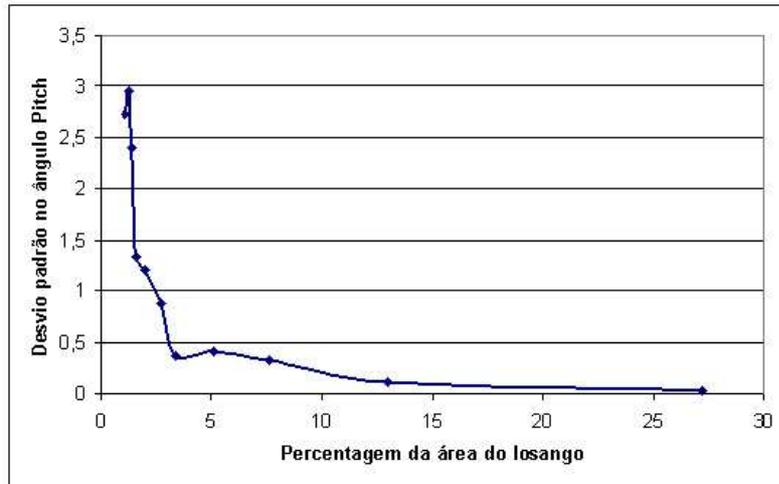


Figura 5.18: Desvio padrão na estimativa do ângulo Pitch em função da razão r

é em torno de 25 cm.

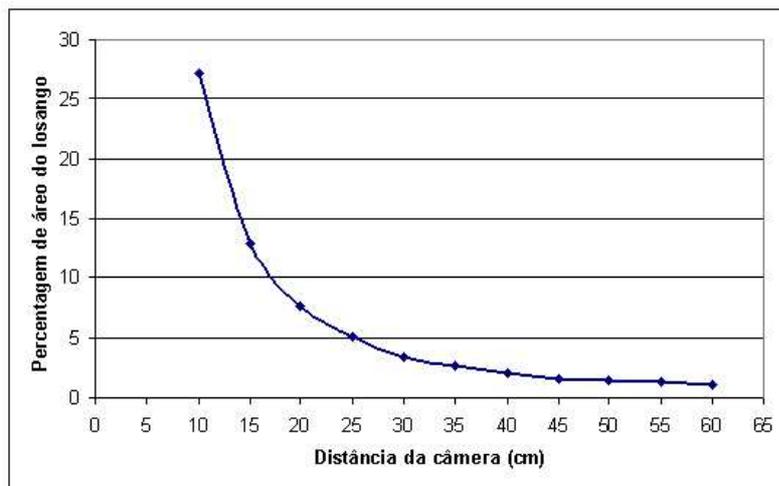


Figura 5.19: Relação entre a distância do robô mestre para a câmera e a razão entre a área do losango e a área da imagem

Para determinar o ângulo crítico θ_c , estudamos o desvio entre várias medições consecutivas na estimativa da pose do humanóide a uma distância $d = 25$ cm e variando o ângulo Pitch. O resultado deste experimento pode ser observado nas figuras 5.20 e 5.21.

Observe que o desvio aumenta bruscamente quando o ângulo Pitch é maior que 40 graus. Já que o ângulo Pitch e θ são complementares (seção 4.6), temos que o

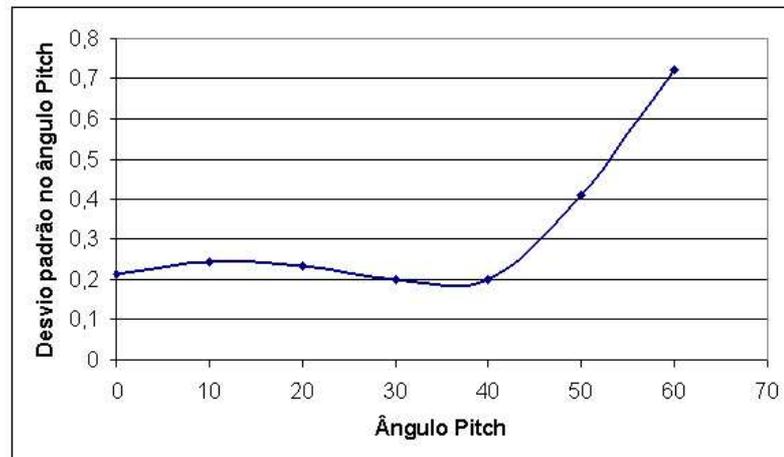


Figura 5.20: Relação entre o ângulo Pitch e o desvio padrão na estimação da pose no ângulo Pitch

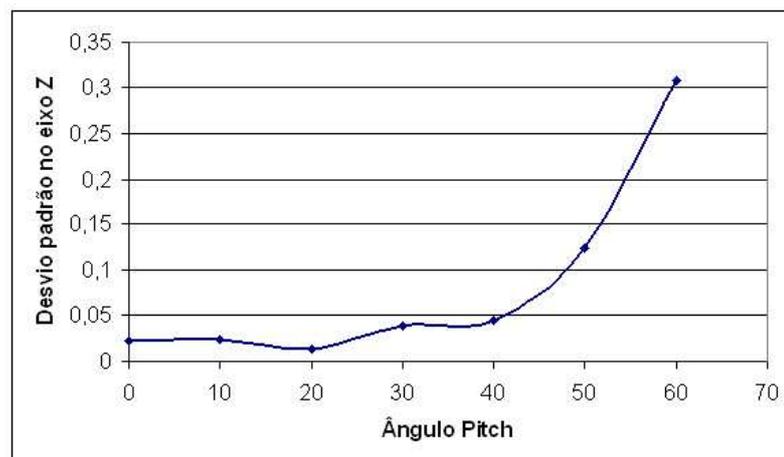


Figura 5.21: Relação entre o ângulo Pitch e o desvio padrão na estimação da pose no eixo z

ângulo crítico $\theta_c = 50$ graus.

5.4.2 Posicionamento do Robô Escravo

Os testes de posicionamento do robô escravo apresentados nesta seção mostram apenas a posição calculada, para a qual ele deveria ser comandado, já que o robô escravo está em fase de construção.

Supondo que o sistema de coordenadas da câmera se encontra na origem do sistema de coordenadas do robô escravo, a posição calculada deste durante um experimento pode ser observada nas figuras 5.22 e 5.23.

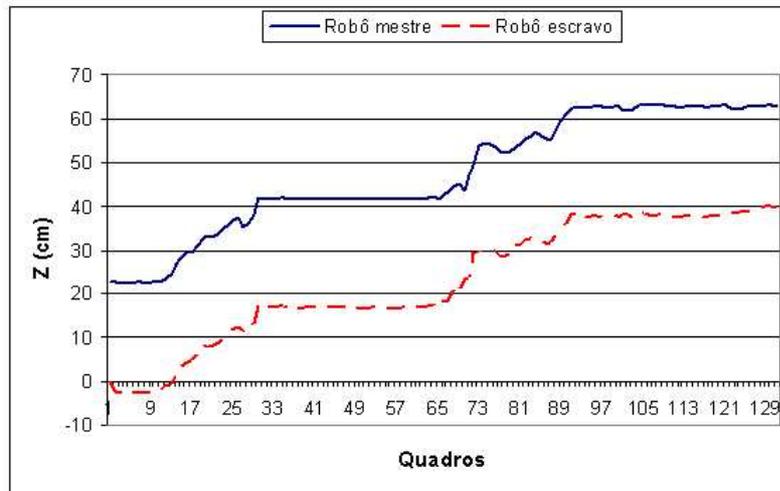


Figura 5.22: Posição do robô mestre e do robô escravo no eixo Z

Na figura 5.24 podemos ver um exemplo de um pose do robô mestre além do ângulo crítico. Neste exemplo, a matriz de transformação entre humanóide e câmera pode ser observada na equação 5.6. O ângulo Pitch foi calculado com sendo 42,6 graus, ou seja, $\theta = 48$. A posição calculada para a câmera foi $(x, y, z) = (15, 0, 5.73)$, supondo que esta se encontrava na origem do sistema de coordenadas do mundo R .

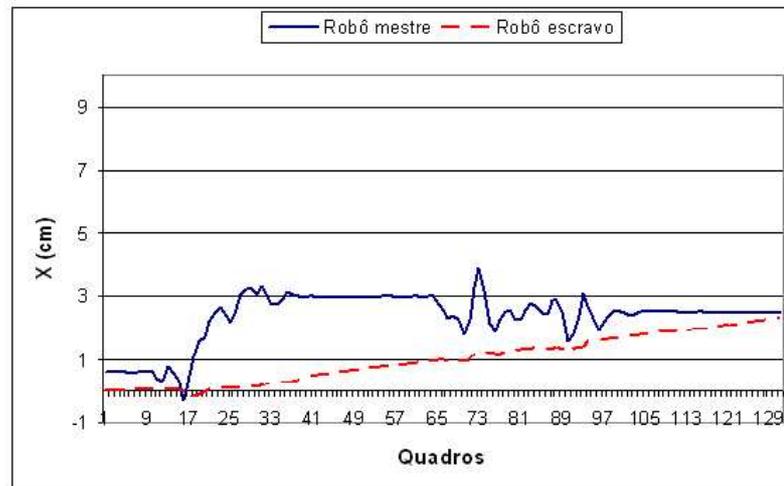


Figura 5.23: Posição do robô mestre e do robô escravo no eixo X



Figura 5.24: Posição onde situação de ângulo crítico foi detectada

$$c_{T_R} = \begin{bmatrix} 0,701 & -0,075 & 0,708 & -1,633 \\ -0,223 & -0,968 & 0,117 & -4,782 \\ -0,677 & 0,24 & 0,695 & 24,3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

Capítulo 6

Conclusões e Perspectivas

De maneira geral, o sistema apresentado se mostrou eficaz para fazer o controle de um robô humanóide de movimentos limitados e que não apresenta nenhum sensor interno, mesmo com a utilização de uma câmera *web* de baixa qualidade.

Na calibração dos parâmetros intrínsecos, foram usados o algoritmo de Hough e o método de Zhang:

- O algoritmo de Hough se mostrou muito preciso na detecção das retas do padrão de calibração interna, fornecendo assim pontos suficientemente precisos para realizar tal fase de forma satisfatória. Contudo, foi observado que a maneira de detectar os pontos no padrão não é a mais adequada para considerar a distorção radial. O algoritmo de Hough, da maneira que foi utilizado, calcula a melhor reta que representa a linha (reta distorcida) presente na imagem. Conseqüentemente a distorção presente na imagem não é totalmente levada em conta. Sugerimos, para um trabalho futuro, representar cada linha usando uma outra função ou vários segmentos de reta, de forma a considerar melhor a distorção radial presente na imagem.
- O método de calibração interna de Zhang se mostrou apropriado ao problema, garantindo uma calibração flexível, robusta e rápida (não no sentido de custo computacional, mas sim no sentido de preparação do padrão e do ambiente para a calibração).

O erro total obtido na calibração de câmera foi pequeno, dada a natureza do problema abordado neste trabalho.

Na calibração dos parâmetros extrínsecos foram utilizados o algoritmo proposto para detecção de arestas e o método P4P de Kamata:

- A detecção dos vértices do losango se mostrou rápida e precisa. Capturando imagens com a resolução de 640×480 o algoritmo obteve uma velocidade de, aproximadamente, cinco quadros por segundo em um Pentium IV 1.8 GHz (em um Pentium III 700 Hz a velocidade foi de dois quadros por segundo). Além de rápido, o método se mostrou robusto, imune a pequenas variações de iluminação no ambiente e com um erro relativamente pequeno. Porém, se o ambiente de trabalho for mal iluminado ou sofre uma variação de iluminação brusca, a detecção dos vértices do losango falha. Para um trabalho futuro poder-se-ia utilizar uma fonte de iluminação direta, instalada no robô escravo, ao lado da câmera.
- Por possuir uma solução analítica, o método P4P de Kamata nos garante uma resposta em tempo constante, sendo ideal para aplicações em tempo real. A desvantagem deste método é não nos fornecer uma matriz de rotação ortogonal, o que requer a sua ortogonalização pela decomposição em valores singulares.

O controle heurístico do robô mestre se mostrou apropriado para o tipo limitado de locomoção realizado pelo humanóide. Caso fosse utilizado um humanóide capaz de realizar movimentos mais finos, certamente uma estratégia de movimentação mais elaborada poderia ser proposta.

O circuito infra-vermelho elaborado não se mostrou muito eficaz, tendo alcance e visada limitados. Para melhor desempenho, sugere-se a aquisição e utilização de um emissor infravermelho disponível no mercado para ser conectado ao computador embarcado no robô escravo.

Foi feito um estudo da melhor estratégia de posicionamento do robô escravo para captação das imagens do robô mestre de forma a minimizar o erro da estimação de pose do humanóide. A etapa seguinte deste trabalho, já em desenvolvimento, será a conclusão e integração do robô escravo ao sistema, possibilitando uma maior gama de testes e aplicações. Melhores análises deste método e do controle heurístico do robô escravo só poderão ser feitas quando o sistema estiver completo. Também poderá ser feito um planejamento conjunto de trajetórias entre o robô humanóide e o robô móvel considerando obstáculos presentes no ambiente.

A aplicação desse sistema se dá em situações onde, além do humanóide, está disponível um outro robô móvel, equipado com uma câmera e com um sistema de localização preciso. Os principais ambientes onde são satisfeitos estes requisitos são laboratórios de pesquisa, onde muitas vezes já se dispõe de robôs móveis mas não de robôs humanóides. O sistema proposto permite que experiências sejam feitas utilizando robôs humanóides de eletrônica embarcada simples e de muito baixo custo.

Referências Bibliográficas

- [AC04] M. E. Albert and J.H. Connell. Visual rotation detection and estimation for mobile robot navigation. In *ICRA - IEEE International Conference on Robotics and Automation*, volume 5, pages 4247–4252, Abril 2004.
- [Anô05a] Autor Anônimo. Robosapien-lirc interface. <http://www.andrew.cmu.edu/user/ebuehl/robosapien-lirc/>, 2005. Visitada em 05 de dezembro de 2005.
- [Anô05b] Autor Anônimo. The unofficial robosapien hacks and mods site. <http://www.robosapien.tk>, 2005. Visitada em 05 de dezembro de 2005.
- [Anô05c] Autor Anônimo. Wow wee. <http://www.robosapienonline.com>, 2005. Visitada em 05 de dezembro de 2005.
- [Bez04] Clauber G. Bezerra. Um sistema de localização para um robô móvel baseado em odometria e marcos naturais. Master's thesis, UFRN, Natal-RN, 2004.
- [Cra86] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Addison-Wesley Publishing Company, 1986.
- [DB93] F. Du and M. Brady. Self calibration of the intrinsic parameters of cameras for active vision systems. *CVPR - IEEE Conference on Computer Vision and Pattern Recognition*, pages 477–482, 1993.
- [DD95] D. Dementhon and L. Davis. Model-based object pose in 25 lines of code. *International Journal of Computer Vision*, 15:123–141, 1995.

- [DRLR88] M. Dhome, M. Richetin, J. T. Lapreste, and G. Rives. Determination of the attitude of 3D objects from a single perspective view. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1265 – 1278, Dezembro 1988.
- [DZLF94] R. Deriche, Z. Zhang, Q. T. Luong, and O. D. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. *European Conference on Computer Vision*, 1994.
- [Fau93] O Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
- [FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [FLM92] O. D. Faugeras, Q. T. Luong, and S. J. Maybank. Camera self-calibration: Theory and experiments. *European Conference on Computer Vision*, pages 321–334, Maio 1992.
- [FM89] J. G. Fryer and S. O. Mason. Rapid lens calibration of a video camera. *Photogramm. Eng. Remote Sens.*, 55:437–442, 1989.
- [Fuk81] I. Fukui. TV image processing to determine the position of a robot vehicle. *IAPR - International Conference on Pattern Recognition*, 14(1-6):101–109, 1981.
- [GS03] Pierre Gurdjos and Peter Sturm. Methods and geometry for plane-based self-calibration. In *CVPR - IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 491–496, Madison, WI, USA, June 2003.
- [Hay98] Monson H. Hayes. *Schaum's Outline of Digital Signal Processing*. McGraw-Hill, 1998.

- [HCLL89] Radu Horaud, Bernard Conio, Olivier Leboulleux, and Bernard Lacolle. An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics and Image Processing*, 47:33–44, 1989.
- [Her96] J.-Y. Herve. Hand/eye coordination: role of the active observer. In *IAPR - International Conference on Pattern Recognition*, volume 1, pages 292–296, 1996.
- [HLNT01] Q. Huang, K. Li, Y. Nakamura, and K. Tanie. Analysis of physical capability of a biped humanoid: Walking speed and actuator specifications. In *IROS - IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, Maui, HI, Outubro 2001.
- [Hor87] R. Horaud. New methods for matching 3-D objects with single perspective views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987.
- [Hou62] P. V. C. Hough. *Methods and means for recognizing complex patterns*. U. S. Patent 3069654, 1962.
- [HSH05] P. Hammarstedt, P. Sturm, and A. Heyden. Degenerate cases and closed-form solutions for camera calibration with one-dimensional objects. In *IEEE International Conference on Computer Vision*, volume 1, pages 317–324, Outubro 2005.
- [Hug86] R. C. Hughes. Enhanced single camera photogrammetry algorithms for real-time control applications. In *Proc. Int. Soc. Photogrammetry and remote sensing*, 1986.
- [IKKI95] M. Inaba, F. Kanehiro, S. Kagami, and H. Inoue. Two-armed bipedal robot that can walk, roll over and stand up. In *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 297 – 302, Agosto 1995.
- [JCC⁺05] Kyungmin Jeong, Youngsoo Choi, Jaiwan Cho, Sung uk Lee, and Seungho Kim. Camera calibration and measurement using circular

- grooves inside pipes. In *12th International Conference on Advanced Robotics*, pages 635–639, 2005.
- [KETK92] Sei-ichiro Kamata, Richard O. Eason, Masafumi Tsuji, and Eiji Kawaguchi. A camera calibration using 4 point-targets. In *IAPR - International Conference on Pattern Recognition*, Hague, Netherlands, September 1992.
- [KGK05] Jun-Sik Kim, P. Gurdjos, and In-So Kweon. Geometric and algebraic constraints of projected concentric circles and their applications to camera calibration. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 27, pages 637–642, Abril 2005.
- [KTII99] F. Kanehiro, Y. Tamiya, M. Inaba, and H. Inoue. Developmental methodology for building whole body humanoid system. In *IROS - IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 1210 – 1215, 1999.
- [Kul] P. Kulkarni. Humanoid robot: Design, control and trajectory planning. Department of Mechanical Engineering, Indian Institute of Technology, Kanpur.
- [Leo98] Steve J. Leon. *Álgebra Linear com Aplicações*. Prentice Hall, 1998.
- [LH81] H. C. Longueit-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Setembro 1981.
- [Low91] D. G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 441–450, 1991.
- [LS01] M. A. Lewis and L. S. Simó. Certain principles of biomorphic robots. *Autonomous Robots*, 2001.

- [MA84] M. J. Magee and J. K. Aggarwal. Determining the position of a robot using a single calibration object. In *ICRA - IEEE International Conference on Robotics and Automation*, Março 1984.
- [MF92] S. J. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, pages 123–152, Agosto 1992.
- [NMA04] M. B. Nogueira, A. A. D. Medeiros, and P. J. Alsina. Development of a fast visual sensor of 3D position for a manipulator robot. In *Jornada de Robótica Inteligente*, 2004.
- [NSNI92] Yoshihiko Nomura, Michihiro Sagara, Hiroshi Naruse, and Atsushi Ide. Simple calibration algorithm for high-distortion-lens camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(11):1095–1099, November 1992.
- [PB02] Jim Paris and Scott Baily. Winlirc. <http://winlirc.sourceforge.net/>, 2002. Visitada em 05 de dezembro de 2005.
- [PPKA81] H. F. L. Pinkney, C. I. Perratt, V. Kratky, and A. A. Ayad. On the application of automatic, real-time single camera photogrammetry to derive the relative spatial orientation and position of objects in machine tasks: A conceptual outline and preliminary evaluation. *NRCC/NAE Laboratory Technical Report, LTR-ST-1007*, Agosto 1981.
- [QC00] P. Quick and D.; Capson. Analysis of determining camera position via Karhunen-Loeve transform. In *Image Analysis and Interpretation*, pages 88–92, 2000.
- [QL99] Long Quan and Zhong-Dan Lan. Linear n-point camera pose determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):774–780, August 1999.
- [RBPD81] P. Rives, P. Bouthémy, B. Prasada, and E. Dubois. Recovering the orientation and the position of a rigid body in space from a single view.

- Technical Report, INRS-Télécommunications, 3, place du commerce, Ile-des-Soeurs, 1981.*
- [RSL05] Srikumar Ramalingam, P. Sturm, and S. K. Lodha. Towards complete generic camera calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1093–1098, Junho 2005.
- [RVSS05] C. Ricolfe-Viala and A. Sanchez-Salmeron. Improving accuracy and confidence interval of camera parameters estimated with a planar pattern. In *IEEE International Conference on Image Processing*, volume 2, pages 1142–1145, Setembro 2005.
- [SB99] Karsten Scheibler and Christoph Bartelmus. Lirc. <http://www.lirc.org/>, 1999. Visitada em 05 de dezembro de 2005.
- [Sil01] Luciano Rottava Silva. Análise e programação de robôs móveis autônomos da plataforma Eyebot. Master's thesis, Universidade Federal de Santa Catarina, Departamento de Engenharia Elétrica, Março 2001.
- [SQ00] P.F. Sturm and L. Quan. Camera calibration and relative pose estimation from gravity. In *IAPR - International Conference on Pattern Recognition*, volume 1, pages 72–75, 2000.
- [Tan03] K. Tanie. Humanoide robot and its application possibility. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems*, 2003.
- [Tsa87] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shell TV cameras and lenses. *IEEE Journal of Robotics and Automation*, pages 323–344, Agosto 1987.

- [WM94] G. W. Wei and S. Ma. Implicit and explicit camera calibration: theory and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 469–480, 1994.
- [WT91] L. Wang and W. Tsai. Camera calibration by vanishing lines for 3-D computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):370 – 376, Abril 1991.
- [WWHB96] W.J. Wilson, C.C. Williams Hulls, and G.S. Bell. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation*, 12:684–696, Outubro 1996.
- [Yua89] J.S.C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Transactions on Robotics and Automation*, 5(2):129–142, 1989.
- [YZ05] Xianghua Ying and Hongbin Zha. Linear approaches to camera calibration from sphere images or active intrinsic calibration using vanishing points. In *IEEE International Conference on Computer Vision*, volume 1, pages 596–603, Outubro 2005.
- [Zha98] Zhengyou Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, 1998.
- [Zha00] Z Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, Novembro 2000.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)