

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

Implementação de Rotinas Computacionais Básicas para Cabeça Estéreo Robótica

Leonardo Campos do Amaral Bezerra

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Dezembro, 2005.

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

Implementação de Rotinas Computacionais Básicas para Cabeça Estéreo Robótica

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 (M.Sc.).

Leonardo Campos do Amaral Bezerra

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

3 de junho de 2006

Resumo

O Laboratório de Sistemas Inteligentes do Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte - UFRN - tem como um de seus projetos de pesquisa - Robosense - a construção de uma plataforma robótica móvel. Trata-se de um robô provido de duas rodas, acionadas de forma diferencial, dois braços, com 5 graus de liberdade cada, um cinturão de sonares e uma cabeça estéreo. Como objetivo principal do projeto Robosense, o robô deverá ser capaz de navegar por todo o prédio do LECA, desviando de obstáculos. O sistema de navegação do robô, responsável pela geração e seguimento de rotas, atuará em malha fechada. Ou seja, sensores serão utilizados pelo sistema com o intuito de informar ao robô a sua pose atual, incluindo localização e a configuração de seus recursos. Encoders (sensores especiais de rotação) foram instalados nas rodas, bem como em todos os motores dos dois braços da cabeça estéreo. Sensores de fim-de-curso foram instalados em todas as juntas da cabeça estéreo para que seja possível sua pré-calibração. Sonares e câmeras também farão parte do grupo de sensores utilizados no projeto. O robô contará com uma plataforma composta por, a princípio, dois computadores ligados a um barramento único para uma operação em tempo real, em paralelo. Um deles será responsável pela parte de controle dos braços e de sua navegação, tomando como base as informações recebidas dos sensores das rodas e dos próximos objetivos do robô. O outro computador processará todas as informações referentes à cabeça estéreo do robô, como as imagens recebidas das câmeras. A utilização de técnicas de imageamento estéreo torna-se necessária, pois a informação de uma única imagem não determina unicamente a posição de um dado ponto correspondente no mundo. Podemos então, através da utilização de duas ou mais câmeras, recuperar a informação de profundidade da cena. A cabeça estéreo proposta nada mais é que um artefato físico que deve dar suporte a duas câmeras de vídeo, movimentá-las seguindo requisições de programas (softwares) apropriados e ser capaz de fornecer sua pose atual. Fatores como velocidade angular de movimentação das câmeras, precisão espacial e acurácia são determinantes para o eficiente resultado dos algoritmos que nesses valores se baseiam. 62

Palavras-chave: Cabeça Robótica, Visão Estéreo

Sumário

Lista de Figuras	ii
1 Introdução	1
1.1 Contribuições	2
1.2 Motivações	3
1.3 Desafios	3
1.4 Importância	3
1.5 Aplicações	4
2 Trabalhos Relacionados e Introdução Teórica	5
2.1 Sistemas Visuais Biológico e Artificial	5
2.2 Visão Estéreo	7
2.3 Modelos cinemáticos de cabeças estéreo	10
2.4 Construção de cabeças estéreo × Cabeças disponíveis no mercado	13
2.5 Calibração Cinemática da Cabeça	14
2.6 Correção de distorção barril	16
2.7 Redução de Imagens	19
3 Implementação Física	23
3.1 Modelo Cinemático	23
3.2 Projeto Físico	28
3.3 Calibração Cinemática Inicial	29
4 Implementações Computacionais	32
4.1 Esquema proposto	33
4.2 Captura dos Quadros	35
4.3 Pré-processamento dos Quadros	36

4.3.1	Correção de Distorção Barril	37
4.3.2	Construção de Imagem em Multi-resolução	38
5	Extração de Características das Imagens	41
5.1	Filtragem de imagens	41
5.2	Detecção de Movimento	44
5.3	Correlação Cruzada para Disparidade	44
6	Experimentos e Resultados	46
6.1	Simulador	46
6.2	Aquisição de Imagens	50
6.3	Redução e Extração de Características de Imagens	51
7	Conclusão e Perspectivas	55
A	Projeto Físico da Cabeça	62

Lista de Figuras

2.1	Esquema estéreo humano.	8
2.2	Modelo cinemático proposto por RUF	10
2.3	Jannus	11
2.4	Cabeça Estéreo do Jannus	11
2.5	Modelo Cinemático proposto por Sheng.	12
2.6	Modelo Cinemático proposto por Sharma.	13
2.7	Comparação entre imagem distorcida e sem distorção.	17
2.8	Exemplos de distorção do tipo barril.	17
2.9	Outro exemplo de distorção do tipo barril.	18
2.10	Imagem da Lena em Multi-resolução.	21
2.11	Comparação entre imagem mono-resolução e multi-resolução.	21
2.12	Sobreposição de quatro Resoluções sobre a mesma imagem.	22
3.1	Modelo Cinemático da Cabeça Estéreo Robótica	24
3.2	Esboço da Cabeça Estéreo	29
4.1	Extração de Características de Imagens	34
4.2	Construção de imagem em multi-resolução.	35
4.3	Arquitetura Proposta	36
4.4	Criação de Imagem em Multi-Resolução	40
5.1	Máscara 3×3 com centro na posição $(i, j)=(2, 2)$	42
6.1	Simulador da Cabeça Estéreo	47
6.2	Parâmetros α e β no Simulador.	49
6.3	Simulador da Cabeça Estéreo Operando no Modo de Movimentação Independente	49
6.4	Cabeça Estéreo Operando no Modo de Movimentação Dependente	50

6.5	Simulador da Cabeça Estéreo perseguindo um objeto	50
6.6	Imagem Original	52
6.7	Imagem em Multiresolução	53
6.8	Filtro Gaussiano	53
6.9	Filtro Gradiente X	53
6.10	Filtro Gradiente Y	54
6.11	Filtro Laplaciano	54
6.12	Detecção de Movimento	54
7.1	Fotos do Robosense atual	56
7.2	Corpo do RoboSense	57
7.3	Detalhe da Cabeça Estéreo do RoboSense	58

Capítulo 1

Introdução

O Laboratório de Sistemas Inteligentes do Departamento de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte - UFRN - tem como um de seus projetos de pesquisa - Robosense - a construção de uma plataforma robótica móvel.

Trata-se de um robô provido de duas rodas, acionadas de forma diferencial, dois braços, com 5 graus de liberdade cada, um cinturão de sonares e uma cabeça estéreo. Como objetivo principal do projeto Robosense, o robô deverá ser capaz de navegar por todo o prédio do LECA, desviando de obstáculos.

O sistema de navegação do robô, responsável pela geração e seguimento de rotas, atuará em malha fechada. Ou seja, sensores serão utilizados pelo sistema com o intuito de informar ao robô a sua pose atual, incluindo localização e a configuração de seus recursos. *Encoders* (sensores especiais de rotação) foram instalados nas rodas, bem como em todos os motores dos dois braços e da cabeça estéreo. Sensores de fim-de-curso foram instalados em todas as juntas da cabeça estéreo para que seja possível sua pré-calibração. Sonares e câmeras também farão parte do grupo de sensores utilizados no projeto.

O robô contará com uma plataforma composta por, a princípio, dois computadores ligados a um barramento único para uma operação em tempo real, em paralelo. Um deles será responsável pela parte de controle dos braços e de sua navegação, tomando como base as informações recebidas dos sensores das rodas e dos próximos objetivos do robô. O outro computador processará todas as informações referentes à cabeça estéreo do robô, como as imagens recebidas das câmeras.

A utilização de técnicas de imageamento estéreo torna-se necessária, pois a informação de uma única imagem não determina unicamente a posição de um dado ponto correspondente no mundo. Podemos então, através da utilização de duas ou mais câmeras, recuperar a informação

de profundidade da cena.

A cabeça estéreo proposta nada mais é que um artefato físico que deve dar suporte a duas câmeras de vídeo, movimentá-las seguindo requisições de programas (softwares) apropriados e ser capaz de fornecer sua pose atual. Fatores como velocidade angular de movimentação das câmeras, precisão espacial e acurácia são determinantes para o eficiente resultado dos algoritmos que nesses valores se baseiam.

1.1 Contribuições

Dentro do projeto Robosense, com o presente trabalho, definimos o modelo cinemático utilizado para a construção da cabeça estéreo da plataforma robótica RoboSense, bem como a sua matriz de transformação e prestamos assessoria na elaboração do projeto físico do dispositivo, ficando fora do escopo deste trabalho as características eléto-mecânicas do projeto.

Participamos também da concepção e implementação de um simulador computacional da cinemática da cabeça estéreo. Definimos e implementamos ainda neste trabalho rotinas computacionais de aquisição e pré-processamento de imagens que darão suporte às aplicações de mais alto nível desenvolvidas para o robô.

Assim, as contribuições principais deste trabalho ao projeto RoboSense se concentram na cabeça estéreo do robô e podem ser desmembradas em um conjunto de ações mais específicas que delineiam melhor o escopo do trabalho. Entre estas contribuições, Podemos enumerar:

- Definição do modelo cinemático da cabeça estéreo;
- Assessoria na definição das características físicas da cabeça;
- Assessoria na construção da cabeça;
- Definição da arquitetura computacional utilizada na captura e processamento de imagens;
- Especificação das câmeras a serem utilizadas, de forma a garantir os requisitos do projeto;
- Implementação da aquisição das imagens por meio de tais câmeras;
- Implementação rotinas de pré-processamento para as imagens adquiridas;
- Garantia de velocidade, precisão e eficiência dos dados providos aos programas de inteligência que serão agregados ao projeto;
- Participação na implementação do simulador cinemático para a cabeça estéreo.

1.2 Motivações

A razão deste trabalho surge em função da especificação do projeto RoboSense. Nela, está prevista, entre outras funcionalidades do robô, a capacidade de se locomover em ambientes fechados e de realizar tarefas utilizando os seus dois braços manipuladores. Daí aparece a necessidade de que o robô "enxergue" e assim sendo decidiu-se por equipar a plataforma com uma cabeça estéreo dotada de câmeras.

1.3 Desafios

Cada um dos objetivos específicos do projeto é por si só um desafio se encarado de forma isolada. Entretanto, o encadeamento e a inter-dependência desses objetivos agrega um outro componente desafiador, na medida em que carregam em si a característica de repassar os erros das etapas anteriores às próximas, aumentando a complexidade do problema.

É importante ressaltar ainda o ponto crucial desse trabalho, que é a "garantia de velocidade, precisão e eficiência dos dados providos aos programas de inteligência que serão agregados ao projeto". Percebemos durante o decorrer da implementação do projeto que se for utilizado o método tradicional de captura e pré-processamento das imagens captadas pelas câmeras, o tempo decorrente dessas operações impossibilitaria a taxa de transferência de 20 quadros por segundo, ou próximo disso, requerida pelo robô para tarefas envolvendo controle direto.

O principal desafio desse trabalho é então a viabilização de tal taxa (ou próxima) com a mínima perda de qualidade possível na imagem adquirida ou nas informações provenientes dessas imagens.

1.4 Importância

A importância de trabalhos práticos não é novidade quando consideramos que a aplicação de conhecimentos teóricos em problemas reais aprofunda em seus executores a relação teoria-prática, além de sedimentar os conhecimentos em ambas as áreas.

A cabeça estéreo da base robótica RoboSense contribuirá com o Laboratório de Sistemas Inteligentes na medida em que proverá aos graduandos, mestrandos, doutorandos e professores envolvidos com o seu projeto uma experiência prática aprofundada no estudo de áreas a ela relacionadas, como a computação gráfica e visão computacional.

Além disso, toda a equipe envolvida com o projeto e construção do RoboSense será beneficiada pela Cabeça Estéreo Robótica, na medida em que ela possibilita a ampliação do conjunto de

funcionalidades do robô. A abertura de um leque de novas tarefas passíveis de serem realizadas agrega conhecimento e aprofunda com isso a capacitação técnica da equipe.

1.5 Aplicações

Um robô com as funcionalidades do RoboSense pode desempenhar vários papéis e realizar diversas funções úteis, podendo, por exemplo, guiar visitantes em amostras de arte, identificando as obras e dando explicações acerca delas. Pode ainda funcionar em bibliotecas procurando livros para leitores interessados. Uma outra função passível de ser desempenhada, em uma versão mais aprimorada do sistema e fazendo uso de linha telefônica celular, é a de vigia, onde o robô transitará por um andar de um prédio e acionará o celular caso identificasse alguma movimentação suspeita. Inúmeras tarefas tele-operadas podem ser desempenhadas pelo RoboSense. Nelas, o operador, protegido de possíveis adversidades, opera o robô, que executa o trabalho a distância.

Assim como essas, inúmeras outras tarefas úteis autônomas ou tele-operadas podem ser desempenhadas pelo RoboSense.

Capítulo 2

Trabalhos Relacionados e Introdução Teórica

Para melhor entender o que estamos propondo, trataremos de trabalhos realizados anteriormente e que se relacionam com o nosso trabalho, segundo alguns temas. São eles: **Sistemas Visuais Biológico e Artificial**, **Visão Estéreo**, **Modelos Cinemáticos de Cabeças Estéreis**, **Construção de Cabeças Estéreis × Cabeças Disponíveis no Mercado**, **Calibração Cinemática da Cabeça**, **Correção de Distorção Barrel** e **Redução de Imagens**.

Inúmeros são os trabalhos, artigos, projetos e textos sobre Visão Computacional, Cabeças Robóticas e seus subtópicos. Fizemos um apanhado de alguns destes trabalhos, que encontram-se compilados nas seções que se seguem.

2.1 Sistemas Visuais Biológico e Artificial

Muitas das técnicas desenvolvidas e utilizadas em visão computacional se baseiam ou derivam de sistemas biológicos, ou ainda foram desenvolvidos com forte inspiração nesse modelo. Ou seja, muitos trabalhos foram e continuam sendo realizados visando entender o sistema visual biológico na tentativa de desenvolver uma implementação computacional que o "imite".

Dentre os nossos sentidos de percepção, a visão desempenha o papel de maior importância. Sem ela, a realização de algumas tarefas se tornaria muito difícil ou até mesmo impossível. A visão provê uma enorme quantidade de informações de boa qualidade, sendo este sentido o mais complexo e o que ocupa um grande volume do cérebro. Isso dá uma idéia da grande quantidade de processamento envolvida com o sentido da visão.

Os sensores da visão adquirem grande quantidade de informação sobre o ambiente que nos cerca, facilitando a nossa interação com o mesmo. Por outro lado, os atuais conhecimentos sobre como sistemas visuais biológicos operam ainda são fragmentados e muito confinados nos estágios diretamente relacionados com o a captação dos sinais pelos sensores. O processamento desses sinais ainda não é conhecido. Na prática, o que se sabe é que esse processamento, e com isso o sistema de visão como um todo, é bastante complexo.

Sem entrar em detalhes biológicos sobre a forma ou sobre o tipo de informações que chegam ao cérebro através dos estímulos nervosos provocados pela luz filtrada incidente na retina ou sobre o processo de formação das imagens por nosso cérebro, pode-se dizer que nós usamos as diferenças entre as imagens captadas por nossos olhos para determinar, por triangulação, a distância a que um determinado objeto se encontra (do centro) de nosso sistema ocular, obtendo assim a noção da profundidade ou terceira dimensão.

O processo de estabelecer uma correspondência entre os objetos presentes nas duas imagens é denominado *matching*. Ele é a base da visão estéreo e é o gargalo de sistemas artificiais de visão ativa por demandar grande quantidade de processamento.

De forma semelhante ao olho humano, um sistema artificial de aquisição de imagens possui dispositivos sensíveis à luz refletida ou emitida pelos objetos. Esses foto-sensores determinam valores para a luminância de pequenas regiões de superfície. Os valores de luminância são quantizados através da divisão do espaço de percepção. A quantidade de subdivisões depende da capacidade e sensibilidade do dispositivo. A denominação usual dada para cada pequena região é píxel (pixmap element).

Para sistemas policromáticos, os níveis de intensidade da luminância de cada uma dessas pequenas regiões do espaço são normalmente separadas em três cores, denominadas primárias.

Para cada uma das cores primárias é criada uma matriz bi-dimensional composta pelo agrupamento dos valores de luminância captado para cada uma das pequenas regiões.

Normalmente a imagem resultante é conseguida através da composição das três matrizes bi-dimensionais, uma para cada cor, usando algum tipo de dispositivo combinante visando gerar a imagem reconstruída final.

Características de objetos tais como forma, textura e posição podem ser extraídas da análise das imagens através de manipulação computacional dos valores de níveis de luminosidade de cada cor.

De forma análoga à visão biológica, o processamento das informações fornecidas por sistemas de aquisição é complexo. O processamento a tempo real ainda não está bem resolvido, sendo o ponto crítico no projeto de protótipos de visão robótica.

2.2 Visão Estéreo

Apesar de o escopo deste trabalho excluir a tarefa de visão estereo, não pode-se deixar de considerar que as rotinas concebidas neste trabalho darão suporte a tal técnica. Assim sendo, o estudo de visão estereo terá que ser levado em conta na parte do planejamento e projeto da cabeça estereo e é fundamental que a equipe desenvolvedora dessas rotinas tenha conhecimento de visão estereo para que os produtos fornecidos atendam às necessidades da equipe desenvolvedora das funcionalidades de mais alto nível da cabeça. Uma introdução teórica sobre o assunto é desenvolvida nessa seção.

Segundo Júlio M. Otuyama[16], visão estereo é o ramo da visão computacional que analisa o problema da reconstrução da informação tridimensional de objetos a partir de um par de imagens capturadas simultaneamente, mas com um pequeno deslocamento lateral.

Baseando-se nessa definição e na necessidade e possibilidade de o robô reconstruir tri-dimensionalmente o espaço que o cerca, é fácil perceber a importância que a visão estereo tem no desenvolvimento das tarefas de mais alto nível da cabeça estereo do robô.

A ciência por muitas vezes se baseia na natureza e com a visão estereo não foi diferente.

A maioria dos animais é dotada de visão. Cada espécie com seu espectro de luz visível, sua estrutura ocular, quantidade e disposição de olhos.

Os animais que são predados na natureza, as presas, costumam possuir dois olhos posicionados na porção lateral do crânio. Assim seu campo de visão é aumentado, dando-lhe maior segurança contra os predadores, sob pena de perda de precisão na percepção de profundidade.

Ao contrário, os predadores têm os olhos posicionados na parte frontal do crânio. Essa disposição lhes fornece duas imagens da mesma cena, o que lhes confere a capacidade de distinguirem com precisão a profundidade de cada objeto nessa cena.

O método de visão estereo é baseado exatamente na visão dos predadores, que a partir de duas imagens da mesma cena captadas de posições ligeiramente diferentes, conseguem reconstruir tri-dimensionalmente o ambiente que os cercam.

A obtenção de visão tri-dimensional a partir de pares de imagens estereo idealizou-se há dois séculos, com a invenção da máquina fotográfica. O modelo utilizado é semelhante ao olho humano ilustrado pela figura 2.1. Da mesma forma, a partir das diferenças de posição das projeções de objetos nas imagens, por triangulação, pode-se determinar as suas posições relativas e uma vez que o sistema esteja calibrado (conhecidas as localizações de alguns poucos objetos), pode-se determinar a distância a que outros objetos se encontram do sistema de aquisição.

Técnicas fotogramétricas, usam os princípios de visão estereo para obtenção de dados para

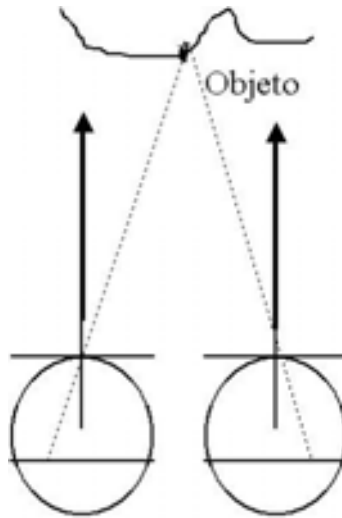


Figura 2.1: Esquema estéreo humano.

mapeamento a partir de fotografias da superfície terrestre, ou até mesmo da sub-superfície, obtidas por uma câmera aero-transportada.

Com o surgimento dos computadores e as técnicas de digitalização de imagens, na década de 70, as técnicas fotogramétricas passaram a sofrer processos de automatização, surgindo processos computacionais para reconstrução a partir de imagens estéreo ou, como é mais conhecida a técnica em questão, reconstrução estéreo (do inglês "*shape from stereo*"). O objetivo da técnica é a obtenção da terceira dimensão (profundidade) a partir de duas imagens digitais de uma mesma cena, obtidas de pontos de vista diferentes, bem como a correta identificação, a partir de modelos, dos objetos presentes na cena. Estas técnicas de reconstrução estão bem definidas, possuindo vasta bibliografia em artigos e livros texto como *Vision*[21], *Robot Vision*[19], *From Images to Surfaces*[22] e *Computer Vision*[20].

Excluindo as distorções ou erros devidos ao processo de aquisição (isto acontece no projeto RoboSense e precisa ser corrigido, como veremos adiante), pode-se considerar uma imagem como uma transformação projetiva de um conjunto de feições no espaço (tri-dimensional) em um plano (bi-dimensional). Os sistemas de aquisição digitais são esquemas discretizados que refletem os objetos contínuos na cena. Este esquema pode ser considerado como uma amostra estatística controlada, uma vez que as projeções de pontos nas imagens guardam uma relação bem definida com pontos na cena.

Desta forma, pode-se estabelecer algumas restrições e implementar algumas regras de reconstrução, com a determinação de parâmetros das transformação inversas, de modo que se possa mapear pontos de imagem (bi-dimensional) em seus correspondentes na cena (tri-dimensional),

reconstruindo a forma dos objetos em três dimensões a partir de suas projeções nas imagens.

Estas aplicações, cujos parâmetros são procurados, são basicamente transformações lineares de rotação, translação e de escala. Uma vez determinados os coeficientes destas transformações e conhecidas as posições de objetos nas imagens, pode-se determinar suas posições na cena.

Na reconstrução estéreo, o processo fundamental e mais custoso em termos do processamento é a identificação, nas duas imagens, das projeções correspondentes a um mesmo ponto objeto. Esta correspondência é também conhecida como casamento, do termo inglês *matching* (usaremos este último em algumas situações por ser muito adotado). A correspondência pode ser feita com utilização de vários processos matemáticos ou estatísticos [23], incluindo correlação de áreas ou de características, uso de diferenças de níveis de cinza entre elementos vizinhos e detecção de arestas.

Normalmente as imagens são pré-filtradas visando a eliminação dos efeitos de altas frequências e visando realçar ou separar algum tipo de característica. Após esta filtragem é então calculado o *matching*. A pré-filtragem faz parte do escopo deste trabalho, ao contrário do *matching*, que será implementado em etapas posteriores do projeto.

O RoboSense requer o processamento de *matching* em tempo real, ou seja, à medida que as imagens são adquiridas. Após o estabelecimento da correspondência serão usadas técnicas de triangulação e integração numérica para a reconstrução da profundidade e conseqüente determinação da forma de objetos.

Note que o ideal é estabelecer a correspondência para todos os pixels das imagens. Na prática, isto não ocorrerá. Geralmente não é possível relacionar todos os pontos devido a oclusão ou ambigüidade de elementos que podem ocorrer entre uma imagem e outra. No final do processo, teremos a determinação de correspondência para vários elementos, a partir dos quais serão interpolados valores de profundidade para o restante.

Se não houver textura na cena (discrepâncias entre atributos de pixels), não será possível a determinação da profundidade, o que pode naturalmente ser entendido com a seguinte comparação: se nos aproximarmos de uma parede lisa e sem discrepância de cores, por exemplo, lisa e branca, nossos olhos verão apenas uma imensidão branca, não percebendo a profundidade. O que ocorre é que não há detalhes de textura que permitam a formação do triângulo olho esquerdo-objeto-olho direito, para determinação da profundidade.

Mais informações sobre o assunto “visão estéreo” podem ser obtidas na literatura a respeito, que conta com vários textos introdutórios, como por exemplo a apostila *Stereo Binocular*[17], de Alberto Vale. O autor aborda diversos temas sobre o assunto em um formato bastante agradável de ler e com razoável aprofundamento.

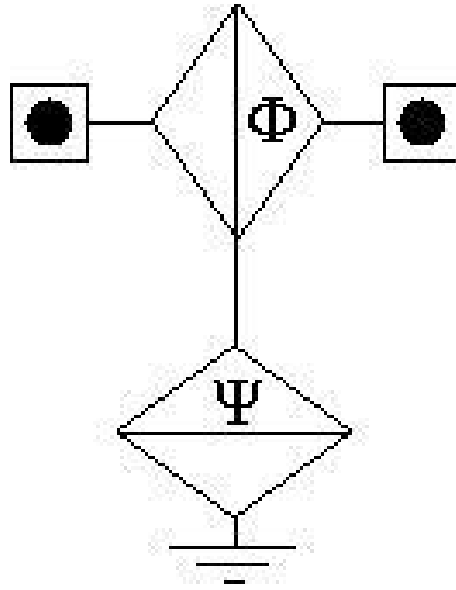


Figura 2.2: Modelo cinemático proposto por RUF

2.3 Modelos cinemáticos de cabeças estéreo

Segundo Paul A. Keenan[14], um modelo cinemático é um conjunto de relações matemáticas, determinado pelas dimensões do robô, que servem para traduzir as coordenadas cartesianas, representando a posição desejada da ferramenta, a partir das posições angulares de cada junta do robô.

Porém, neste capítulo, o modelo cinemático é tratado sob uma ótica ligeiramente diferente.

O que está em foco aqui é a forma, o arranjo e a capacidade de movimentação das juntas do robô, em detrimento às transformações matemáticas que o modelam. Tais Características como quantidade de graus de liberdade, tipo das juntas e disposição de tais juntas na cadeia são relevantes para essa abordagem, enquanto as dimensões dos elos e amplitude de movimentação, por exemplo, são características de importância secundária e não serão mencionadas.

No trabalho de Andreas RUF [7], é apresentado um sistema computacional constituído de um braço manipulador com seis graus de liberdade, observado por uma cabeça estéreo com dois graus de liberdade cujo modelo cinemático segue o exposto na Figura 2.2.

O modelo cinemático escolhido para a cabeça estéreo representa a forma mais simplificada de implementação do sistema Pan-Tilt para duas câmeras. Nele, as posições angulares de Pan e Tilt, bem como suas velocidades angulares, são as mesmas para as duas câmeras. Na arquitetura apresentada por RUF, as câmeras têm vergência fixa de 15° . Ele garante em seu trabalho uma precisão de $0,01^\circ$, tendo a sua cabeça 20 passos de PAN e 10 de TILT, todos com amplitude de

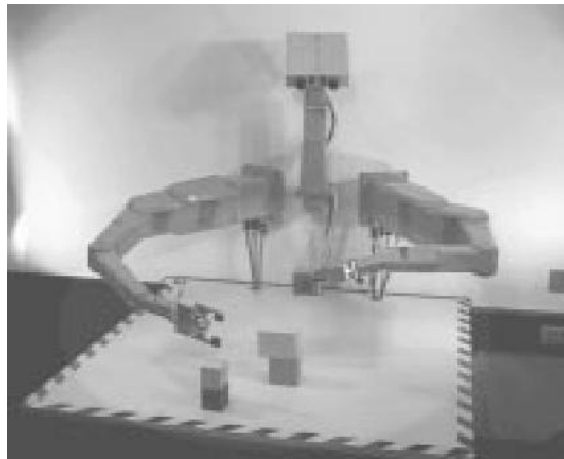


Figura 2.3: Jannus: Robô Antropomórfico baseado na parte superior de um tórax humano.



Figura 2.4: Cabeça Robótica Estéreo do Jannus.

1°.

A cabeça estéreo proposta e implementada por RUF tem um modelo cinemático bastante simplificado e movimentos bastante restritos, pois possui poucos passos de amplitude relativamente alta. Entretanto, a cabeça deve atender bem às expectativas, uma vez que as atribuições requeridas não apresentam grandes complicadores, tendo em vista que o espaço tri-dimensional por ela mapeado (que coincide com o espaço de trabalho do braço) é reduzido, conhecido e fixo em relação à sua base.

Um outro esquema bastante simples de uma cabeça estéreo com duas câmeras é proposto por Christophe Garcia[8]. Trata-se do projeto do robô antropomórfico baseado na parte superior do tórax humano exposto na figura 2.3. Ele deve ser capaz de manipular objetos sem colisão dentro de seu espaço de trabalho baseando-se em informações visuais advindas de sua cabeça estéreo. Ela é mostrada na figura 2.4 e possui 2 graus de liberdade, conforme exposto na figura 2.3.

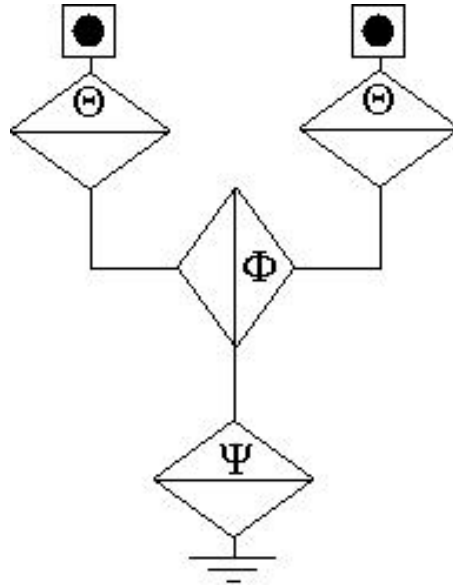


Figura 2.5: Modelo Cinemático proposto por Sheng.

Esse modelo se diferencia do modelo de RUF[7] somente pela posição relativa entre a junta rotacional ϕ e as câmeras, mantendo a mesma quantidade de graus de liberdade, a vergência fixa e a dependência entre as câmeras como características do projeto.

Pesquisadores da National Taiwan University apresentam um modelo de cabeça robótica com três graus de liberdade [9]. No trabalho, ela encontra-se montada em dois trilhos perpendiculares que permitem a translação da cabeça sobre no plano horizontal para fins de calibração. O terceiro grau de liberdade da cabeça apresentada fica por conta da vergência variável entre as câmeras. O modelo conforme exposto por eles é reproduzido na Figura 2.5. Deve-se ressaltar, entretanto, que o ângulo de vergência ϕ como proposta pelos autores assume o mesmo valor para as duas câmeras, que são rotacionadas segundo acionamento de um só motor.

Um modelo um pouco mais rebuscado foi utilizado por Rajeev Sharma e Narayan Srinivasa[10] em 1998. O projeto em questão consta de um braço manipulador com três juntas rotacionais observado por uma cabeça estéreo robótica. No trabalho, a exemplo de todos os outros já citados, os movimentos de PAN e de TILT da cabeça estéreo são comuns às duas câmeras. Entretanto, neste trabalho os autores apresentam uma diferença em relação aos anteriores no que tange ao modelo cinemático. Nele foi utilizado acionamento independente para o movimento de vergência de cada câmera, o que confere à cabeça um quarto grau de liberdade. Assim sendo, os quatro graus de liberdade do projeto são PAN, TILT, VERGÊNCIA ESQUERDA e VERGÊNCIA DIREITA, conforme exposto na Figura 2.6.

O modelo cinemático adotado no projeto RoboSense será abordado e suas funcionalidades

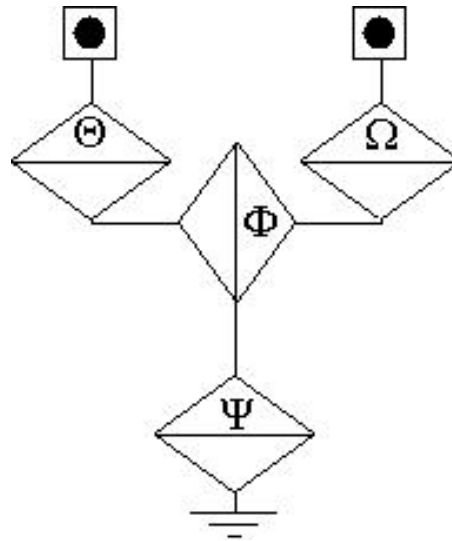


Figura 2.6: Modelo Cinemático proposto por Sharma.

detalhadas mais adiante.

2.4 Construção de cabeças estéreo × Cabeças disponíveis no mercado

Duas possibilidades foram levadas em conta nas fases iniciais do projeto:

- Construir a Cabeça Estéreo Robótica;
- Adquirir uma Cabeça no mercado que atendesse aos requisitos especificados.

Adquirir a cabeça simplificaria o trabalho e agilizaria o projeto.

Apesar disso, dois fatores pesaram na escolha tomada pelos projetistas, que se resolverem por construir a cabeça, ao invés de comprá-la pronta.

A primeira das razões levadas em consideração foi o aprendizado resultante do processo de construção obtido pela equipe envolvida. A segunda, foi o custo da aquisição de uma cabeça pronta, que é superior ao custo de construção de um dispositivo próprio.

Mesmo certos da construção da cabeça, decidiu-se fazer uma pesquisa entre as cabeças robóticas disponíveis no mercado, tendo em vista que ela proveria dados de suporte à escolha dos requisitos da cabeça projetada, delinear um horizonte de custos aceitáveis para a cabeça e proporcionaria aos membros do projeto um posicionamento claro com relação ao *estado-da-arte* nessa área.

Conseguimos encontrar somente uma família de Cabeças Estéreo Robóticas vistas na literatura como disponíveis comercialmente no mercado, a Biclops. Todos os outros projetos acadêmicos construíram as suas próprias cabeças ou as compraram de fabricantes que descontinuaram a sua produção (Triclops, Bisight).

A única família de cabeças estereo robóticas disponíveis no mercado é a Biclops, do Laboratório TracLabs. Algumas de suas características são as seguintes:

- Massa: 1,1Kg;
- Resolução: 1,8 arco-minuto para PAN e TILT;
- Amplitude: Amplitude de $-165^{\circ}/+165^{\circ}$ de PAN, $-60^{\circ}/+60^{\circ}$ de TILT e $0/11^{\circ}$ de vergência;
- Velocidade Angular: $300^{\circ}/s$ em PAN e TILT e $50^{\circ}/s$ em vergência.

As características mecânicas mencionadas para a família Biclops de cabeças robóticas são bastante satisfatórias e alguns pontos devem ser ressaltados:

O bom funcionamento de todo o sistema depende da precisão com a qual a cabeça estereo posiciona as câmeras no espaço. O que é fácil de concluir quando partimos do princípio de que a metodologia de visão estereo se baseia não somente nas imagens captadas, mas também da posição relativa das câmeras para calcular a posição real dos objetos na cena.

Por esse motivo a boa resolução apresentada pela Biclops deve ser encarada como um bom suporte ao método de visão estereo e, justamente por isso, como um forte ponto a seu favor.

Um outra característica desejável a uma cabeça estereo que se propõe a funcionar em tempo real é a agilidade de seus movimentos. Ela deve ser capaz de acompanhar um objeto que se movimenta em velocidade razoável a sua frente, o que implica em velocidades angulares relativamente altas de suas juntas. E esse é um outro aspecto que pesa a favor da Biclops, uma vez que seus $300^{\circ}/s$ de PAN e TILT e $50^{\circ}/s$ de vergência são mais que suficientes para essa finalidade.

Entretanto os custos associados à aquisição dos equipamentos dessa família variam de US\$3500,00 a US\$5500,00, acrescidos ainda de impostos. Além de as câmeras não fazerem parte do conjunto e, portanto, deverem ser compradas separadamente. O elevado custo do equipamento influenciou também a opção de construção da cabeça em detrimento de adquirí-la já construída no mercado.

2.5 Calibração Cinemática da Cabeça

Qualquer aplicação envolvendo uma relação entre uma imagem bi-dimensional e a realidade tri-dimensional requer que seja feita a calibração da câmera, que corresponde a obtenção dos seus

parâmetros intrínsecos.

As aplicações em robótica requerem ainda uma calibração extra para a obtenção dos parâmetros extrínsecos do robô.

Ambas as calibrações devem ser realizadas pelo menos uma vez, no início da operação do sistema.

Os parâmetros intrínsecos são as características geométricas e ópticas das câmeras, como por exemplo seu zoom, foco e distorção causada por suas lentes.

Os parâmetros extrínsecos são as características físicas da cabeça em determinado instante. Dimensão, posição e a orientação de cada junta são exemplos de parâmetros extrínsecos.

Como as câmeras adquiridas para o projeto não variam automaticamente nenhum de seus parâmetros intrínsecos, a calibração de tais parâmetros não se faz necessária.

Não se pode deixar de citar, entretanto, que a correção da distorção imposta pelas lentes pode ser encarada como uma forma de calibração intrínseca e será tratada de forma mais aprofundada em seções mais adiante.

Com relação às características extrínsecas, a posição e a orientação de cada junta da cabeça precisam ser calibradas quando no início do funcionamento do robô. A partir do instante de encerramento desse procedimento pode-se monitorar a pose assumida pela cabeça e, com isso, a posição e orientação das câmeras em relação a um sistema de coordenadas globais, fixo no espaço.

Aqui tratamos exclusivamente da calibração dos parâmetros extrínsecos da cabeça, ou calibração cinemática, deixando o estudo da calibração dos parâmetros intrínsecos para a seção 4.3.1.

Um exemplo de calibração dos parâmetros cinemáticos de uma cabeça robótica binocular é apresentado por Sheng et al [9], que propõe um método baseado apenas na medição da posição tri-dimensional de um ponto de calibração situado no campo de visão da cabeça. Ela possui quatro juntas rotacionais e está montada sobre dois trilhos perpendiculares que lhe conferem dois graus de liberdade adicionais, funcionando os mesmos como duas juntas prismáticas.

Esse modelo se propõe a realizar a calibração cinemática sem necessitar de sensores de posição nas juntas, a partir dos quais a configuração do robô é aferida.

O autor subdivide o problema da calibração da cabeça e o encara como sendo a união das calibrações de cada junta. Em seu algoritmo, a calibração da i -ésima junta se dá movendo (rotacionando ou transladando) a próxima junta da cadeia. O algoritmo identifica a variação da posição que o ponto de calibração assume nas imagens capturadas pelas duas câmeras e calcula o parâmetro cinético (posição) de tal junta. Tal procedimento é aplicado a cada junta da cadeia até que toda a cabeça seja calibrada.

Uma outra forma de calibração dos parâmetros cinemáticos das juntas de uma ferramenta

robótica é dotar cada uma dessas juntas de sensores de fim de curso e rotacioná-las até o disparo de tais sensores, indicando o posicionamento da junta. Esta foi a estratégia adotada pelo RoboSense.

2.6 Correção de distorção barril

A importância de se realizar um estudo sobre a distorção do tipo barril surge do fato de as câmeras adquiridas pelo projeto apresentarem distorção desse tipo, necessitando de correção.

Distorção em imagens é, por definição, o fenômeno ótico que ocorre quando os pontos de um objeto são mapeados em posições diferentes de onde eles deveriam originalmente aparecer em uma imagem.

Várias são as razões que podem vir motivar a ocorrência da distorção, como por exemplo a forma da lente, sua inclinação em relação ao eixo objeto-observador e a inclinação do plano de projeção em relação ao mesmo eixo.

A distorção do tipo barril, que dá à imagem uma impressão de convexidade esférica, ocorre com relativa frequência durante o processo de aquisição de imagens. Essa distorção é ocasionada pelo formato de algumas lentes, que causa diferença na ampliação do objeto em diferentes pontos da imagem.

Os pontos do objeto são representados fora de sua localização correta e essa discrepância varia de acordo com a posição e distância do ponto em relação ao centro da imagem. Um exemplo de distorção do tipo barril é exibido nas Figuras 2.7¹, 2.8² e 2.9³, onde pode-se perceber que os pontos mais distantes do centro são mais afetados pela distorção que os pontos centrais.

As câmeras adquiridas para o RoboSense têm como características a geração da distorção do tipo barril, que deve ser corrigida antes do processamento aplicado às imagens. Caso a correção não fosse implementada, o sistema visual do robô se basearia em imagens que não refletem integralmente a realidade, dificultando a sua implementação e podendo inclusive apresentar comportamentos insatisfatórios sob determinadas condições.

Assim sendo, é conveniente decrevermos algumas propostas de correção da distorção do tipo barril neste trabalho.

Uma implementação em hardware para a correção de distorção barril é proposta por K.T. Gibbon[11] para uma taxa de captura de 60 quadros por segundo. Para isso, foi utilizado um FPGA (Field Programmable Gate Array), que é uma matriz reprogramável de blocos lógicos dis-

¹Fonte: <http://foto.hut.fi/opetus/300/luennot/3/3.html>

²Fonte: http://www.cofa.unsw.edu.au/units/resource/equipment_guide/still_camera/distortion.shtml e

http://www.dpreview.com/learn/?/Image_Techniques/Barrel_Distortion_Correction_01.htm

³Fonte: <http://ftp.pcworld.com/pub/screencams/1-29-figure1.jpg>



Figura 2.7: Comparação entre imagem distorcida e sem distorção.



Figura 2.8: Exemplos de distorção do tipo barril.

postos de forma a se conectarem segundo as especificações do usuário. O dispositivo foi configurado de forma a implementar um algoritmo de correção de distorção barril associado a Interpolação Bi-Linear. Para isso, o modelo de distorção utilizado foi o proposto por Bailey[12]. O resultado conseguido foi satisfatório.

A implementação de um algoritmo alternativo para a correção de distorção do tipo barril é proposta por Michael R. Bax em *Real-Time Lens Distortion Correction: 3D Video Graphics Cards*

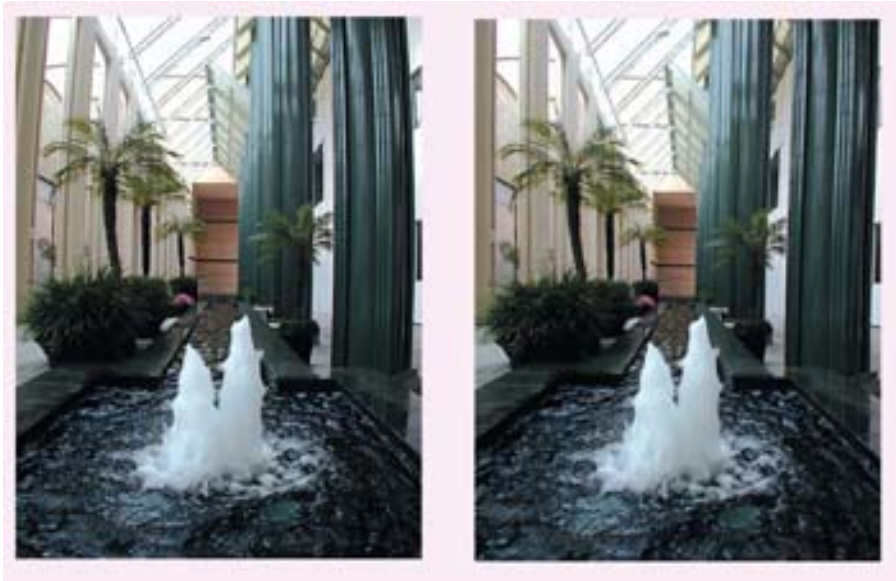


Figura 2.9: Outro exemplo de distorção do tipo barril.

Are Good for More than Games[13]. Neste trabalho ele implementa em uma placa aceleradora de vídeo 3D, ou Unidade de Processamento Gráfico (GPU em Inglês) toda a operação de correção de distorção, deixando livre a CPU do computador para outras finalidades. Tal característica vem a ser bastante desejável para este projeto.

O Processador Gráfico utilizado no trabalho foi uma placa de vídeo GeForce2-32MB, da NVIDIA e o computador equipado com um processador Pentium III-900MHz, da Intel, com 512MB de memória e rodando a plataforma Windows XP, da Microsoft.

Foram confrontadas por ele quatro metodologias de correção de distorção do tipo barril, sendo uma processada pixel-a-pixel pela CPU do computador e as outras três com processamento utilizando a representação de imagens como utilizada nas GPU's e mapeamento de texturas. Os resultados desse confronto foram bastante favoráveis ao processamento por GPU. Essa abordagem, com taxa de conversão de cerca de 25ms/quadro, obteve desempenho de processamento mais de cinco vezes melhor que o modelo utilizando CPU, que precisava de 168ms para corrigir a distorção de apenas uma imagem.

A metodologia utilizada no trabalho citado acima não é simples, requerendo um estudo específico mais aprofundado para sua implementação. Entretanto a sua utilização pode ser tentada no projeto do RoboSense caso a necessidade venha a ser detectada. Ao invés disso, a metodologia descrita a seguir, por ser mais simples e usual foi escolhida e adotada inicialmente no projeto, tendo sido implementada.

O livro de Trucco [24] propõe um algoritmo simples mas eficiente para a correção da distorção

barril. Toda câmera tem um coeficiente de distorção k . O problema resulta em calcular esse coeficiente e depois calcular o local exato de cada pixel na imagem. Para calcular essa nova localização pode-se usar as seguintes equações:

$$X = X_d(1 + k_1r^2 + k_2r^4) \text{ e}$$

$$Y = Y_d(1 + k_1r^2 + k_2r^4)$$

Onde X_d e Y_d são os pontos da imagem distorcida, o r^* é a distância do centro da imagem até o ponto e k_1 e k_2 são os coeficientes de distorção. Vale salientar que o modelo proposto por Trucco tem ordem infinita. Aqui truncamos o modelo para diminuir o processamento, uma vez que não percebe-se perda significativa de qualidade nos resultados. Nessa equação utilizam-se dois coeficientes de distorção.

Pela equação, percebe-se que quanto mais perto do centro menor será a distorção, sendo ela nula quando estiver no centro.

Ou seja, a partir da determinação de pontos nas imagens de uma figura geométrica padrão (um quadrado) calcula-se os dois coeficientes de distorção e aplica-se uma transformação em cada pixel da imagem segundo as equações quadráticas acima, envolvendo os coeficientes obtidos. O resultado é a retirada de distorção do tipo barril. Este foi o método implementado por nós neste projeto.

O algoritmo de Devernay e Faugeras[27], também soluciona o problema de retirada de distorção de imagem de forma bem simples, tendo sido implementado posteriormente por outros alunos [26].

2.7 Redução de Imagens

Um desafio enfrentado na maioria dos projetos computacionais que se utilizam de visão estéreo é o elevado tempo de execução dos algoritmos de processamento das imagens captadas por suas câmeras. Isso se dá devido à grande quantidade de informações contida em uma imagem, que requer, por sua vez, elevada capacidade de processamento.

Uma vez que o processo acontece em tempo real e exige do sistema uma taxa mínima de quadros a serem processados por segundo, a agilidade das rotinas é essencial e deve ser buscada.

O sistema de redução de imagens pretendido neste trabalho se propõe a descartar das imagens as informações desnecessárias ou menos importantes com o intuito de agilizar o seu processamento. Por essa razão que achou-se por bem realizar um estudo a esse respeito.

O tempo necessário ao processamento de uma imagem geralmente varia em função da quantidade de *pixels*. Assim sendo, quanto menor as dimensões das imagens, e conseqüentemente a sua quantidade de *pixels*, mais célere é o funcionamento do algoritmo. Por outro lado, a redução

das dimensões das imagens trazem consigo redução do campo de visão ou perda na quantidade de informações representadas.

A compactação de dados em imagens pode ser feita de algumas formas.

A compactação de imagens **por codificação**, exposta por **Gonzalez e Woods**[3], baseia-se na alteração do código que representa a cor de cada pixel de forma a diminuir o espaço necessário para o armazenamento da imagem, mas sem alterar as dimensões da imagem nem sua quantidade de *pixels*. A aplicação de tal estratégia não aceleraria o processo de pré-processamento das imagens adquiridas pelas câmeras, tendo efeito inverso. Ou seja, este procedimento aumentaria o tempo necessário para a execução de tal tarefa, uma vez que necessita da criação de uma tabela de mapeamento que define o novo código adotado na representação da imagem, que deve ser consultada a cada referência a um *pixel* da imagem.

Uma outra técnica de redução dos dados em imagens baseia-se na alta correlação entre *pixels* vizinhos. Ela representa uma imagem através da diferença entre *pixels* adjacentes, o que reduz a quantidade de *bits* necessária para o armazenamento da imagem, mas ainda mantém a quantidade de *pixels* da imagem, o que não é desejado no projeto. Entretanto, daqui podemos retirar um importante conceito: o da **redundância interpixels**, ou seja, muito da informação carregada por um *pixel* pode ser prevista pelos seus vizinhos. Essa constatação é relevante e embasa teoricamente o esquema proposto de redução de dados.

Uma técnica de abstração e redução de imagens inicialmente aplicada para visão robótica baseia-se na fóvea de primatas e é apresentada por Bolduc et. al.[5]. Nela, a redução dos dados é feita de forma a construir uma imagem circular com gradativo aumento de resolução axialmente na direção do centro da figura. Para o projeto, tal abstração seria útil na medida em que reduz a quantidade de *pixels* da imagem, mantendo a qualidade em seu centro (fóvea) e reduzindo-a gradual na direção da periferia da imagem. Entretanto, a questão que inviabiliza o uso desse método no projeto é a sua elevada complexidade, o que, no caso do RoboSense, traria consigo o problema que se deseja eliminar: o elevado intervalo de tempo necessário para sua execução.

Um exemplo de imagem resultante de transformação baseada na retina humana é exibido na Figura 2.10⁴. Nela, percebe-se maior resolução no centro da imagem e gradativa perda de resolução a medida que os *pixels* se afastam dele.

A comparação entre outras imagens normais e em multi-resolução pode ser vista na figura 2.11.

A Figura 2.12 mostra uma imagem em multi-resolução. Trata-se entretanto, de um esquema diferente dos demais, onde a resolução vai diminuindo gradativamente a medida que os *pixels* se afastam do centro da imagem. Ao invés disso, a resolução diminui em degraus. Pode-se perceber

⁴Fonte: http://live.ece.utexas.edu/research/fsvic/hvs_model.htm.

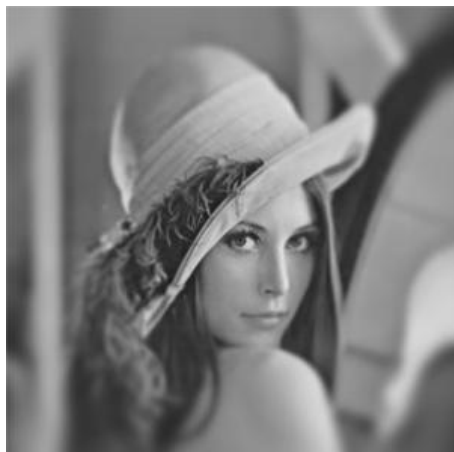


Figura 2.10: Imagem da Lena em Multi-resolução.



Figura 2.11: Comparação entre imagem mono-resolução e multi-resolução.

regiões quadradas concêntricas (e com centro no centro da imagem) de resolução constante.

Esse modelo de multi-resolução, que será detalhado mais adiante, é o proposto para trabalho no RoboSense.

Muitos outros estudos e possibilidades acerca de imagens em multi-resolução são encontrados na literatura, como vários pontos focais na mesma imagem, multi-resolução baseada em segmentação de imagem e multi-resolução aplicada a vídeos. Entretanto esses aprofundamentos extrapolam as necessidades desse trabalho e por isso não serão detalhados.

O esquema de redução dos dados proposto para o projeto é um sistema simplificado da redução



Figura 2.12: Sobreposição de quatro Resoluções sobre a mesma imagem.

baseada na fóvea de primatas e no princípio da **redundância interpixels**. Trata-se de um algoritmo mais ágil que divide a imagem em apenas cinco níveis de resolução.

Uma proposta semelhante é feita por Young, Scott e Bandera [15]. Nela a imagem é processada de forma a ser representada pelo conjunto de quatro em diferentes níveis de resolução. A metodologia do trabalho se assemelha muito à adotada no projeto RoboSense, diferenciando apenas pelo uso de uma rede neural para a construção da imagem em multi-resolução. O RosoSense adotou uma forma de construção baseada no princípio da **redundância interpixels**, que é mais simplificada.

Capítulo 3

Implementação Física

Para a implementação física da cabeça robótica estéreo, devemos especificar o modelo cinemático escolhido para a sua construção, os sistemas de coordenadas e suas matrizes de transformação.

Devemos ainda definir as características físicas da cabeça, a arquitetura do sistema de aquisição de imagens e os equipamentos a ela necessários. As características dos elos da cabeça, suas dimensões e o material utilizado na construção encontram-se documentados no Anexo A.

Um terceiro aspecto a ser abordado é o modelo escolhido para calibração dos parâmetros cinemáticos da cabeça.

3.1 Modelo Cinemático

Do ponto de vista cinemático, a cabeça estéreo robótica pode ser considerada como um conjunto de corpos rígidos, chamados *elos*, interligados, formando duas cadeias cinemáticas abertas através de juntas, com uma extremidade fixa na base da cabeça estéreo, e em sua outra extremidade uma câmera.

A posição de cada câmera é dada por uma transformação matemática baseada no modelo cinemático da cabeça, na posição de cada junta rotacional interligada aos elos e das dimensões de cada elo.

A Figura 3.1 representa o modelo cinemático correspondente ao da cabeça estéreo. Nele o sistema de coordenadas número 0 (zero) encontra-se na base da cabeça, o sistema de coordenadas número 1 (um) é responsável pelo movimento de giro em torno do eixo vertical de toda a cabeça, os sistemas de coordenadas 2e e 2d (esquerdo e direito) quantificam o grau de vergência de cada câmera e, por fim, os sistemas 3e e 3d (esquerdo e direito) permitem o giro de cada câmera individualmente em torno do eixo horizontal. É importante ressaltar a coincidência das origens dos

sistemas de coordenadas $2d$ e $2e$ e dos sistemas $2d$ e $2e$ entre si, que acontece propositalmente com o intuito de simplificar os cálculos inerentes às transformações entre os sistemas de coordenadas.

Do modo como foi pensado o modelo cinemático, θ_{3e} e θ_{3d} têm valores zero quando as câmeras apontam para frente e crescem com o aumento de sua vergência, ou seja, quando o objeto focalizado aproxima-se. Já θ_{2e} e θ_{2d} foram pensados de forma a assumirem o valor zero quando as câmeras estiverem direcionadas na posição horizontal e aumentarem com o movimento de “abaixar os olhos”.

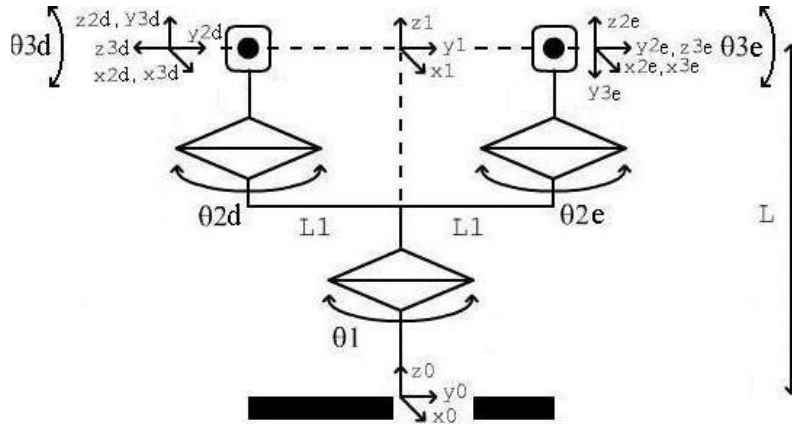


Figura 3.1: Modelo Cinemático da Cabeça Estéreo Robótica

Sendo $L = 210\text{mm}$, $L1 = 125\text{mm}$.

A partir dos valores das variáveis de junta em qualquer instante de tempo pode-se determinar a posição e orientação das câmeras, permitindo a visualização da cena captada pelas câmeras. A cinemática direta é gerada utilizando transformações homogêneas que relacionam a localização de um elo em relação ao elo anterior na cadeia cinemática. A escolha do posicionamento e orientação dos sistemas de coordenadas foi motivada pela facilidade de entendimento dos programadores que implementarão as rotinas de visão computacional em mais alto nível em detrimento da padronização desenvolvida por Denavit-Hartenberg. A análise cinemática de um lado do modelo da cabeça estereo robótica pode ser feita obtendo-se as matrizes de transformações de elos consecutivos, a partir das quais encontra-se a matriz de transformação entre uma das câmeras (junta $3e$ ou $3d$) e a base (junta 0). Devido à simetria da cabeça, a análise do lado oposto é simplificada.

Para o modelo completo deve-se aplicar a cinemática direta para cada câmera, obtendo um sistema com duas matrizes de cinemática direta, que possibilitam o mapeamento das posições de cada uma das câmeras em relação a um referencial fixo na base da cabeça.

As matrizes de transformações de cinemática direta de ambas as câmeras são desenvolvidas a seguir a partir das matrizes de transformação elo-a-elo desde a base até as câmeras.

Para simplificar a leitura e enxugar a notação, consideraremos $c_1 = \cos(\theta_1)$, $s_1 = \sin(\theta_1)$, $c_{2e} = \cos(\theta_{2e})$, $s_{2e} = \sin(\theta_{2e})$, $c_{2d} = \cos(\theta_{2d})$, $s_{2d} = \sin(\theta_{2d})$, $c_{3e} = \cos(\theta_{3e})$, $s_{3e} = \sin(\theta_{3e})$, $c_{3d} = \cos(\theta_{3d})$ e $s_{3d} = \sin(\theta_{3d})$.

Sendo ${}^A T_B$ a matriz de transformação homogênea que relaciona os pontos no sistema de coordenadas B com seus equivalentes no sistema de coordenadas A e $Trans(C, D_i)$ uma translação simples de C metros ao longo do eixo D_i e $Rot(E^o, F_j)$ uma rotação simples de E^o em torno do eixo F_j , tem-se:

Para a câmera do lado esquerdo:

$${}^0 T_{3e} = {}^0 T_1 \cdot {}^1 T_{2e} \cdot {}^2 T_{3e}$$

Portanto, para chegarmos na matriz de transformação homogênea para a câmera esquerda ${}^0 T_{3e}$ partimos de ${}^0 T_1$, ${}^1 T_{2e}$ e ${}^2 T_{3e}$, que são as matrizes de transformação homogênea de cada elo conforme mostrado abaixo.

$${}^0 T_1 = Trans(L, Z_0) \cdot Rot(\theta_1, Z_1)$$

$${}^0 T_1 = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ c_1 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1 T_{2e} = Trans(L_1, Z_0) \cdot Rot(-\theta_{2e}, Z_{2e})$$

$${}^1 T_{2e} = \begin{bmatrix} s_{-2e} & c_{-2e} & 0 & 0 \\ c_{-2e} & -s_{-2e} & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -s_{2e} & c_{2e} & 0 & 0 \\ c_{2e} & s_{2e} & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2 T_{3e} = Rot(-90^\circ, X_{2e}) \cdot Rot(\theta_{3e}, Z_{3e})$$

$${}^2 T_{3e} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{3e} & c_{3e} & 0 & 0 \\ c_{3e} & -s_{3e} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2eT_{3e} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{3e} & c_{3e} & 0 & 0 \\ c_{3e} & -s_{3e} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2eT_{3e} = \begin{bmatrix} s_{3e} & c_{3e} & 0 & 0 \\ c_{3e} & -s_{3e} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Como:

$${}^0T_{3e} = {}^0T_1 \cdot {}^1T_{2e} \cdot {}^2eT_{3e}$$

Temos:

$${}^0T_{3e} = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ c_1 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -s_{2e} & c_{2e} & 0 & 0 \\ c_{2e} & s_{2e} & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{3e} & c_{3e} & 0 & 0 \\ c_{3e} & -s_{3e} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

O resultado da multiplicação das três matrizes acima encontra-se exposto abaixo e é a matriz de transformação direta da câmera esquerda ${}^0T_{3e}$.

$${}^0T_{3e} = \begin{bmatrix} c_{3e}(-s_1s_{2e} + c_{3e}c_{2e}) - s_{3e}(s_1c_{2e} + c_1s_{2e}) & s_{3e}(-s_1s_{2e} + c_1c_{2e}) + c_{3e}(s_1c_{2e} + c_1s_{2e}) & 0 & L_1c_1 \\ -c_{3e}(c_1s_{2e} + s_1c_{2e}) + s_{3e}(-c_1c_{2e} + s_1s_{2e}) & -s_{3e}(c_1s_{2e} + s_1c_{2e}) - c_{3e}(-c_1c_{2e} + s_1s_{2e}) & 0 & -L_1s_1 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Para a câmera do lado direito:

$${}^0T_{3d} = {}^0T_1 \cdot {}^1T_{2d} \cdot {}^2dT_{3d}$$

Portanto, a exemplo da câmera do lado esquerdo, para chegarmos na matriz de transformação homogênea para a câmera direita ${}^0T_{3d}$ partimos de 0T_1 , ${}^1T_{2d}$ e ${}^2dT_{3d}$, que são as matrizes de transformação homogênea de cada elo conforme mostrado abaixo.

$${}^0T_1 = Trans(L, Z_0) \cdot Rot(\theta_1, Z_1)$$

$${}^0T_1 = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ c_1 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_{2d} = \text{Trans}(-L_1, Z_0) \cdot \text{Rot}(\theta_{2d}, Z_{2d})$$

$${}^1T_{2d} = \begin{bmatrix} s_{2d} & c_{2d} & 0 & 0 \\ c_{2d} & -s_{2d} & 0 & 0 \\ 0 & 0 & 1 & -L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2d}T_{3d} = \text{Rot}(90^\circ, X_{2d}) \cdot \text{Rot}(-\theta_{3d}, Z_{3d})$$

$${}^{2d}T_{3d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & s_{90} & c_{90} & 0 \\ 0 & c_{90} & -s_{90} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{-3d} & c_{-3d} & 0 & 0 \\ c_{-3d} & -s_{-3d} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2d}T_{3d} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -s_{3d} & c_{3d} & 0 & 0 \\ c_{3d} & s_{3d} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{2d}T_{3d} = \begin{bmatrix} -s_{3d} & c_{3d} & 0 & 0 \\ c_{3d} & s_{3d} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Como:

$${}^0T_{3d} = {}^0T_1 \cdot {}^1T_{2d} \cdot {}^{2d}T_{3d}$$

Temos:

$${}^0T_{3d} = \begin{bmatrix} s_1 & c_1 & 0 & 0 \\ c_1 & -s_1 & 0 & 0 \\ 0 & 0 & 1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_{2d} & c_{2d} & 0 & 0 \\ c_{2d} & -s_{2d} & 0 & 0 \\ 0 & 0 & 1 & -L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} -s_{3d} & c_{3d} & 0 & 0 \\ c_{3d} & s_{3d} & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

O resultado da multiplicação das três matrizes acima encontra-se exposto abaixo e é a matriz de transformação direta da câmera direita ${}^0T_{3_d}$.

$${}^3dT_0 = \begin{bmatrix} s_1(s_{2d}s_{3d} + c_{2d}c_{3d}) + c_1(-c_{2d}s_{3d} + s_{2d}c_{3d}) & s_1(-s_{2d}c_{2d} + c_{2d}s_{3d}) + c_1(c_{2d}c_{3d} + s_{2d}s_{3d}) & 0 & -L_1c_1 \\ c_1(s_{2d}s_{3d} + c_{2d}c_{3d}) - s_1(-c_{2d}s_{3d} + s_{2d}c_{3d}) & c_1(-s_{2d}c_{2d} + c_{2d}s_{3d}) - s_1(c_{2d}c_{3d} + s_{2d}s_{3d}) & 0 & L_1s_1 \\ 0 & 0 & -1 & L \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.2 Projeto Físico

As interfaces para aquisição de imagens normalmente utilizadas para transferir as informações de câmeras ou qualquer fonte de sinal de vídeo num computador são conhecidas como *Frame Grabber*. Os *Frame Grabbers* mais modernos são para barramento PCI, que permite uma alta taxa de transferência para a memória RAM do computador, requisito fundamental para as aplicações mais modernas de processamento e análise de imagens. Existem *Frame Grabbers* para câmeras analógicas e digitais, coloridas e monocromáticas, além de câmeras de linha, chamadas *line-scan*¹.

No caso específico do projeto, que deveria capturar duas imagens simultaneamente, fez-se necessária a utilização de dois canais da placa *Frame Grabber*, que dispõe de quatro canais de captura de vídeo na interface *Composite Video*, além de um canal na interface *S-Video*. Entretanto, segundo o próprio *site* da Sensoray [2] (fabricante da placa), a taxa de aquisição de imagens com duas câmeras conectadas à mesma placa cai para oito quadros por segundo², o que tornou necessária a utilização de duas placas de aquisição de imagens no mesmo computador. Uma estrutura alternativa cogitada foi a utilização de dois computadores, cada um equipado com uma placa *Frame Grabber*, que receberia a imagem de uma das câmeras e a pré-processaria. Um dos computadores receberia as imagens pré-processadas do outro via placa de rede e seria responsável pelo processamento final das imagens pré-processadas. Os testes e resultados encontram-se expostos na seção 6.2.

Um esboço da cabeça estéreo idealizada para o projeto é exibido na Figura 3.2, conforme descrito por Bezerra et al[4]. Nele as câmeras estarão posicionadas nas estruturas metálicas em forma de 'U'. O motor responsável pelo movimento de giro em torno do eixo vertical localiza-se na base do 'pescoço' da cabeça estéreo. Os motores responsáveis pelo giro em torno do eixo vertical

¹As câmeras de linha são tipicamente usadas para aplicações que exigem alta velocidade e alta resolução. Os objetos em estudo movem-se numa única direção. Mantendo a câmera de linha fixa, a varredura é feita continuamente numa única linha, colhendo assim as informações.

²Lembrando que o projeto prevê a taxa de 20 quadros por segundo como mínima aceitável.

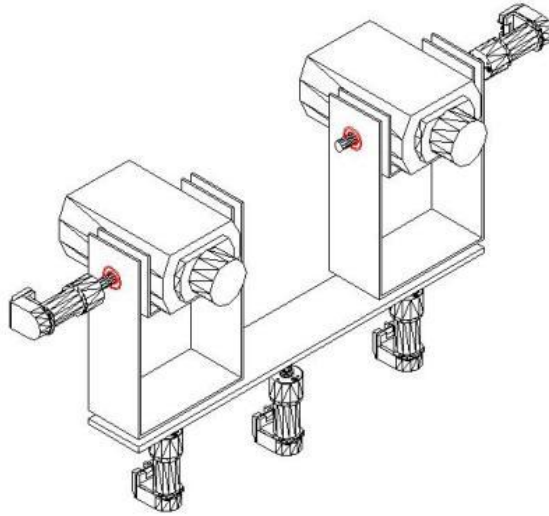


Figura 3.2: Esboço da Cabeça Estéreo

independente para cada câmera localiza-se na base da estrutura de suporte da câmera. Os motores responsáveis pelo movimento de giro em torno do eixo horizontal independente para cada câmera localiza-se junto à lateral externa da estrutura de suporte de cada câmera.

Essa estrutura, foi construída com metais leves, garantindo o baixo peso do conjunto e gerando reduzidos momentos de inércia para os motores das juntas. Com essa configuração, cada motor atua no eixo que passa pelo centro de massa da estrutura movimentada por ele, de forma a reduzir o esforço necessário para o movimento de rotação e permitindo a utilização de motores com menor potência e conseqüente aumento na precisão no funcionamento do sistema.

Os *encoders* encontram-se acoplados diretamente aos motores, permitindo, assim, que a leitura da posição angular de cada motor possa ser aferida e repassada ao programa responsável pelo controle dinâmico da cabeça.

As dimensões máximas da cabeça são 365mm de largura, 110mm de profundidade (sem levar em consideração os cabos que se conectam à parte anterior das câmeras) e 235mm de altura. As dimensões de cada uma das peças do projeto encontram-se no apêndice A.

3.3 Calibração Cinemática Inicial

Espera-se algumas tarefas básicas de um sistema equipado com um braço manipulador robótico. Como por exemplo a capacidade de alcançar determinado ponto dentro de seu espaço de trabalho, ou de evitar tocá-lo. De assumir determinada pose no espaço ou ainda seguir uma trajetória pré-determinada pelo programador.

Para o cumprimento de todas essas funções é imprescindível que o sistema conheça a configuração do braço manipulador a cada instante, isto é, as dimensões de seus elos e a posição (angular ou translacional) de cada uma de suas juntas.

Com essas informações é possível reconstruir a pose do braço no espaço, bem como a posição e orientação tri-dimensional da ferramenta.

Cabe aqui uma explicação acerca de sensores de posição, que podem ser absolutos ou incrementais.

- Os Sensores Incrementais possuem como saída um pulso que indica que a posição atual se distanciou da posição anterior, positiva ou negativamente;
- Os Sensores Absolutos possuem como saída a posição absoluta em relação a um ponto fixo.

Acontece que no início do funcionamento do sistema, quando acabam de ser ligados os sensores de posição de cada junta, surge a necessidade de se conhecer as suas posições reais.

No caso, os sensores absolutos são capazes de sinalizar a posição real assumida por aquela junta naquele instante. Ao contrário, no caso dos sensores incrementais, independente da posição real, o indicado por eles será "zero" (admitindo que a junta encontra-se parada), pois não está havendo incremento de posição na junta.

No segundo caso, é clara a necessidade de se conhecer a posição inicial para o correto funcionamento do sistema, pois sem a posição inicial de nada serve a indicação incremental dos sensores. A determinação dessa posição, dá-se o nome de calibração cinemática do robô.

A calibração cinemática é uma prática recomendada mesmo quando são utilizados sensores absolutos. Esse procedimento se justifica pois o sensor pode ter saído de sua posição original com o tempo, perder precisão, ou ainda ter sido mal-posicionado. A calibração cinemática insere no sistema um *off-set* que se propõe a corrigir os erros da medição inicial dos sensores.

No nosso caso, o de uma cabeça robótica, o conhecimento preciso do posicionamento e da orientação de ambas as câmeras é essencial para a precisão do algoritmo de visão estéreo e conseqüentemente para a qualidade da reconstrução tri-dimensional do ambiente em volta do robô.

Dessa forma, o processo de calibração cinemática da cabeça deve ser feito.

Uma estratégia bastante simples, mas que tem o custo financeiro como desvantagem, é a instalação de sensores de fim-de-curso em todas as juntas da cabeça.

Equipado com tais sensores, o robô deve ser programado para, no início de sua operação, acionar cada um de seus motores em determinado sentido até que os respectivos sensores de fim-de-curso sejam acionados, marcando a posição inicial (conhecida) das juntas.

A partir desse instante deve-se utilizar as indicações dos sensores de posição como incrementos dessa posição inicial.

Até a escrita desse documento nem os sensores de fim de curso nem os sensores de posição haviam sido instalados e os resultados, por esse motivo, não foram aferidos. Espera-se contudo, pela simplicidade do método, que a sua implementação ocorra com sucesso.

Capítulo 4

Implementações Computacionais

Uma vez esgotado o assunto do modelo cinemático, do projeto físico e da construção propriamente dita da cabeça estéreo como artefato físico, deve-se voltar o foco central do trabalho à parte computacional do projeto.

Devemos então dar ênfase aos aspectos lógicos do projeto e às suas implementações. Assim sendo, três são os temas que daqui em diante tomam posição de destaque: as estratégias normalmente utilizadas na captura de imagens provenientes de câmeras digitais; a maneira como tais imagens são usualmente pré-processadas para sua posterior utilização e a estratégia adotada pelos integrantes do projeto RoboSense que garante à sua cabeça estéreo um funcionamento adequado.

Neste capítulo, será abordado o paradigma normalmente empregado no processo de captura de imagens de câmeras de vídeo pelo computador e do pré-processamento dessas imagens recém-capturadas. Serão confrontadas ainda as peculiaridades desse paradigma com a realidade que o projeto impõe. Será esmiuçada a estratégia de captura e pré-processamento de imagens utilizada efetivamente no projeto, bem como estudados também as etapas de tal estratégia.

A correção da distorção barril introduzida pelas câmeras e a transformação de imagem em multi-escala têm seu funcionamento mais aprofundadamente estudado nas seções 4.3.1 e 4.3.2, respectivamente.

Um problema existente na maioria dos projetos computacionais que se utilizam de visão estéreo é o elevado tempo de execução dos algoritmos de pré-processamento das imagens captadas por suas câmeras. A agilidade das rotinas se torna essencial, uma vez que o processo acontece em tempo real e exige do sistema uma taxa mínima de quadros processados por segundo.

4.1 Esquema proposto

Segundo **Gonzalez e Woods**[3], o interesse em métodos de processamento de imagens digitais decorre de suas duas principais áreas de aplicação: melhoria de informação visual para interpretação humana e o processamento de dados de cenas para percepção automática através de máquinas. No nosso caso, o de interpretação de cenas por máquinas, o procedimento comumente usado é a extração de características relevantes para uma futura tomada de decisão. Informações como variação de cores ou sua permanência em uma região da imagem pode indicar presença ou ausência de objetos naquela região. No caso de vídeos, que nada mais são que uma seqüência de imagens, a variação das tonalidades (ou cores) de uma determinada região no tempo pode indicar movimentação de objetos naquela região ou ainda mudança de iluminação, por exemplo. A identificação de regiões da imagem contendo texturas também pode ser de grande utilidade em determinadas aplicações. Para cada aplicação um conjunto diferente de características é necessário.

Deve ser ressaltado que todas essas características encontram-se sobrepostas na imagem original, que muitas vezes pode estar distorcida, borrada, ou conter alguma outra forma de ruído ou imperfeição que pode vir a dificultar enormemente a tarefa de extração das informações da imagem.

A abordagem mais utilizada na visão computacional é submeter as imagens exatamente como são capturadas das câmeras, ainda brutas ou *“in natura”*, a uma etapa chamada pré-processamento. Nessa etapa, as imperfeições nocivas ao processo devem ser eliminadas, assim como os ruídos danosos. No caso do RoboSense, as câmeras adquiridas para o projeto distorcem a imagem capturada, o que deve ser corrigido na fase de pré-processamento. Uma outra função da etapa de pré-processamento de imagens é a extração das características relevantes para o problema isoladamente e, a partir dessas características, gerar novas imagens contendo exclusivamente tais características. A extração de bordas em uma imagem é um exemplo de características que podem ser extraídas de imagens e é mostrada na figura 4.1, onde pode-se perceber a imagem original e a imagem obtida a partir da extração das bordas da imagem original.

Para o projeto, algumas características são consideradas importantes e devem ser extraídas a partir das imagens originais. Algumas das características relevantes para o projeto são as bordas horizontais, bordas verticais e movimento. Os métodos para a extração de características relevantes e construção de nova imagem a partir delas serão tratados na seção 4.3.

Entretanto, como será explicado na seção 4.3, a extração dessas características para construção de novas imagens é uma tarefa extremamente custosa em processamento e, conseqüentemente, em tempo. A taxa de captura requerida pelo projeto é de 20 quadros por segundo, o que exige todo



Figura 4.1: Imagem original (esquerda). Imagens obtidas a partir da extração das bordas verticais e horizontais da imagem original (superiores à direita). Combinação das bordas verticais e horizontais e superposição de bordas na imagem original (inferiores à direita).

o processamento de cada quadro em um intervalo de $1/20s = 50ms$, que é o intervalo de tempo no qual um quadro é considerado atual e depois do qual um novo quadro é capturado.

No esquema proposto, acrescentou-se uma etapa à fase de pré-processamento das imagens capturadas.

A partir da imagem original, já livre de distorção, são construídas cinco pequenas imagens¹, num esquema de multi-resolução baseado na retina humana.

Nela, a porção da imagem capturada pela sua parte central (fóvea) tem uma maior riqueza de detalhes que a porção da imagem capturada por sua parte periférica. No nosso esquema cinco são as imagens, com cinco diferentes resoluções. Cada imagem representa a região central da imagem, mas cada uma em um nível de *zoom* diferente. A figura 4.4² exibe o esquema adotado, nela percebe-se as áreas representadas por cada imagem. Quanto maior a região representada pela imagem, menor sua resolução.

Ao final do processo temos cinco pequenas imagens ($5 \times 32 \times 32$ *pixels*) que representam a imagem original (304×294 *pixels*). Como as cinco imagens têm menos *pixels* (5120) que a original (89376), a extração de características delas é mais rápida. As imagens criadas nesse método mantêm as dimensões da imagem em multi-resolução. Os resultados experimentais encontram-se expostos na seção 6.3.

¹Enquanto a imagem original tem dimensões 304×294 *pixels*, as cinco pequenas imagens têm dimensões 32×32 *pixels*.

²Imagem retirada de "Um Modelo para Redução e Abstração de Dados em uma Cabeça Estéreo"[4].

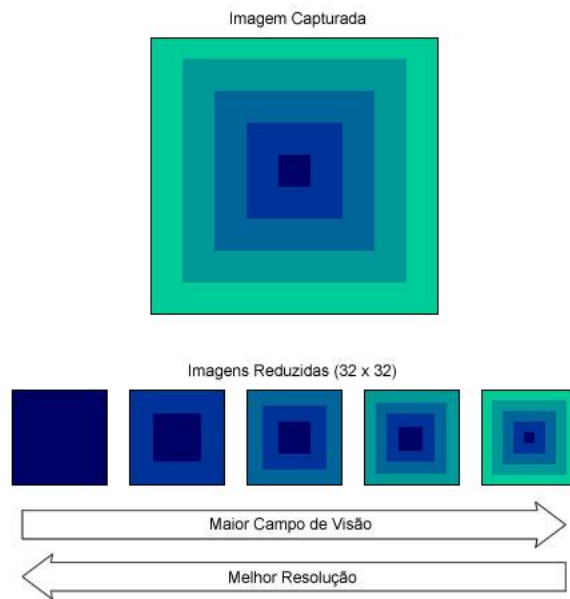


Figura 4.2: Construção de imagem em multi-resolução.

Uma vez explicada a fase de pré-processamento, deve-se agora atentar para o esquema como um todo. A figura 4.3 exibe a arquitetura proposta de aquisição das duas imagens e o processamento (desde o pré-processamento até as tarefas de alto nível) das imagens. Nela estão explícitas as aquisições das imagens pelas câmeras esquerda e direita da cabeça, que são submetidas ao pré-processamento explicado para só então serem submetidos ao programa de *match*, que é o responsável pelo cruzamento das informações extraídas das imagens capturadas pelas duas câmeras. O resultado do *match* servirá de referência para os comportamentos de alto nível do robô, como navegação, execução de tarefas e identificação de objetos.

4.2 Captura dos Quadros

Para que as imagens captadas pelas câmeras de vídeo possam ser armazenadas na memória do computador é necessário haver um *software* que desempenhe essa função junto à placa de aquisição (*Frame Grabber*).

O programa implementado para o projeto tem, entre suas funcionalidades, a incumbência de adquirir, junto à placa de aquisição, o último quadro capturado e disponibilizá-lo para as operações que a seguem, como a correção de distorção, a construção da imagem em multi-resolução e a extração de características relevantes da imagem. Implementado em C++ sobre plataforma Linux Red Hat 6.0, o software faz uso, para comunicação com a placa de aquisição, da biblioteca

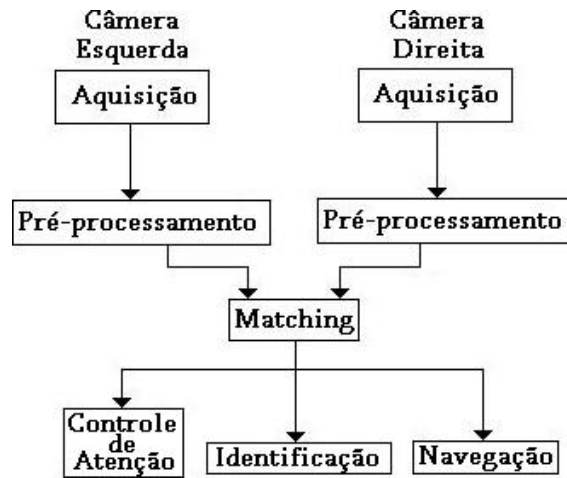


Figura 4.3: Arquitetura proposta para a aquisição e processamento de imagens pelo RoboSense.

Video4Linux, projetada e desenvolvida pela GNU.

4.3 Pré-processamento dos Quadros

Após a captura das imagens e sua aquisição por um computador, é necessário extrair delas informações úteis e relevantes para as tomadas de decisão do programa que implementará a inteligência do robô. Entretanto, esse processo não é dos mais simples e por essa razão não ocorre em uma única etapa. Para simplificar essa tarefa, costuma-se extrair das imagens algumas das suas características mais importantes e algumas das informações que elas carregam para formar, a partir dessas características e informações, novas imagens, que servirão, essas sim, como parâmetros de entrada nos processos de tomada de decisão. No caso do RoboSense os programas que controlarão a navegação do robô e a execução de tarefas receberão essas imagens chamadas pré-processadas.

Como já foi dito na seção 4.1, após a captura da imagem bruta, segue-se uma série pré-processamentos à imagem a fim de prepará-la para a extração de informações e de extrair efetivamente tais informações.

No caso do RoboSense, antes da extração de informações da imagem bruta, é necessário que seja corrigida a distorção causada pela câmera na imagem e antes da construção uma imagem em multi-resolução equivalente à original.

A partir da imagem em multi-resolução são extraídas as bordas horizontais, bordas verticais e movimento. Imagens criadas a partir dos filtros Laplaciano e Gaussiano também são necessárias e devido à semelhança na estratégia de construção das imagens dessas cinco diferentes características,

trataremos os quatro casos conjuntamente.

As bordas horizontais e verticais são extraídas a partir do uso do filtro Gradiente nessas duas direções. A combinação das informações das duas imagens resultantes, conforme Jain [6], informa a direção e a "intensidade" das bordas de uma imagem e servirá, juntamente com as demais características, no reconhecimento de objetos e formas.

O filtro Gaussiano, ou filtro de borrimento, tem como resultado de sua aplicação a atenuação de ruídos de alta frequência e por essa razão se faz necessário no projeto.

Já o filtro Laplaciano, derivativo de segunda ordem, tem a capacidade, segundo Jain[6], de reconhecer as regiões onde as bordas são mais suaves. O Laplaciano utilizado no projeto foi o Laplaciano da Gaussiana.

Todos os filtros acima são explicados na seção 5.1.

Para a extração das características supracitadas aplicamos na imagem uma técnica chamada *filtragem no domínio da frequência* ou uma equivalente chamada *filtragem espacial*. Como essa última foi a técnica utilizada no projeto, ela será a explicada na seção 5.1, em detrimento àquela.

Duas outras características são igualmente importantes para o projeto, entretanto a forma como elas são conseguidas diferem das anteriores, o que motiva seu tratamento diferenciado. São elas: o movimento e a correlação cruzada.

A característica do movimento, explorada na seção 5.2, indica quais as regiões da imagem que se modificaram em relação ao instante anterior e que modificação ocorreu.

A correlação cruzada, ao contrário de todas as características anteriores, é extraída a partir da combinação das imagens capturadas pelas duas câmeras simultaneamente e tem a finalidade de quantificar a semelhança entre regiões das duas imagens. Uma boa precisão dessa característica é de fundamental importância para o algoritmo de visão estéreo do robô e será melhor estudada na seção 5.3.

4.3.1 Correção de Distorção Barril

Conforme mencionado anteriormente, as câmeras utilizadas na cabeça estéreo não possuem ajuste automático de seus parâmetros intrínsecos (zoom, foco, características das lentes). Portanto, a necessidade da calibração inicial surge graças à ocorrência de distorção espacial adicionada às imagens pelas câmeras, quando de sua aquisição. Assim sendo, a esta retirada de distorção deve acontecer antes de qualquer outro pré-processamento, devolvendo com isso as características reais à cena.

Conforme visto acima, no modelo adotado por Trucco [24], a distorção pode ser representada

segundo a equação 4.1, onde X , Y , X_d e Y_d são as posições horizontal e vertical de cada *pixel* na imagem distorcida e corrigida, respectivamente. Nessa mesma equação, K_1, K_2, \dots, K_n são os parâmetros que acentuam ou atenuam a distorção. Ou seja, o conjunto dos parâmetros K_s , segundo esse modelo, são suficientes para representar qualquer distorção do tipo barril.

$$X = X_d(1 + k_1r^2 + k_2r^4 + \dots + k_sr^{2^s})Y = Y_d(1 + k_1r^2 + k_2r^4 + \dots + k_sr^{2^s}) \quad (4.1)$$

Como os parâmetros K_s dependem das características intrínsecas das lentes e da própria câmera e essas características não podem ser alteradas, pode-se determinar os valores para os coeficientes K_s em testes anteriores ao funcionamento do robô e fixá-los nestes valores sem perda de desempenho.

Chamamos esse tipo de procedimento de calibração *a frio*, uma vez que quando o robô é posto em funcionamento ele já carrega consigo os seus parâmetros intrínsecos, não necessitando da calibração *a quente*, que é a metodologia que obtém os parâmetros intrínsecos no início do funcionamento do equipamento.

Foram realizados testes com as câmeras com o intuito de definir os parâmetros K_s que melhor representassem a distorção adicionada.

Os testes se basearam na aquisição da imagem de um tabuleiro de xadrez disposto sobre o plano normal à linha central da imagem. Como o tabuleiro é retangular, deveria se perceber na imagem as suas bordas como sendo retas. Ao invés disso elas eram representadas como arcos tão mais curvos quanto maior a sua distância em relação ao centro da imagem.

A partir daí foi implementado o algoritmo de correção de distorção barril apresentado por Trucco[24], estimados valores para os parâmetros K_s e a ele foi submetida a imagem distorcida. Por meio de ajuste contínuo dos parâmetros baseado nos resultados obtidos, foram estabelecidos as constantes das câmeras que melhor representam a distorção por elas introduzidas às imagens.

De posse desses valores, é ajustado o algoritmo de correção de distorção do tipo barril que funcionará no RoboSense. A partir dele serão corrigidas as imagens antes de qualquer outro pré-processamento.

4.3.2 Construção de Imagem em Multi-resolução

Nessa etapa, após os ajustes iniciais de calibração e correção de distorção, têm-se as imagens originais com as imperfeições impostas pelas câmeras já corrigidas.

A partir das imagens tratadas, deve-se então começar a etapa de extração de características importantes e construção de novas imagens que as contenham. Entretanto, a eficiência em tempo

da extração de características pelo método da *Filtragem Espacial* depende das dimensões da imagem. Testes foram realizados com a imagem original, cujas dimensões são 304×294 *pixels*, e mostraram ser necessários $47ms$ para a extração de algumas das características desejadas. Como a taxa mínima de processamento é de 20 quadros por segundo, a utilização de $47ms$ apenas na extração de algumas características de um quadro torna o projeto inviável, pois restaria às demais etapas (aquisição, correção de distorção, extração de algumas outras características e ainda o processo de inteligência do robô) apenas $3ms$.

A fim de contornar essa limitação, resolveu-se adotar um esquema de multi-resolução para as imagens adquiridas pelas câmeras. Tal esquema consiste em transformar cada imagem captada em cinco imagens de dimensões 32×32 *pixels* com diferentes resoluções que representam a original. A partir dessas cinco pequenas imagens serão extraídas as características desejadas.

Conforme descrito em "Um Modelo para Redução e Abstração de Dados em uma Cabeça Estéreo"[4], o processo utilizado na transformação das imagens em multi-resolução é o seguinte: no nível de maior resolução (e conseqüentemente menor área de cobertura da imagem original) a obtenção da imagem reduzida é simplesmente a extração direta da porção central da imagem original.

Para os demais níveis de resolução, um método diferente é utilizado. Nesses casos, cada imagem reduzida será formada por um processo de amostragem de *pixels* combinada a uma operação de média sobre alguns dos *pixels* da vizinhança do ponto correspondente na imagem original.

Isso é feito deslocando-se uma máscara de dimensões $\mathbf{h} \times \mathbf{h}$ pela região de interesse a intervalos de \mathbf{h} *pixels* na horizontal e \mathbf{h} *pixels* na vertical. Na primeira amostragem centra-se a máscara em um *pixel* **P1**, na próxima amostragem, em um outro *pixel* **P2** distante horizontalmente \mathbf{h} *pixels* de **P1** e assim por diante, até que se obtenha um total de 32 *pixels*. Para evitar o efeito indesejado de ruídos na aquisição da imagem reduzida, uma média simples entre o *pixel*-alvo ($\mathbf{P}(\mathbf{x}, \mathbf{y})$) e os *pixels* de sua vizinhança horizontal ($\mathbf{P}(\mathbf{x} + \mathbf{subh}, \mathbf{y})$ e $\mathbf{P}(\mathbf{x} - \mathbf{subh}, \mathbf{y})$) e vertical ($\mathbf{P}(\mathbf{x}, \mathbf{y} - \mathbf{subh})$ e $\mathbf{P}(\mathbf{x}, \mathbf{y} + \mathbf{subh})$), onde **subh** é um terço do valor da dimensão \mathbf{h} ³.

Tal procedimento é executado pelo algoritmo abaixo:

Na figura 4.4 encontra-se exposta uma imagem adquirida pela câmera com dimensões 304×294 *pixels*, bem como uma montagem feita a partir das cinco imagens nas diferentes resoluções, explicitando a perda de qualidade na imagem em multi-resolução e também as cinco imagens obtidas a partir da original pelo método descrito.

Os resultados obtidos com a extração de características a partir das cinco imagens pequenas

³Caso \mathbf{h} não seja múltiplo de três, deve-se usar o múltiplo de três imediatamente inferior a ele, assim, garantimos que **subh** seja um valor inteiro.

Entrada: Imagem Im , Nível de resolução N

Saída: SubImagem $SubIm$;

Calcular h ;

Calcular $subh$;

for $i = 0$ até $i = 32$ **do**

for $j = 0$ até $j = 32$ **do**

$SubIm(i, j) = (Im(i*h, j*h) +$
 $+ Im(i*h + subh, j*h) +$
 $+ Im(i*h - subh, j*h) +$
 $+ Im(i*h, j*h + subh) +$
 $+ Im(i*h, j*h - subh)) / 5;$

end for

end for

encontra-se no capítulo 6.

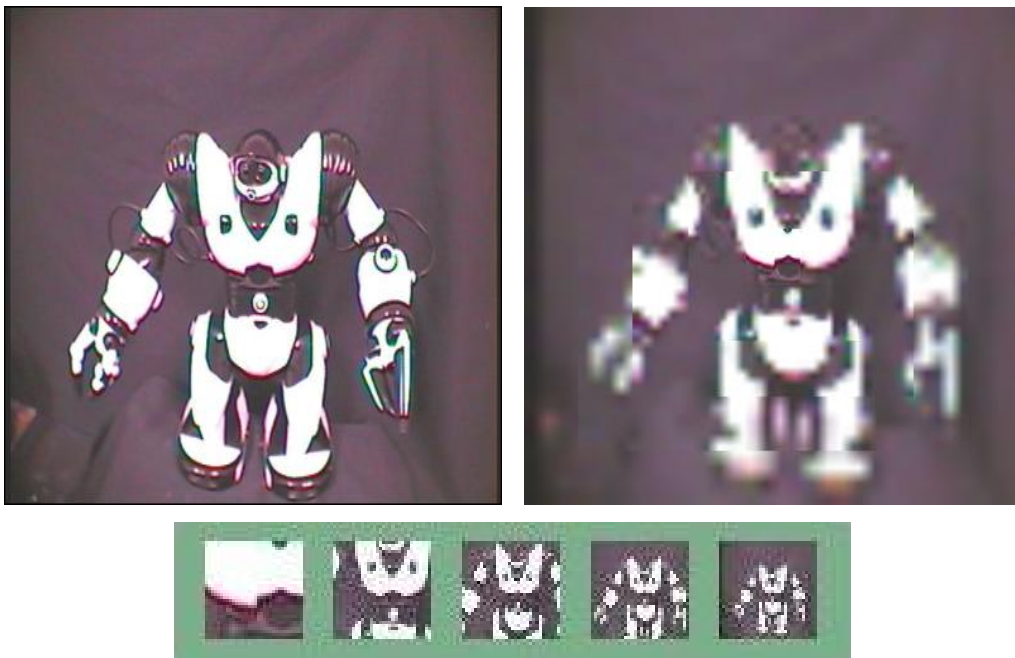


Figura 4.4: **Acima:** (Esquerda) Imagem Original 304×294 pixels. (Direita) Montagem explicitando a perda de qualidade na imagem em multi-resolução. **Abaixo:** Imagens em multi-resolução ($5 \times 32 \times 32$ pixels).

Capítulo 5

Extração de Características das Imagens

Uma vez concluída a fase de multi-resolução, inicia-se a fase de retirada de características das imagens. A filtragem de imagens é a técnica utilizada na extração de algumas das características neste trabalho. Detectamos também movimento e sugerimos o cálculo de disparidade estéreo usando a técnica de *Correlação Cruzada*, que é largamente utilizada na estimativa da posição tri-dimensional (no espaço real) de um ponto contido nas imagens captadas pelas câmeras de um sistema computacional como o RoboSense. Tal técnica é melhor explicada na seção 5.3.

Todos os algoritmos envolvidos na extração de características das imagens foram implementados usando a linguagem de programação C++.

5.1 Filtragem de imagens

As técnicas de filtragem são transformações da imagem *pixel* a *pixel* que não dependem apenas do nível de luminância de um determinado *pixel*, mas também do valor dos níveis de luminância dos *pixels* vizinhos, na imagem original.

O processo de filtragem é feito utilizando-se matrizes denominadas máscaras que são aplicadas sobre a imagem original. No processo, a máscara se desloca por sobre a imagem original e alimenta os valores de uma segunda imagem que vai sendo formada a partir do resultado da composição dos valores dos pesos da máscara com a região filtrada da imagem original.

Como a nossa imagem em multi-resolução é armazenada sob a forma de 5 imagens, a filtragem utilizada no projeto RoboSense dá sobre cada uma delas, que pré-processadas, são compostas

por 3 matrizes numéricas de 32 linhas por 32 colunas.

No projeto uma máscara de 3×3 cujos elementos $MASC_i$ são os pesos na operação de filtragem, é utilizada conforme exibido na figura 5.1.

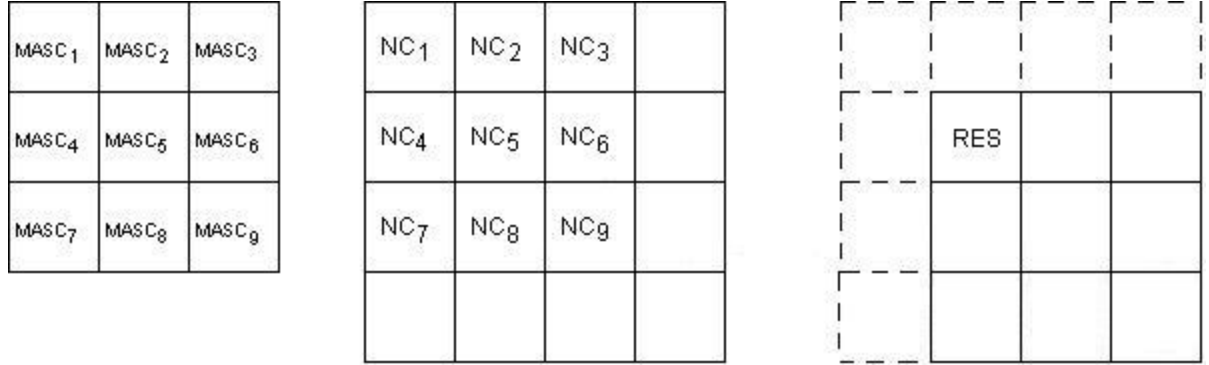


Figura 5.1: Máscara 3×3 com centro na posição $(\mathbf{i}, \mathbf{j})=(2, 2)$.

A aplicação da máscara 3×3 com centro na posição (\mathbf{i}, \mathbf{j}) , sendo \mathbf{i} o número de uma dada linha e \mathbf{j} o número de uma dada coluna sobre a imagem, consiste na substituição do valor do *pixel* na posição (\mathbf{i}, \mathbf{j}) por um novo valor, calculado pela expressão 5.1.

$$RES = \frac{1}{K} \cdot \sum_{i=1}^9 NC_i \cdot MASC_i \quad (5.1)$$

Sendo K uma constante geralmente utilizada para a normalização da máscara, o que garante a manutenção das características estatísticas, tais como a média, da imagem original na imagem filtrada. Para tanto $K = \sum_{i=1}^9 MASC_i$.

O resultado da aplicação de um filtro é uma nova imagem, cujas linhas e colunas iniciais e finais necessitam de um tratamento especial, uma vez que não possuem vizinhança superior, inferior ou lateral. As imagens resultantes das filtrações do projeto têm suas linhas e colunas iniciais e finais preenchidas com valor 0 (zero), a cor preta.

A utilização de filtros de borramento é bastante comum na fase de pré-processamento e visa a redução de ruídos, que podem vir a atrapalhar as etapas seguintes. Dentre os filtros mais comuns filtros suavizadores utilizados está o filtro gaussiano, que na sua versão bidimensional isotrópica (circularmente simétrica) utiliza a fórmula mostrada na equação 5.2 para efetuar a construção de sua máscara, sendo σ^2 o desvio padrão da curva de Gauss. Na nossa utilização, o desvio padrão quantifica a variação de intensidade luminosa entre *pixels* vizinhos na imagem.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5.2)$$

A máscara 3×3 do filtro gaussiano utilizado neste trabalho pode ser vista na Tabela 5.1.

$$\frac{1}{16} \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

Tabela 5.1: Filtro gaussiano

A detecção e extração de bordas em uma imagem são fundamentais para a construção dos vetores de descritores que irão ser utilizados na identificação e classificação dos objetos em cena. Os métodos mais comuns para realizar esta tarefa utilizam o gradiente. A magnitude do gradiente de uma imagem $f(x, y)$, na posição (x, y) , é dada pela equação 5.3. Implementamos também o gradiente do Gaussiano, como opção para aumentar a blindagem do algoritmo contra variações substanciais de intensidade entre *pixels* vizinhos ocasionadas por ruídos.

$$\nabla f = \text{mag}(\nabla \mathbf{f}) = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad (5.3)$$

Para determinar a direção resultante do vetor gradiente em um dado pixel (x, y) , utiliza-se a equação 5.4 que retorna o valor do ângulo em relação ao eixo \mathbf{x} .

$$\alpha(x, y) = \tan^{-1} \left(\frac{G_y}{G_x} \right) \quad (5.4)$$

Para a implementação dos filtros de gradiente, foram escolhidos os operadores de Sobel por apresentarem a característica de incorporarem um efeito de suavização ao processo de derivação, fornecendo, assim, um melhor resultado. As tabelas 5.2 e 5.3 apresentam as máscaras utilizadas no cálculo dos gradientes nas direções \mathbf{x} e \mathbf{y} , respectivamente.

-1	-2	-1
0	0	0
1	2	1

Tabela 5.2: Filtro gradiente na direção \mathbf{x}

-1	0	-1
-2	0	-2
-1	0	-1

Tabela 5.3: Filtro gradiente na direção \mathbf{y}

O operador laplaciano de uma imagem é definido pela equação 5.5. Geralmente utilizado em conjunto com os filtros de gradiente, este filtro facilita algumas tarefas de segmentação na imagem. Aqui implementamos também a opção de borramento em conjunto com o Laplaciano, isto é, o uso do filtro Laplaciano do Gaussiano, para permitir a redução de ruídos e altas frequências. Esta opção é a que está sendo utilizada.

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (5.5)$$

A máscara utilizada no cálculo do Laplaciano da imagem é apresentada na Tabela 5.4.

0	-1	0
-1	4	-1
0	-1	0

Tabela 5.4: Filtro laplaciano

5.2 Detecção de Movimento

A imagem *detecção de movimento* é o resultado de uma comparação do quadro atual com o imediatamente anterior a ele. Ela indica onde e com que intensidade houve alteração de cor entre os dois últimos quadros adquiridos. A partir de tal imagem pode-se observar o que mudou na cena durante o intervalo de tempo transcorrido entre as duas últimas capturas, informação extremamente útil ao robô.

A construção da imagem *detecção de movimento* é conseguida através da subtração *pixel a pixel* do quadro atual no quadro anterior, sendo o resultado de cada operação, em módulo, armazenado no *pixel* de mesma coordenada (x, y) da imagem resultado.

5.3 Correlação Cruzada para Disparidade

Os algoritmos de visão computacional têm por necessidade o conhecimento da localização de um objeto em ambas as imagens, pois a estimativa de posicionamento do objeto na cena se baseia em tais informações.

Com o objetivo de encontrar regiões semelhantes nas duas imagens, utiliza-se a técnica de *Correlação Cruzada*, que compara uma região de uma imagem com toda a outra imagem, indicando a região que mais se assemelha a ela.

A técnica consiste de uma varredura por toda a imagem pesquisada de uma janela com os valores dos *pixels* da vizinhança a ser comparada. O resultado do algoritmo é uma imagem cujas intensidades de cada *pixel* reflete o grau de semelhança da janela com a vizinhança pesquisada centrada nele. Quanto maior o valor do *pixel*, maior a semelhança entre sua vizinhança e a janela-padrão.

No caso da cabeça estéreo do projeto operando em modo independente, a correlação cruzada não é necessária, uma vez que as imagens não têm relação entre si, pois focalizam cenas teoricamente diferentes.

Sua utilidade se dá, entretanto, no modo dependente da cabeça estéreo, quando as duas câmeras apontam para o mesmo objeto ou ponto. Nesse caso pode-se assumir que o deslocamento de um objeto de uma imagem para a outra acontece somente no sentido horizontal, uma vez que as câmeras têm a mesma angulação do movimento de *Pan* (ou seja, θ_{3d} e θ_{3e} têm o mesmo valor).

Assim sendo, para localizar-se um objeto de uma imagem na outra imagem, basta executar uma varredura na linha onde o objeto se encontra na imagem original, o que reduz significativamente o processamento da operação.

No projeto, devido a perda de precisão acrescentada pela criação da imagem em multi-resolução e para reforçar a precisão da informação coletada, adotar-se-á um esquema duplo de *Correlação Cruzada*. Nele, o procedimento será aplicado da imagem esquerda em relação a direita e vice-versa. Com isso, espera-se minimizar o erro decorrente da redução da resolução da imagem original.

Devido ao acionamento da cabeça não ter estado disponível quando do desenvolvimeto deste trabalho, resolveu-se excluir do mesmo a implementação dessa técnica, que deve ser absorvida pelas fases posteriores do projeto RoboSense.

Capítulo 6

Experimentos e Resultados

Foram realizados vários experimentos, usando o simulador cinemático da cabeça estéreo, que será melhor explicado aqui, incluindo sua função, os modos de funcionamento e algumas características, além de mostrarmos algumas de suas imagens.

Um dos desafios enfrentados foi garantir que a taxa de aquisição das imagens pelas câmeras fosse satisfatória. A estratégia adotada, bem como os resultados conseguidos encontram-se expostos adiante.

Foram feitos experimentos e os resultados conseguidos com a aplicação da metodologia proposta para a criação de imagens em multi-resolução e extração de suas características foram confrontados com os resultados obtidos com a aplicação do método convencional.

6.1 Simulador

O simulador da cabeça estéreo foi implementado em OpenGL com o intuito de simular sua cinemática e dinâmica. O modelo cinemático já encontra-se implementado, restando como trabalho futuro a implementação do modelo dinâmico.

A programa foi implementado pelo então graduando Jose Savio A de Sousa Segundo, como parte do seu trabalho de Iniciação Científica, tendo o apoio da nossa assessoria.

A cabeça estéreo foi concebida de forma a poder funcionar segundo dois modos. No primeiro, os movimentos das câmeras são dependentes entre si e o centro das duas imagens devem conter o objeto selecionado, enquanto que no segundo as câmeras se movem de forma independente, podendo capturar imagens de diferentes cenas.

O simulador pode funcionar segundo os dois modos de funcionamento da cabeça.

No modo de movimentação dependente o usuário do simulador navega com o objeto a ser

focalizado no espaço tri-dimensional, ambas as câmeras se movimentam de forma a focalizar o objeto. Nesse modo as linhas que passam pelo centro da imagem de cada câmera se interceptam no centro do objeto focalizado e não são controladas diretamente pelo usuário. Ao invés disso, o usuário tem o controle via teclado da posição tri-dimensional do objeto na cena, que é seguido pelas duas câmeras da cabeça de forma automática, mediante a lógica interna implementada no simulador.

No modo de movimentação independente o usuário controla a posição angular de cada junta do robô. Dessa forma cada câmera tem a possibilidade de focalizar um objeto diferente. A direção das linhas que atravessam o centro do objeto visualizado por cada câmera pode ser controlada diretamente pelo usuário.

A imagem exibida no simulador reflete a primeira concepção da cabeça. A movimentação das partes que compõem a cabeça estéreo (sua base, pescoço e câmeras) ocorre de acordo com o seu modo de operação e é feita de forma a simular o movimento real.

O modelo cinemático que dá suporte ao simulador é o modelo adotado na construção da cabeça e a movimentação executada por cada uma de suas partes.

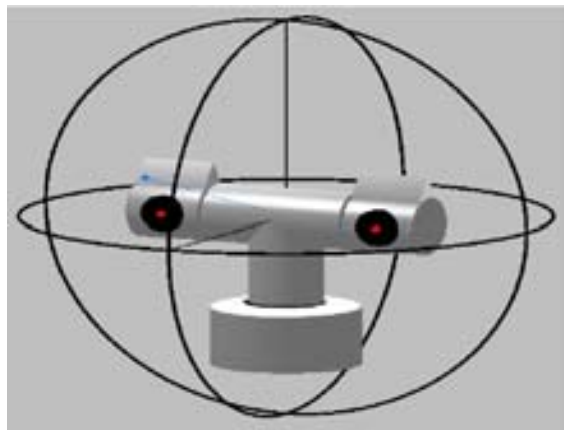


Figura 6.1: Simulador da Cabeça Estéreo

A figura 6.1 exibe uma imagem do simulador com um esboço da cabeça do início do projeto, antes de ser levado em consideração o material para a construção da mesma. Esse modelo, apesar de visualmente diferente, possui os mesmos graus de liberdade e mesmo modelo cinemático que o esboço para a construção da cabeça estéreo robótica.

No desenho inicial, usado até hoje pelo simulador, as câmeras estariam montadas nas extremidades laterais da estrutura, no interior da região cilíndrica, representados por uma circunferência preta. O motor responsável pelo movimento de giro em torno do eixo vertical localiza-se na base da cabeça estéreo. Os motores responsáveis pelo movimento de giro em torno do eixo horizontal

independente para cada câmera localizam-se entre as câmeras. Os motores responsáveis pelo giro em torno do eixo vertical independente para cada câmera localiza-se em uma saliência em cima de cada câmera.

Nessa composição da cabeça estéreo robótica, baseada em cilindros, destaca-se na visualização a representação das câmeras, o movimento de *pan* é representado através da movimentação de um ponto vermelho na lente da câmera, indicando o grau de vergência da câmera e uma linha vermelha, com opção para ser habilitada ou não. A linha vermelha indica o vetor do ponto central da visualização de cada câmera. Isto facilita a verificação da posição angular assumida por cada um dos motores das juntas da cabeça estéreo como também a visualização de o que as câmeras estão focando.

Uma abordagem proposta para facilitar a transformação de coordenadas no espaço tri-dimensional para o espaço de juntas do robô é a representação de pontos no espaço 3D utilizando coordenadas polares.

O centro do sistema de coordenadas polares foi escolhido de forma a simplificar ao máximo a transformação e, por isso, encontra-se sobre o ponto de encontro dos eixos de giro do pescoço e dos movimentos de TILT de cada olho.

O sistema de coordenadas esféricas é composto por 3 parâmetros, que aqui chamamos de α , β e raio. Ou seja, qualquer posição assumida pelo objeto a ser visualizado pelas câmeras pode ser representado pelo vetor α , β e raio. Os dois primeiros quantificam a distância angular entre um plano de referência e o ponto a ser localizado.

No nosso caso α assume o valor de zero quando o objeto encontra-se a sobre o plano X exibido na figura 6.2 e tem o seu valor aumentado a medida que o ponto se afasta desse plano. Da mesma forma, β assume o valor de zero quando o objeto encontra-se a sobre o plano Y exibido na figura 6.2 e tem o seu valor aumentado a medida que o ponto se afasta desse plano. O raio representa a distância que o objeto encontra-se do centro do sistema de coordenadas polares e aumenta a medida que o objeto se afasta do mesmo.

Dessa forma, quando o simulador está em modo de movimentação dependente, o valor de α é diretamente assumido pela junta do pescoço e o valor de β , pelas juntas de TILT das câmeras da cabeça.

A única transformação não-direta entre o sistema de coordenadas polares proposto e o espaço de juntas do robô é a das juntas de vergência das câmeras. Entretanto o seus valores são obtidos de forma bastante simplificada por triangulação, dependendo apenas do parâmetro raio e da distância entre o centro de cada câmera para o centro de coordenadas, que é fixa e conhecida.

No modo de simulação de movimentação independente, como foi comentado, o que o usuário

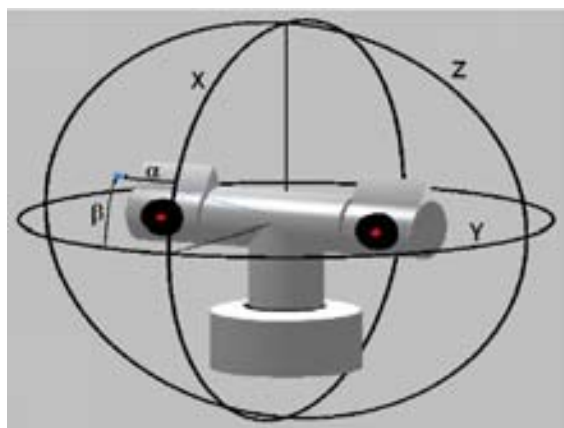


Figura 6.2: Parâmetros α e β no Simulador.

controla é a posição angular de cada junta da cabeça, sendo desnecessária a transformação entre a posição tri-dimensional focalizada por cada câmera e o espaço de juntas do robô.

Para facilitar a visualização tri-dimensional na tela, são desenhados três círculos ao redor da cabeça, representando de cada um dos planos X, Y e Z. O valor do raio dos círculos foi escolhido grande o suficiente para que a cabeça nunca os toque e pequeno o suficiente para não extrapolar a região do espaço exibida pelo simulador.

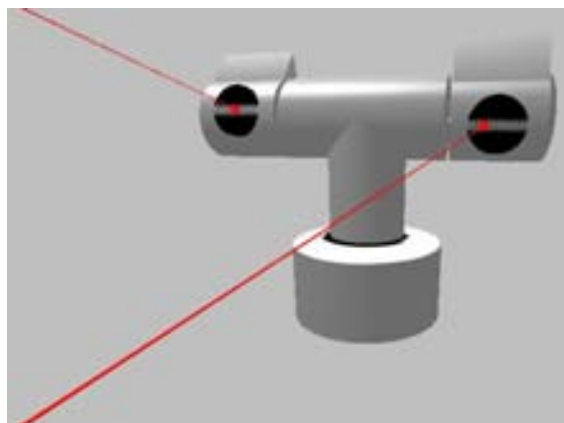


Figura 6.3: Simulador da Cabeça Estéreo Operando no Modo de Movimentação Independente

As figuras 6.3 e 6.4 mostram a simulação da cabeça estereó robótica em seus dois modos de operação. Na primeira figura, a cabeça estereó robótica está operando no modo de movimentação independente, cada câmera está focalizando um objeto distinto, enquanto que na segunda figura a cabeça opera no modo de movimentação dependente, existe uma grande correlação entre as imagens capturadas por cada câmera, já que ambas as câmeras focalizam um mesmo objeto, tornando possível a implementação do algoritmo da visão estereó. Esses experimentos iniciais

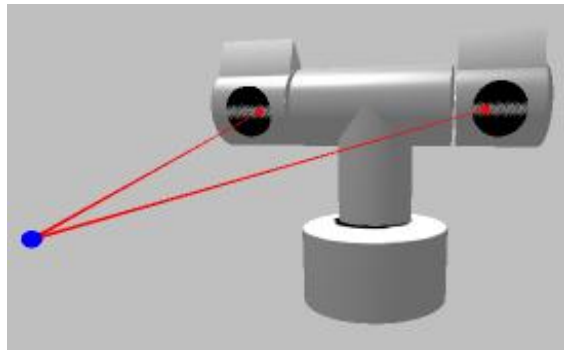


Figura 6.4: Cabeça Estéreo Operando no Modo de Movimentação Dependente

mostraram o correto funcionamento do modelo cinemático desenvolvido da parte física da cabeça.

A figura 6.5 demonstra os resultados obtidos através da simulação da cabeça operando no modo de movimentação dependente, demonstrando o 'movimento' de 'perseguição' a um objeto, confirmando a implementação do modelo cinemático da cabeça.

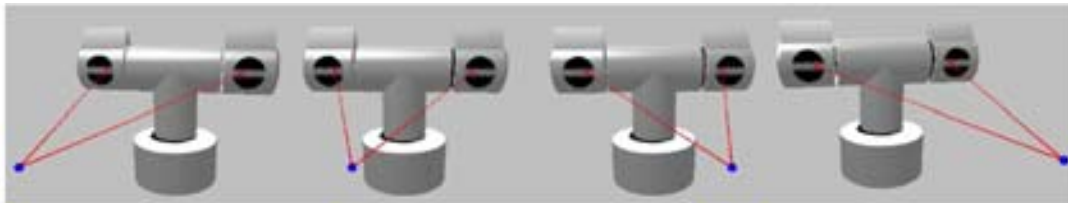


Figura 6.5: Simulador da Cabeça Estéreo perseguindo um objeto

6.2 Aquisição de Imagens

Para a captura das imagens pelas câmeras e aquisição pelo computador, algumas possibilidades de arranjo das câmeras nas placas de aquisição *Frame Grabber* foram consideradas e testadas. Levando em consideração que cada placa possui quatro canais de entrada de dados, poderia-se conectar as duas câmeras na mesma placa ou uma câmera em cada placa, sendo ambos os *Frame Grabbers* conectados à mesma placa mãe ou ainda poderia-se utilizar duas placas mães, cada uma suportando uma placa *Frame Grabbers* conectada a uma câmera.

A terceira configuração foi descartada ainda em fase de projeto, pois a cabeça disporá de apenas uma placa mãe. Os dois primeiros arranjos foram testados através de rotinas computacionais que adquiriam cinquenta quadros consecutivos das câmeras e cronometravam o tempo necessário, a partir do qual era calculado o tempo estimado para a aquisição de 1 (um) quadro. Os programas foram executados algumas vezes para cada um dos arranjos e os resultados são os seguintes:

Uma Câmera conectada a uma Placa de Aquisição (*Frame Grabber*)

tempo de captura de cada imagem:	69621	microssegundos	(14,36 fps)
tempo de captura de cada imagem:	67398	microssegundos	(14,84 fps)
tempo de captura de cada imagem:	67400	microssegundos	(14,84 fps)
tempo de captura de cada imagem:	67399	microssegundos	(14,84 fps)
tempo de captura de cada imagem:	68067	microssegundos	(14,69 fps)
tempo de captura de cada imagem:	68066	microssegundos	(14,69 fps)
tempo de captura de cada imagem:	66733	microssegundos	(14,98 fps)
tempo de captura de cada imagem:	67400	microssegundos	(14,83 fps)
tempo de captura de cada imagem:	66731	microssegundos	(14,98 fps)
tempo de captura de cada imagem:	67400	microssegundos	(14,83 fps)

Duas Câmeras conectadas a uma Placa de Aquisição (*Frame Grabber*)

tempo de captura de cada imagem:	89063	microsegundos	(11,22 fps)
tempo de captura de cada imagem:	88084	microsegundos	(11,35 fps)
tempo de captura de cada imagem:	94095	microsegundos	(10,63 fps)
tempo de captura de cada imagem:	94760	microsegundos	(10,55 fps)
tempo de captura de cada imagem:	92758	microsegundos	(10,78 fps)
tempo de captura de cada imagem:	94092	microsegundos	(10,63 fps)
tempo de captura de cada imagem:	88752	microsegundos	(11,27 fps)
tempo de captura de cada imagem:	88756	microsegundos	(11,27 fps)
tempo de captura de cada imagem:	87419	microsegundos	(11,44 fps)
tempo de captura de cada imagem:	93425	microsegundos	(10,70 fps)

Devido ao melhor desempenho do primeiro arranjo sobre o segundo, esse será o adotado. Ainda assim, fica evidente a necessidade de otimização na captura de imagens, que precisa atingir, pelo menos, a taxa de 20 fps (“quadros por segundo”).

Vale ressaltar que a taxa de processamento de quadros pode ser multiplicada simplesmente pela adoção do padrão de imagens em preto-e-branco, uma vez que esse padrão reduz a representação de uma imagem de três matrizes para somente uma. Tal padrão não foi testado por vislumbrarmos a capacidade de se conseguir os resultados esperados no padrão RGB, que preserva determinadas informações da imagem capturada no padrão preto-e-branco.

Também não foi testada a tecnologia *Wire Fire* de transmissão de dados devido à idéia somente ter sido cogitada após a aquisição das câmeras e da placa de aquisição de imagens. Fica a sugestão, entretanto, para futuras implementações ou remontagens do projeto.

6.3 Redução e Extração de Características de Imagens

Como teste para a estratégia de redução de imagens adotada, foi escrito inicialmente um programa que adquire da câmera um quadro por vez. O processo de redução gasta em média 244 microssegundos, como pode ser visto na Tabela 6.1.

Etapa	Multiresolução (μs)	Original (μs)
Multiresolução	244	-
Gaussiano	584	10480
Gradiente	1169	21020
Laplaciano	579	10506
Movimento	62	5355
Total	2638	47361

Tabela 6.1: Resultados obtidos

Então, a cada imagem captada aplica-se cada um dos filtros explicados nas sessões anteriores, excetuando o filtro de movimento, que só pode ser calculado a partir da segunda aquisição, pois opera uma subtração entre o quadro corrente e seu anterior.

A Figura 6.6 mostra uma imagem antes do processamento ser realizado. A tabela 6.1 exibe os resultados em μs conseguidos pelo programa para o teste com essa imagem. As Figuras 6.7-6.12 mostram as imagens resultantes do processo de redução e abstração dos dados.



Figura 6.6: Imagem Original

Tendo em vista que a taxa desejada é de 20fps e que o processo de aquisição do novo quadro será executado por um processo paralelo ao do processamento gráfico, sabe-se que o tempo disponível para o processamento gráfico adicionado ao tempo empregado na implementação da inteligência



Figura 6.7: Imagem em Multiresolução



Figura 6.8: Filtro Gaussiano

do robô é de 50ms. A tabela 6.1 nos demonstra que um tempo de 3ms é suficiente para a filtragem das imagens reduzidas, restando aos outros processamentos gráficos e de inteligência do robô um tempo de 47ms. Confrontando com o tempo necessário para a filtragem da imagem original, percebemos um ganho de cerca de 1800%, o que retoma a viabilidade da taxa de aquisição.

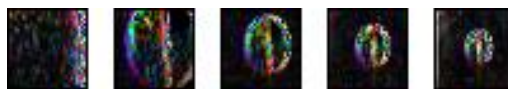


Figura 6.9: Filtro Gradiente X

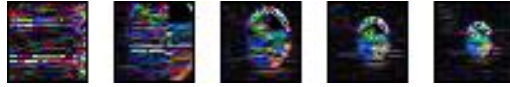


Figura 6.10: Filtro Gradiente Y



Figura 6.11: Filtro Laplaciano



Figura 6.12: Detecção de Movimento

Capítulo 7

Conclusão e Perspectivas

Neste trabalho, colaboramos para o projeto e implementação da parte física da cabeça estéreo do projeto Robosense, bem como elaboramos várias ferramentas para as fases iniciais do processamento de imagens fornecidas pelas câmeras da cabeça. Para isso, utilizamos técnicas de redução de dados visando aumentar a velocidade de processamento e técnicas de filtragem de imagens visando extração de características.

Mostramos que a idéia da utilização de um algoritmo baseado na fóvea humana simplificada permite que a cabeça estéreo do projeto RoboSense possa operar em tempo real.

A construção da imagem em multi-resolução e a extração de algumas de suas características já encontram-se implementadas, demonstrando ganhos substanciais e imprescindíveis no tempo de processamento das imagens. Tais resultados comprovam a viabilidade da idéia e são um forte argumento para a sua utilização no projeto.

A implementação da correlação cruzada entre as duas imagens adquiridas pelas câmeras da cabeça estéreo já encontra-se implementada por trabalho posterior[25], seguindo as idéias iniciais.

Os passos seguintes, com relação à parte computacional do projeto, são a melhoria da taxa de captura dos quadros pelas placas de aquisição, além da implementação dos comportamentos de mais alto nível da cabeça, ou seja, os algoritmos de visão propriamente ditos, levando em consideração o estudo de movimentos sacádicos, controle de atenção e reconhecimento de padrões e objetos.

A construção física do robô, bem como da cabeça, conforme mostrado pelas Figuras 7, 7 e 7.3 encontram-se bastante avançadas, restando apenas para a sua conclusão a implementação das placas de acionamento dos motores de todo o robô, responsáveis pelo movimento das rodas, dos elos dos braços e da cabeça. A finalização da construção do robô possibilitará brevemente o início

dos testes e a implementação, no simulador, do comportamento cinemático e dinâmico reais de sua cabeça

A adaptação do simulador cinemático já existente para simulação dinâmica dos movimentos da cabeça será bastante útil ao projeto, uma vez que essa nova funcionalidade certamente agregará conhecimento ao corpo técnico envolvido com o projeto. A adaptação facilitará os testes aos quais será submetida a cabeça.

Uma outra implementação prevista é a adaptação do simulador para o controle da cabeça via internet, tecnologia esta que está sendo desenvolvida em projetos paralelos.

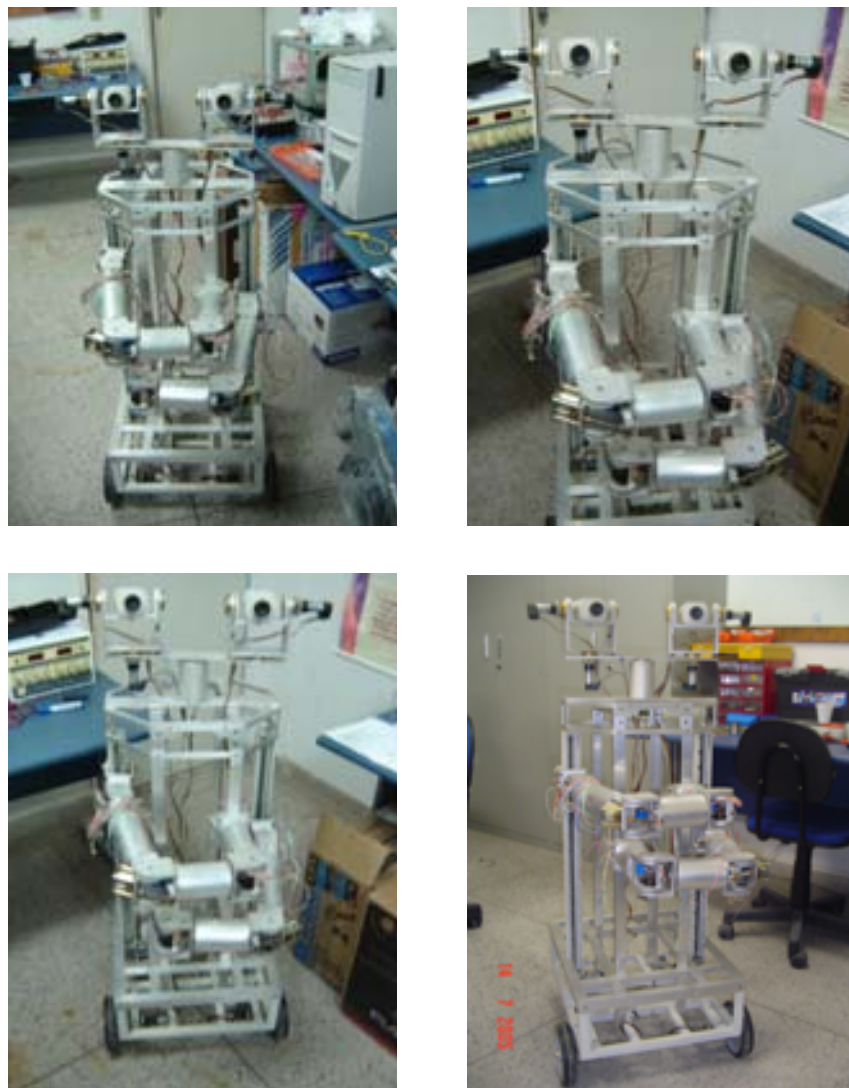


Figura 7.1: Fotos do Robosense atual

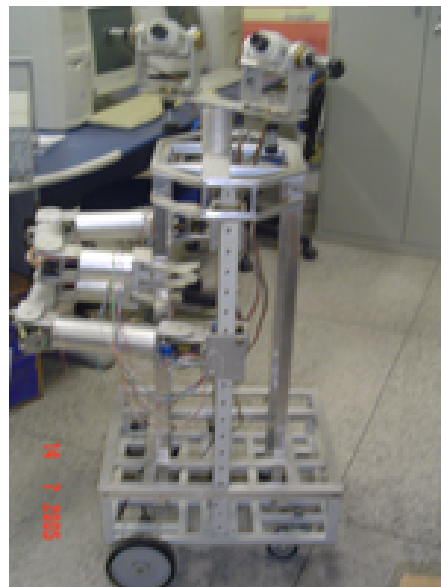
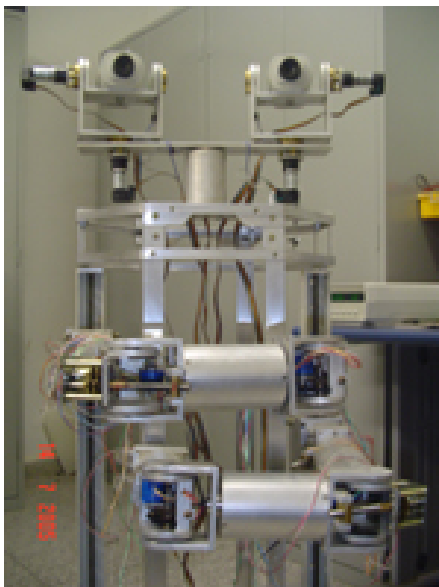


Figura 7.2: Corpo do RoboSense

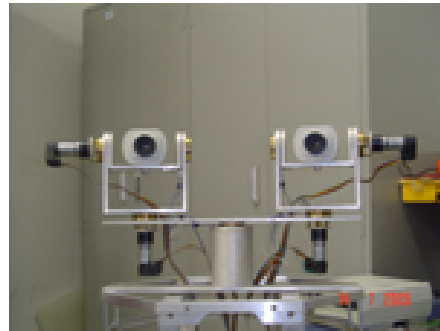
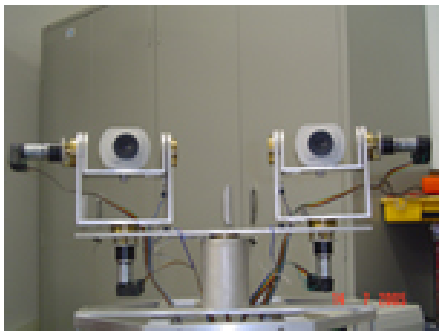


Figura 7.3: Detalhe da Cabeça Estéreo do RoboSense

Apêndice A

Projeto Físico da Cabeça

Este apêndice foi inserido no trabalho com a finalidade de registrar oficialmente e evitar a perda da memória do projeto físico dos elos da cabeça robótica do RoboSense.

O projeto foi desenvolvido pelo Eng. Anfraserai de Moraes.

Referências Bibliográficas

- [1] SDC Engenharia e Sistemas <http://www.sdc.com.br/prod/Imagem/Camera/camera.htm>
(Visitado dia 09 de maio de 2004).
- [2] Sensoray - Fabricante da Placa de Aquisição de Imagens
<http://www.sensoray.com/html/611data.htm> (Visitado dia 09 de maio de 2004).
- [3] Gonzalez, Woods. *Processamento Digital de Imagens*, Editora Edgard Blücher Ltda (2000).
- [4] Silveira, Bezerra, Bezerra, Segundo, Gonçalves. *Um Modelo para Redução e Abstração de Dados em uma Cabeça Estéreo* EnRI 2004 - Salvador BA.
- [5] Bolduc, M., Levine, Martin D. *A Foveated Retina for Robotic Vision Research in Computer and Robotic Vision*. In C. Archibald, P. Kwok, World Scientific Press, Singapore, February 1995. p. 93-115.
- [6] Jain, Anil K. *Fundamentals os Digital Image Processing*, Prentice Hall (1989). p. 348-353.
- [7] Andreas RUF. *The Coordination of Perception and Action: the projective approach* Tutorial em Slides para a European Conference on Computer Vision, (ECCV'00), Dublin, Irlanda, June 2000. <http://www.inrialpes.fr/movi/people/Ruf/rufECCV00b.pdf> (consultado em 22 de Outubro de 2005)
- [8] C. Garcia. *Fully Vision-Based Calibration of a Hand-Eye Robot* Publicado em *Autonomous Robots* - 6(1999), p.223-238. 1999 Kluwer Academic Publishers, Boston. <http://www.csd.uoc.gr/~cgarcia/PAPERS/AUTOROB-00.pdf> (consultado em 24 de Outubro de 2005)
- [9] Sheng-Wen Shih, Yi-Ping Hung, and Wei-Song Lin. *New Closed-Form Solution for Kinematic Parameter Identification of a Binocular Head Using Point*

- Measurements IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS - PART B: CYBERNETICS, VOL. 28, NO. 2, APRIL 1998
http://ippr.csie.ntu.edu.tw:8080/Publication/17_New(consultado em 10 de Outubro de 2005)
- [10] Rajeev Sharma e Narayan Srinivasa. A Framework for Active Vision-based Robot Control Using Neural Robotica (1998) volume 16, p. 309-327. Impresso no Reino Unido© 1998 Cambridge University Press networks.
<http://profusion.bu.edu/techlab/Docs/srinivasa98.pdf> (consultado em 08 de Outubro de 2005)
- [11] K.T. Gribbon, C.T. Johnston, and Bailey D.G. A Real-time FPGA Implementation of a Barrel Distortion Correction Algorithm with Bilinear Interpolation
<http://www.celoxica.com/techlib/files/CEL-W040226WP5-259.pdf> (consultado em 13 de Outubro de 2005)
- [12] Bailey D.G. A New Approach to Lens Distortion Correction, Proceedings Image and Vision Computing New Zealand 2002, pp 59-64 (2002).
- [13] Michael R. Bax Real-Time Lens Distortion Correction: 3D Video Graphics Cards Are Good for More than Games.
http://www.stanford.edu/group/iee/ecj/articles/bax_spr04.pdf (consultado em 01 de Novembro de 2005).
- [14] Keenan, Paul A. Method for Calibrating a SCARA Robot, United States Patent 4725965 (1988) <http://freepatentsonline.com/4725965.html> (Consultado em 17 de Setembro de 2005).
- [15] Young S., Scott P., Bandera C. Foveal Automatic Target Recognition Using a Multiresolution Neural Network - IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 7, NO. 8, Agosto 1998. <http://ece.uic.edu/~sankar1/sriram/papers/5.pdf> (consultado em 12 de setembro de 2005)
- [16] Otuyama, J. Visão Estéreo - <http://www.inf.ufsc.br/~visao/1998/otuyama/1.htm> (consultado em 13 de outubro de 2005).
- [17] Vale, A. Stereo Binocular - Apostila sobre o assunto.
<http://omni.isr.ist.utl.pt/~vale/projects/visao/visao.html> (consultado em 29 de setembro de 2005).

- [18] Iocchi, L. Calibration of Stereo Cameras for Mobile Robotsr
<http://www.dis.uniroma1.it/~iocchi/stereo/calib.html> (consultado em 29 de setembro de 2005).
- [19] B. K. Paul HORN. *Robot Vision*. MIT Press, 1986.
- [20] D. H. BALARD, C. M. BROWN. *Computer Vision*. Prentice-Hall. Englewood Cliffs, NJ, 1982.
- [21] D. MARR. *Vision, A Computational Investigation into the Human Representation and Processing of Visual Information*. MIT Press, 1982.
- [22] W. E. L. GRIMSON. *From Images to Surfaces: A Computational Study of the Human Early Visual System*. MIT Press. Cambridge, Mass. 1981.
- [23] Athanasios PAPOULIS. *Probability, Random Variables, and Stochastic Processes*. MacGRAW-HILL, 1991.
- [24] Emanuele Trucco e Alessandro Verry. *Introductory Techniques for 3D Computer Vision* MacGRAW-HILL, 1998.
- [25] L. M. Goncalves, J. P. Bezerra, C. G. Serrano Algoritmo Estéreo em Tempo Real para Módulo de Percepção do Projeto RoboSense. Anais do Workshop de Iniciação científica do SIBGRAPI 2005. Natal, RN, Brasil.
- [26] Gianna Rodrigues Araújo, Ricardo Wendell e L. M. Goncalves Eliminação de Distorção em Imagens. Anais do Workshop de Iniciação científica do SIBGRAPI 2005. Natal, RN, Brasil.
- [27] Devernay, Frédéric and Faugeras, Olivier. Straight lines have to be straight: Automatic calibration and removal of distortion from scenes of structured environments. *Machine Vision and Applications*, 1, 2001, p. 14-24.