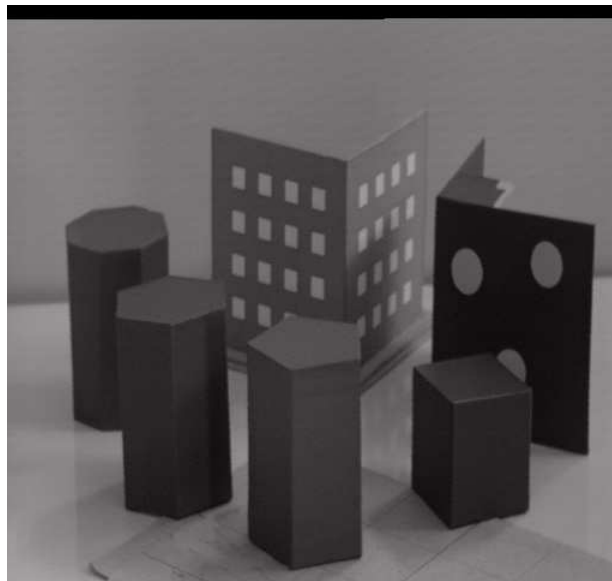


UnB – Universidade de Brasília
FT – Faculdade de Tecnologia
ENE – Departamento de Engenharia Elétrica

Sistema de Visão Estéreo para Formas Polimétricas



Projeto de Graduação de

Cristiano Jacques Miosso Rodrigues

Mirele de Almeida Mencari

Orientador

Dr. Adolfo Bauchspiess

Brasília, DF, Brasil
Março, 1999

Visão Computacional Estéreo Aplicada a Formas Poliédricas

Resumo

Uma das mais importantes aplicações do processamento digital de imagens na área de controle e automação é a guiagem de robôs. Trata-se de determinar, a partir das imagens captadas de um ambiente, as coordenadas de uma trajetória a ser seguida, de objetos a serem manipulados ou ainda de obstáculos presentes, para que se evitem colisões.

Para este fim, a utilização de uma única imagem de uma cena impõe sérias restrições. Na estrutura bidimensional de uma imagem não há informação suficiente para a reconstituição de objetos tridimensionais, ou seja, a noção de profundidade é perdida. Uma solução comumente utilizada é o emprego de sensores ativos, a infravermelho, “laser” ou ultra-som, que permitam recuperar a informação de profundidade e, em conjunto com as imagens mono ou isoladamente, reconstituir os ambientes tridimensionais.

Este trabalho aborda um dos temas atualmente mais pesquisados na área de visão computacional, e que constitui outra técnica viável e promissora para a navegação de robôs: a visão estereo. Por meio do processamento em conjunto das imagens de uma cena geradas por câmeras em posições distintas, denominadas imagens estereo, podem ser calculadas as três coordenadas espaciais dos objetos presentes partindo-se de relações geométricas simples.

Um dos problemas centrais deste intento é a determinação dos pontos homólogos nas imagens estereo. A utilização de mapas de correspondências densos, que permitam determinar os homólogos de cada “*pixel*”, exige um grande esforço computacional, dada a grande complexidade do processamento de imagens.

É proposto então um algoritmo para cálculo da correspondência entre pontos estratégicos nas imagens, levando à obtenção de mapas de profundidade esparsos. Numa fase de pré-processamento, é efetuada uma detecção de bordas das imagens estereo, o que reduz em cerca de 90% a quantidade de pontos utilizados nos cálculos seguintes. Este pré-processamento exige, contudo, que os objetos filmados possam ser bem representados por segmentos de reta. A correspondência é então edetuada a partir da retificação das imagens binárias obtidas.

O sistema foi implementado utilizando o MATLAB 5.0. A aquisição das imagens foi feita utilizando uma câmera monocromática CCD marca PULNIX, série TN-6. Foram construídos um padrão de pontos de referência, para calibração da câmera, e quatro prismas, para teste dos algoritmos desenvolvidos.

Para a validação do sistema implementado foi feita a reconstituição dos cinco sólidos a partir das correspondências manual e automática das imagens estereo. Os resultados obtidos nos dois casos foram equivalentes, mostrando-se satisfatórios diante dos modelos utilizados. Observou-se contudo uma grande sensibilidade a pequenos deslocamentos da câmera em relação às posições de calibração, comprovando a necessidade de uma boa precisão no posicionamento das câmeras.

“O primeiro passo é uma intuição - que surge como uma explosão - em seguida aparecem os problemas. Falham isto e aquilo, vão aparecendo os pequenos erros e dificuldades. Meses de ansiosa observação, estudo e trabalho são necessários antes que o sucesso ou fracasso seja alcançado. “

Thomas Edison

Agradecimentos

Sou grato a todos os que, direta ou indiretamente, auxiliaram-me durante estes últimos cinco anos de estudos, contribuindo de forma significativa para minha formação. Pelo suporte inestimável à realização desta pesquisa, não poderia faltar um registro de especial gratidão:

À minha mãe, Alaide, a quem devo todas as oportunidades de estudo que tive. Sua dedicação, coragem, inteligência e insuperável capacidade de trabalho me inspiram;

Aos meus irmãos Júnior, Alexandre e Simone, ao primo Filipe e à minha tia Marilda, sempre prestativos, que contribuíram com um agradável convívio diário e souberam perdoar todas as minhas falhas;

Ao Professor Dr. Adolfo Bauchspiess, pela dedicação e disponibilidade durante todo o trabalho e pelo conhecimento e experiência que me foram passados. A excelente orientação e as soluções inteligentes e práticas propostas durante as fases mais difíceis da pesquisa foram de grande auxílio;

Ao Professor Dr. Lúcio Martins da Silva, orientador de minha primeira pesquisa de Iniciação Científica, pela excelente orientação, pela confiança em mim depositada e, sobretudo, por ter me mostrado a importância e o prazer da pesquisa científica. Foi para mim uma honra trabalhar ao seu lado;

À pesquisadora Carla e ao técnico Antônio Marrocos, membros do GRACO, sempre dispostos a auxiliar na execução da etapa experimental da pesquisa;

E à colega Mirele de Almeida Mencari, pelo grande empenho e seriedade mostrados em mais de um ano de pesquisa. Sua competência foi imprescindível para a realização deste projeto.

Cristiano Jacques Miosso

Agradecimentos

A realização de um Projeto Final de curso requer um grande esforço e uma considerável dedicação para obtenção das metas propostas. Sua realização se torna mais agradável e fácil com a ajuda de boa parte das pessoas que nós cercam, aos quais sou grata. Mas é indispensável o registro em especial de gratidão:

À família pelo apoio e estímulo durante sua realização e pela compreensão pelos momentos de ausência.

Ao técnico do GRAGO, Antônio Marrocos, pela atenção demonstrada a realizar cada uma de nossas solicitações.

À pesquisadora Carla pelo auxílio, apoio e interesse demonstrado em relação ao nosso trabalho

Em especial ao Professor Doutor Adolfo Bauchspiess pela paciência e dedicação ao nos orientar na realização deste projeto.

E principalmente ao meu colega de trabalho, Cristiano Jacques Miosso Rodrigues, pela amizade, respeito e dedicação conjunta durante o desenvolvimento do projeto.

Mirele de Almeida Mencari

Capítulo 1

Introdução

Visão é capacidade de detectar, reconstituir e reconhecer objetos. O processo de visão começa pela captação dos feixes de luz produzidos ou refletidos por estes objetos por um elemento fotossensível, formando uma imagem. O último passo é o processamento da imagem captada, que permite reconstituir e/ou reconhecer o objeto existente na mesma. Reconstituir um objeto significa determinar as suas dimensões e sua localização no espaço. Reconhecer um objeto significa qualificá-lo, ou seja, dizer qual é o objeto, dentre um conjunto de possibilidades, que se encontra na imagem.

Para os seres humanos a visão é aparentemente um processo simples que não causa esforço já que é realizado de forma involuntária, porém é de fato um processo altamente sofisticado. Estima-se que a visão seja responsável por um terço da atividade do cérebro humano [4].

Cada passo do fenômeno da visão requer um ou mais processamentos, alguns dos quais interagem de forma complexa. Com frequência, a conclusão de determinadas etapas requer informações obtidas a partir de outras que, contudo, não podem ser realizadas sem informações parciais das primeiras.

Visão Computacional é também uma das áreas da inteligência artificial que tenta recriar o processo de visão biológico. Devido à complexidade, para cada um dos processamentos existentes no fenômeno da visão existem linhas específicas de pesquisa.

Algumas das aplicações da visão computacional são:

1. Controle dimensional;
2. Inspeção de embalagens, rótulos, etiquetas e tampas;
3. Reconhecimento de caracteres e leitura de código de barras;
4. Verificação da presença e disposição de componentes em montagens;
5. Identificação de rebarbas;
6. Verificação do nível em envasamento (nível de líquido numa garrafa, por exemplo);
7. Classificação e seleção por tamanho, formato e cor;
8. Inspeção de cápsulas;
9. Inspeção de seringas descartáveis;
10. Contagem de partículas;
11. Inspeção e calibração de displays e indicadores;
12. Inspeção de placas eletrônicas;
13. Inspeção de superfícies; e
14. Visão para robôs e manipuladores.

Os sensores utilizados em visão computacional podem ser passivos ou ativos. Sensor ativo é aquele emite um sinal - onda sonora (sonar), raios luminosos (laser), onda eletromagnética (radar) – e capta o mesmo sinal após este ter sido refletido no objeto existente na cena. Sensor passivo é aquele que capta os sinais luminosos refletidos nos objetos da cena, sinais estes gerados por uma fonte externa ao sistema (sol, lâmpada, etc.). Os olhos dos seres vivos são sensores passivos.

Visão estéreo é o processo pelo qual os seres vivos percebem a profundidade de uma determinada cena. Um ponto de uma cena é projetado em diferentes posições em em cada um dos olhos sendo esta diferença denominada *disparidade*. O uso da disparidade e de relações trigonométricas permitem calcular a profundidade do ponto. A visão computacional estéreo utiliza este mesmo princípio para toda cena.

Existem várias técnicas de visão computacional para o cálculo da profundidade. A vantagem da visão estéreo é ser uma técnica que utiliza sensores passivos, e portanto mais barata (um sistema de visão ativo utilizaria muito mais sensores – ver o exemplo do apêndice A). Uma desvantagem em relação aos sistemas ativos é que o sensor passivo demanda um processamento mais elaborado (maior número de dados e cálculos).

Em relação a sistemas que utilizem apenas uma imagem a vantagem do sistema estéreo é permitir o cálculo da profundidade.

O objetivo deste projeto foi a construção de um sistema de visão estéreo que fornecesse as coordenadas de pontos chaves na imagem. Estes pontos chaves seriam as bordas de objetos que permitissem a reconstrução do mesmo (dimensões e posição no espaço), criando um modelo do espaço em que se encontra a câmera. Este modelo poderia ser então utilizado para a navegação de um robô móvel ou de um manipulador. Outras utilidades seriam as descritas nos itens 4, 5 e 7 das aplicações da visão computacional.

O capítulo 2 descreve brevemente o sistema implementado sem entrar em detalhamento matemático. O objetivo do capítulo é mostrar a importância de determinados temas desenvolvido nos capítulos teóricos.

O capítulo 3 aborda o modelamento matemático de uma câmera.

O capítulo 4 descreve o método de obtenção dos parâmetros da câmera que são indispensáveis no cálculo da profundidade.

O capítulo 5 trata da transformação de retificação das imagens estéreo, que permite o alinhamento dos pontos homólogos e a consequente redução da complexidade computacional do cálculo dos mapas de correspondência.

O capítulo 6 mostra algumas técnicas de processamento de imagem. Neste capítulo é descrito o funcionamento da detecção de bordas, pré-processamento utilizado no sistema implementado.

O capítulo 7 descreve a metodologia utilizada durante a implementação do sistema bem como suas limitações.

O capítulo 8 mostra os resultados experimentais obtidos.

Capítulo 2

Descrição do Sistema Estéreo

A seguir apresenta-se uma breve descrição do sistema desenvolvido neste trabalho, preocupando-se apenas com conceitos e métodos, sem detalhamento matemático. A figura 2.1 ilustra o diagrama de blocos do sistema que tem como objetivo calcular as coordenadas em 3D de uma cena através do uso de imagens em 2D, capturadas da cena 3D por câmeras CCD monocromáticas. Sendo portanto, um sistema de visão passivo.

2.1 Câmera CCD

A câmera CCD pode ser modelada pela câmera escura de orifício, mostrada na figura 2.2. Os raios luminosos, vindos de uma fonte de luz (Sol, lâmpada, etc.) refletem no objeto, atravessam o orifício da câmera e são amparados pela parede oposta ao orifício. Estes raios luminosos formam uma imagem invertida do objeto neste parede que é denominada de plano de imagem.

Os raios luminosos viajam em linha reta de modo que o raio luminoso que sai do ponto A e passa pelo orifício O é amparado na parte inferior do plano de imagem. Este fato explica a razão da inversão da imagem em relação ao objeto original.

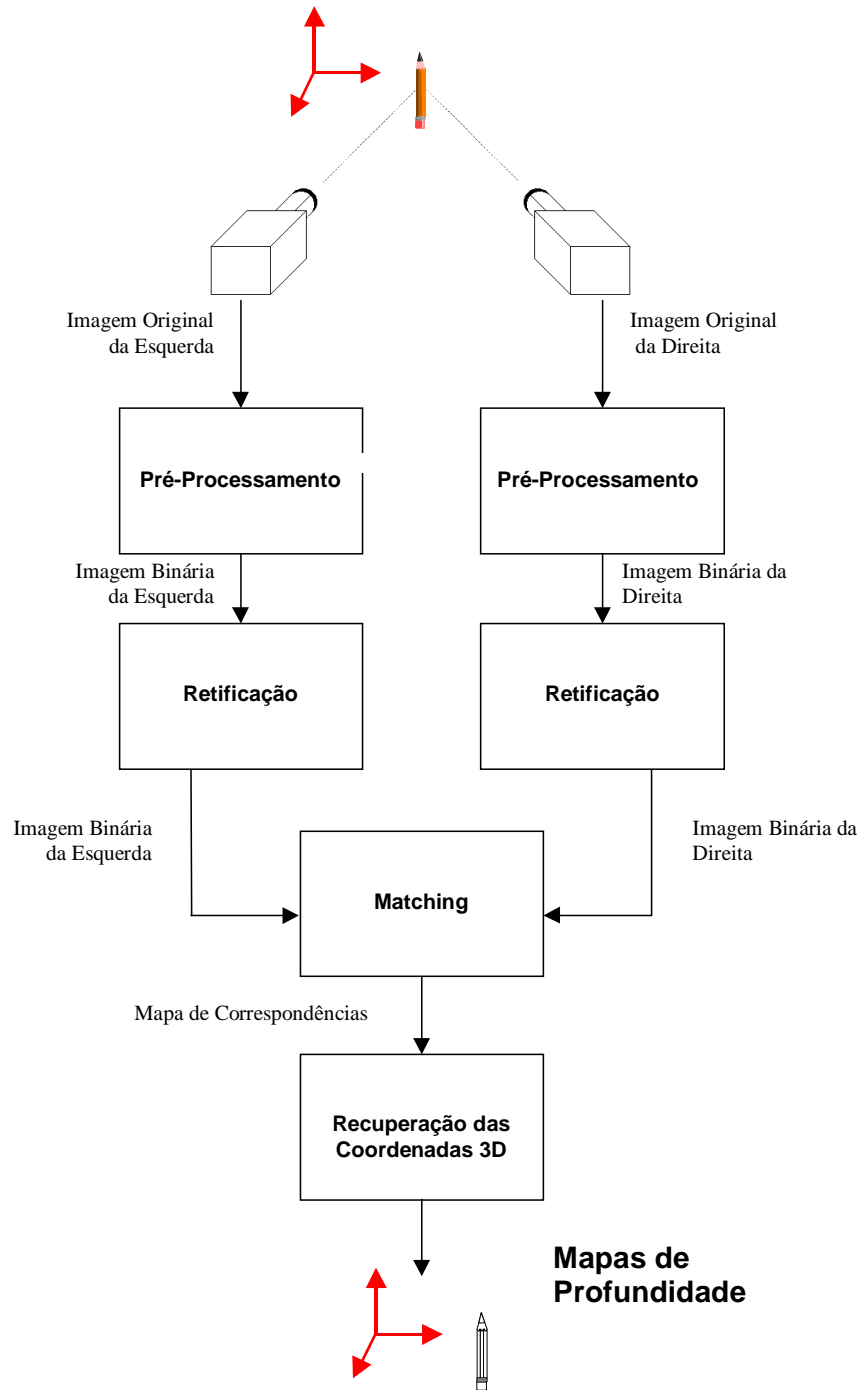


Figura 2.1 : Diagrama em blocos do sistema de visão estéreo desenvolvido neste trabalho.

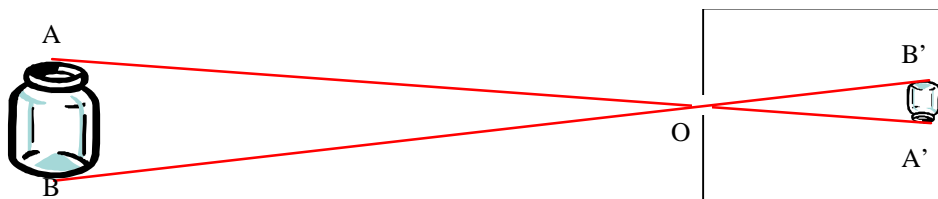


Figura 2.2 : Câmera escura de orifício

Considere a imagem $A'B'$ de um lápis formada no plano de imagem, figura 2.3. Não é conhecido a posição e as dimensões do objeto que a criou. A tentativa de determiná-los mostra que existem infinitas soluções para o problema. O ponto A' da imagem pode ter sido formada por qualquer ponto existente na reta $\overline{A'O}$. Do mesmo modo o ponto B' pode ter sido formada por qualquer ponto da reta $\overline{B'O}$. Possíveis soluções para o caso anterior são os lápis A_1B_1 , A_2B_2 e A_3B_3 .

Uma imagem isolada não pode dar sozinha informações sobre a profundidade do objeto. Como se deseja obter as dimensões 3D do objeto, torna-se necessário utilizar duas câmeras, sistema de visão estéreo. A partir de informações sobre um mesmo ponto nas imagens das duas câmeras é possível calcular a profundidade da cena. A explicação detalhada será dada em capítulos posteriores.

2. 2 Pré-Processamento

Uma imagem digitalizada consiste de uma quantidade significativa de informação, que do ponto de vista computacional representa um grande volume de dados a ser processado. Processamento de imagens consome um tempo considerável, que para certas aplicações é inaceitável.

Uma maneira de se reduzir o tempo de processamento é realizar um pré-processamento que reduza quantidade de dados sem perda de informação. No caso do sistema da figura 2.1 o processamento corresponde ao cálculo das coordenadas 3D de objetos existentes na imagem.

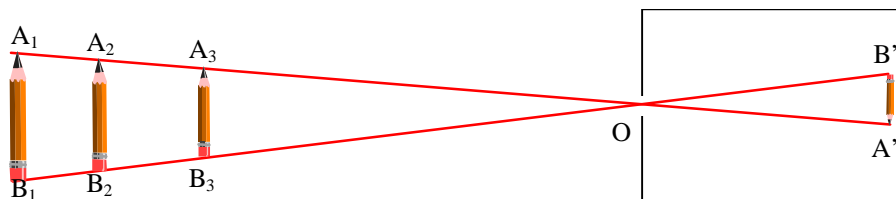


Figura 2.3 : Dada uma imagem $A'B'$ no plano de imagens existe uma infinidade de possibilidades de posições e dimensões do objeto que a formou.

Para se obter as coordenadas 3D de objetos na imagem não é necessário conhecer todas coordenadas de todos os pontos que o constitui. O objeto pode ser reconstituído conhecendo-se as coordenadas das suas extremidades (como os vértices de um prisma).

No sistema do diagrama em blocos, o pré-processamento significa isolar os pontos das extremidades do objetos. Isto é realizado com a utilização de técnicas de detecção de bordas. A detecção de bordas consiste em localizar na imagem a fronteira entre duas regiões que possuem características de intensidade distintas. Como cada “face” do objeto na imagem apresenta uma tonalidade distinta, logo a detecção das fronteiras entre duas regiões distintas significa a detecção das extremidades do objeto existente na imagem. A imagem resultante deste pré-processamento é uma imagem binária, onde o 1 lógico representa a existência de borda e o 0 ausência.

Esta imagem binária pode ser usada de duas formas nos processamentos posteriores: como matéria prima do processamento ou como indicador de onde que o processamento deve ser feito na imagem original. No sistema estéreo deste texto a imagem binária é usada como matéria prima do processamento.

2.3 Retificação

As duas câmeras de vídeo podem, teoricamente, ocupar qualquer posição no espaço. Porém há uma configuração que reduz consideravelmente os cálculos, chamada configuração canônica [5]. Nesta configuração a retina das duas câmeras são colocadas sobre um mesmo plano, espaçadas com uma distância d , e paralelamente uma em relação a outra, figura 2.4.

Uma imagem digitalizada é formada por *pixels* que podem ser definidos por um sistema de coordenadas semelhante ao do plano cartesiano ou como elementos de uma matriz, sendo este último utilizado no desenvolvimento do sistema de visão estéreo em questão. Cada ponto do sistema ou elemento da matriz contém a informação de cor.

A configuração canônica apresenta a característica de que o *pixel* da imagem da direita e o da esquerda que representam o mesmo ponto da cena 3D estarem em posições de linha idênticos em ambas as matrizes, figura 2.5.

Apesar da configuração canônica facilitar os cálculos, dificilmente ela é fisicamente construída devido a limitações físicas do ambiente em que o sistema estéreo será montado. Normalmente as câmeras se encontram sobre a reta e espaçadas por uma

distância d , porém apresentam uma certa inclinação relativa, de modo que haja interseção das áreas de visão das duas câmeras, configuração divergente, figura 2.6.

Embora não seja fisicamente construída, a configuração canônica pode ser obtida através de transformações matemáticas aplicadas nas imagens obtidas pela configuração divergente. Este tipo de transformação é denominado de retificação da imagem, e é aplicado sobre cada ponto da imagem.

No sistema estéreo do diagrama em bloco a retificação é feita sobre a imagem binarizada, mas especificamente nos *pixels* com valor lógico 1. Esta escolha é explicada pela redução de cerca 90% dos pontos a serem retificados em relação a imagem original (esta porcentagem se refere a imagens com um objeto simples – cubo, prisma de base pentagonal, etc.).

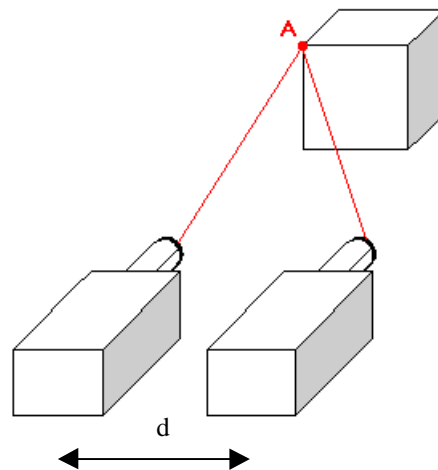


Figura 2.4 : Configuração canônica das câmeras para um sistema de visão estéreo

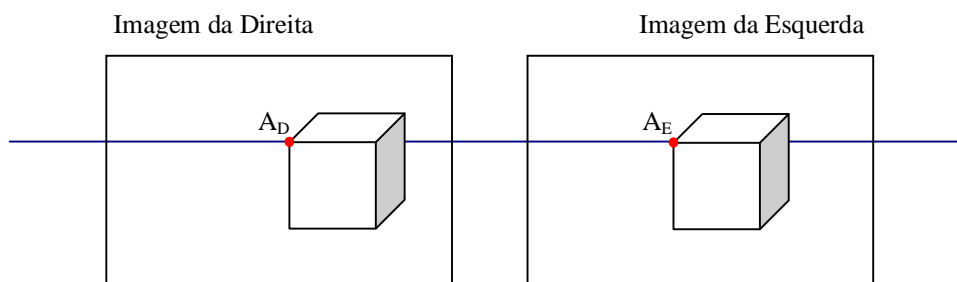


Figura 2.5 : Par de imagens obtido de um sistema de visão estéreo

2.4 Correspondência de Imagens

Para se calcular as coordenadas 3D é necessário encontrar o mesmo ponto da cena 3D nas duas imagens. Daí a importância da retificação que nos assegura que um ponto da cena 3D encontra-se em ambas as imagens na mesma linha, reduzindo a busca bidimensional de pontos correspondentes a uma busca unidimensional [3].

Correspondência de imagens é o processo de busca do *pixel* correspondente numa das imagens ao *pixel* existente na outra.

A saída do bloco *matching* no diagrama em blocos da figura 2.1 é uma matriz de três colunas em que cada linha representa as informações de um ponto sendo a primeira coluna o índice da linha na matriz imagem, a segunda o índice da coluna do *pixel* na imagem da direita e a terceira o índice da coluna do *pixel* na imagem da esquerda. Esta matriz é denominada de mapa de correspondências [5].

2.5 Reconstituição dos Pontos em 3D

A partir do mapa de correspondências os pontos em 3D são calculados. Para efetuar este cálculo é necessário levantar as características da câmera que fornecem uma matriz de transformação. Esta matriz calcula as coordenadas do *pixel* correspondente a um ponto 3D da cena na imagem.

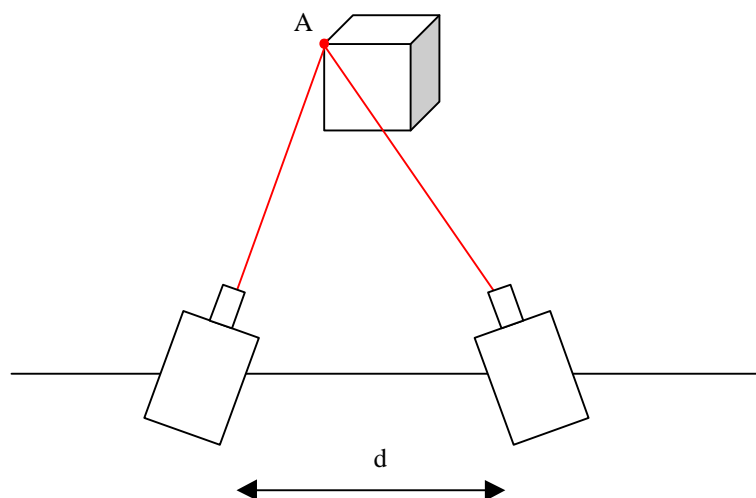


Figura 2.6 : Configuração divergente das câmeras de um sistema estéreo

Este processo de levantamento das características da câmera só precisa ser feito uma vez. A matriz de transformação, denominada de matriz de calibração, depende da posição da câmera no espaço e qualquer alteração nesta posição fará necessário ajustes na matriz [5]

A matriz de calibração junto com a coordenada de um *pixel* permite a construção de um sistema com quatro variáveis (coordenadas x , y e z e escalada da imagem) e três equações, que possui infinitas soluções. Esta é a interpretação matemática para a utilização de duas ou mais câmeras. Num sistema estéreo de duas câmeras, as duas matrizes de calibração e as coordenadas de *pixels* correspondentes nas duas imagens permite a construção de um sistema de quatro variáveis (coordenadas x , y e z e escala da imagem) e seis equações cuja a solução são as coordenadas 3D que formaram os *pixels*.

Capítulo 3

Modelamento Matemático de Câmeras Digitais

3.1 Modelamento geométrico – a câmera escura de orifício

O processo de formação de imagens pelos mais diversos tipos de câmeras pode ser modelado a partir de um dispositivo ideal bastante simples, a câmera escura de orifício. Trata-se de uma caixa de paredes opacas, uma das quais apresenta um orifício de dimensões ínfimas. A figura 3.1 esquematiza este dispositivo; o orifício da câmera é representado pelo ponto C.

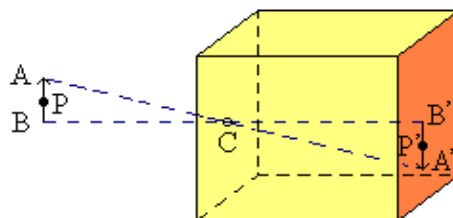


Figura 3.1 : A câmera escura de orifício

A seguir será descrito o processo de formação de imagens neste dispositivo. Para tanto, considere-se o objeto arbitrário AB representado. Como as paredes da câmera são opacas e o orifício C é um ponto, um único raio de luz proveniente de um ponto P de AB penetra a caixa; a intersecção deste raio com a parede oposta à que contém o orifício é a imagem de P formada, designada por P'. Deve-se observar que, nesta situação, a imagem de qualquer objeto será semelhante a ele, porém invertida.

Um modelo geométrico ainda mais simples pode ser empregado, como mostrado na figura 3.2. A câmera passa a ser completamente descrita por três objetos matemáticos:

- o ponto C, designado o **foco** da câmera;
- o plano R em que são formadas as imagens, denominado **plano de retina**;
- o plano F que contém C e é paralelo a R, denominado **plano focal**.

Observe-se que a imagem M' de um ponto arbitrário M é, neste modelo, a intersecção do plano R com a reta que passa por M e C.

Outro conceito importante relacionado ao modelo apresentado na figura 3.2 é o de eixo óptico. Trata-se da reta perpendicular ao plano R e que contém o ponto C. O ponto de intersecção entre o eixo óptico e o plano de retina será denominado **ponto principal**, representado por c.

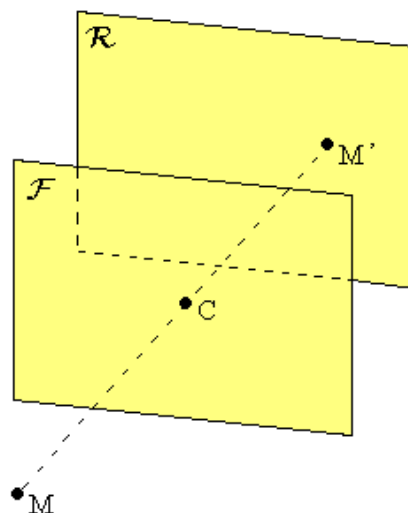


Figura 3.2 : Modelo geométrico simplificado da câmera escura de orifício (foco C, plano focal F e plano de retina R).

3.2 Determinação das coordenadas da imagem de um objeto pontual

Serão deduzidas a seguir duas expressões úteis referentes à formação de imagens numa câmara escura de orifício. Sejam v e u as coordenadas da imagem de um ponto P , referidas a um sistema ortogonal de centro c ; os eixos deste sistema são supostos paralelos a dois eixos \mathbf{v} e \mathbf{h} perpendiculares ao eixo óptico e que se interceptam em C . A figura 3.3 descreve a situação.

Deseja-se relacionar o vetor \mathbf{D} , que define a posição do ponto P em relação ao sistema de coordenadas $(C, \mathbf{a}, \mathbf{h}, \mathbf{v})$ padrão da câmara, às coordenadas u, v da imagem P' formada. Considere-se, para tanto, a figura 3.4, correspondente ao plano $h=0$ na situação descrita na figura 3.3. O vetor \mathbf{D}_{av} mostrado corresponde à projeção de \mathbf{D} no plano considerado.

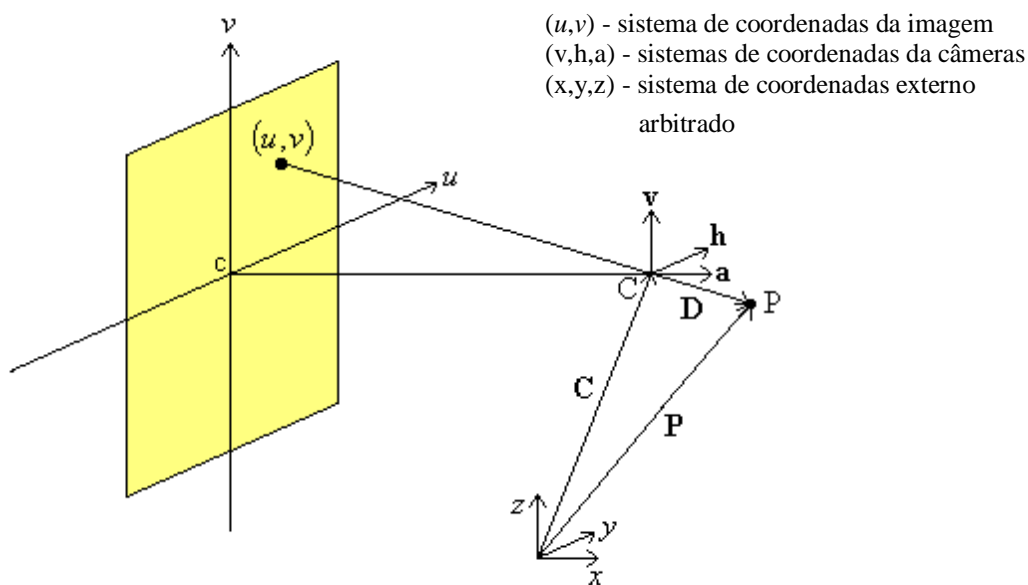


Figura 3.3 : A formação de imagem numa câmara escura de orifício (Projeção Perspectiva).

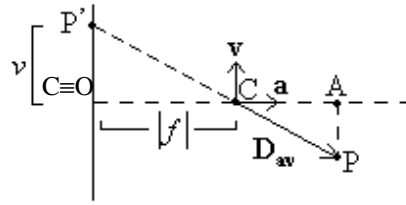


Figura 3.4 : Visão bidimensional da projeção perspectiva ilustrada em 3.3.

Pela semelhança dos triângulos COP' e CAP, tem-se:

$$\frac{|\mathbf{D}_{av} \cdot \mathbf{v}|}{|\mathbf{D}_{av} \cdot \mathbf{a}|} = \frac{|v|}{|f|}, \quad (3.1)$$

onde $|f|$ é a distância do ponto C ao plano R, isto é, a distância focal da lente da câmera.

Sendo f a componente na direção \mathbf{a} da posição do ponto c em relação ao foco C, tem-se ainda:

$$\frac{\mathbf{D}_{av} \cdot \mathbf{v}}{\mathbf{D}_{av} \cdot \mathbf{a}} = \frac{v}{f} \quad (3.2)$$

Finalmente, como $\mathbf{D}_{av} \cdot \mathbf{v} = \mathbf{D} \cdot \mathbf{v}$ e $\mathbf{D}_{av} \cdot \mathbf{a} = \mathbf{D} \cdot \mathbf{a}$, obtém-se:

$$\frac{\mathbf{D} \cdot \mathbf{v}}{\mathbf{D} \cdot \mathbf{a}} = \frac{v}{f} \quad (3.3)$$

De forma análoga, obtém-se a relação correspondente envolvendo a coordenada u :

$$\frac{\mathbf{D} \cdot \mathbf{h}}{\mathbf{D} \cdot \mathbf{a}} = \frac{u}{f} \quad (3.4)$$

No caso específico das câmeras digitais, como a utilizada neste projeto (câmera CCD), as imagens são formadas em células retangulares que compõem o plano de

retina, sendo que um único nível de cinza é registrado em cada uma das células. Tais imagens correspondem, portanto, a matrizes cujos elementos, denominados *pixels* (*picture elements*), são associados ao nível de cinza proveniente de cada célula.

Sejam j e i os índices do elemento de uma imagem digital em que é registrado o nível de cinza da célula situada em (u, v) . Sejam ainda j_0 e i_0 os índices associados ao ponto principal c . Tem-se então:

$$u = (j - j_0) \cdot \delta u \quad (3.5)$$

$$v = (i - i_0) \cdot \delta v, \quad (3.6)$$

onde δu e δv são os comprimentos das células do plano de retina nas direções u e v , respectivamente¹.

Assim, as duas relações antes estabelecidas resultam, no caso das câmeras digitais:

$$\frac{\mathbf{D} \cdot \mathbf{v}}{\mathbf{D} \cdot \mathbf{a}} = \frac{(i - i_0) \cdot \delta v}{f} \quad (3.7)$$

$$\frac{\mathbf{D} \cdot \mathbf{h}}{\mathbf{D} \cdot \mathbf{a}} = \frac{(j - j_0) \cdot \delta u}{f} \quad (3.8)$$

3.3 A matriz de transformação perspectiva

Conforme já mencionado, um dos objetivos do modelamento matemático apresentado é determinar a relação entre as coordenadas x , y , z de um ponto referidas a um sistema arbitrário e a posição do “pixel” correspondente na imagem gerada. Esta relação será útil aos processos de retificação das imagens estéreo (capítulo 5) e de reconstituição da cena filmada (capítulo 8).

Nesta seção, será estabelecida esta relação em coordenadas homogêneas, com o auxílio da denominada **matriz de transformação perspectiva**.

¹ Os valores de δu e de δv são determinados pelas dimensões do *pixel* em cada direção.

Para simplificar a notação a ser utilizada nos cálculos seguintes, considere-se inicialmente o teorema apresentado a seguir [5].

Teorema:

Definindo

$$\mathbf{V} = \frac{\mathbf{v}}{\delta v} + \frac{i_0}{f} \cdot \mathbf{a} \quad (3.9)$$

e

$$\mathbf{H} = \frac{\mathbf{h}}{\delta u} + \frac{j_0}{f} \cdot \mathbf{a}, \quad (3.10)$$

tem-se:

$$\frac{\mathbf{D} \cdot \mathbf{V}}{\mathbf{D} \cdot \mathbf{a}} = \frac{i}{f} \quad \text{e} \quad \frac{\mathbf{D} \cdot \mathbf{H}}{\mathbf{D} \cdot \mathbf{a}} = \frac{j}{f} \quad (3.11)$$

Demonstração:

Para o valor de \mathbf{V} definido:

$$\frac{\mathbf{D} \cdot \mathbf{V}}{\mathbf{D} \cdot \mathbf{a}} = \frac{\mathbf{D} \cdot \left(\frac{\mathbf{v}}{\delta v} + \frac{i_0}{f} \cdot \mathbf{a} \right)}{\mathbf{D} \cdot \mathbf{a}}$$

$$\frac{\mathbf{D} \cdot \mathbf{V}}{\mathbf{D} \cdot \mathbf{a}} = \frac{\mathbf{D} \cdot \mathbf{v}}{\delta v (\mathbf{D} \cdot \mathbf{a})} + \frac{i_0}{f} \cdot \frac{\mathbf{D} \cdot \mathbf{a}}{\mathbf{D} \cdot \mathbf{a}} \quad (3.12)$$

Da equação 3.7, segue:

$$\begin{aligned} \frac{\mathbf{D} \cdot \mathbf{V}}{\mathbf{D} \cdot \mathbf{a}} &= \frac{i - i_0}{f} \cdot \frac{\delta v}{\delta v} + \frac{i_0}{f} \\ \frac{\mathbf{D} \cdot \mathbf{V}}{\mathbf{D} \cdot \mathbf{a}} &= \frac{i}{f} \end{aligned} \quad (3.13)$$

A demonstração da igualdade $\frac{\mathbf{D} \cdot \mathbf{H}}{\mathbf{D} \cdot \mathbf{a}} = \frac{j}{f}$ é análoga.

Definindo $S = \mathbf{D} \cdot \mathbf{a}$, $U = f \cdot \mathbf{D} \cdot \mathbf{H}$, $V = f \cdot \mathbf{D} \cdot \mathbf{V}$, tem-se:

$$\begin{cases} S = D_x a_x + D_y a_y + D_z a_z \\ V = f \cdot (D_x V_x + D_y V_y + D_z V_z) \\ U = f \cdot (D_x H_x + D_y H_y + D_z H_z) \end{cases} \quad (3.14)$$

ou, em notação matricial:

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} fH_x & fH_y & fH_z \\ fV_x & fV_y & fV_z \\ a_x & a_y & a_z \end{pmatrix} \cdot \begin{pmatrix} D_x \\ D_y \\ D_z \end{pmatrix} \quad (3.15),$$

com:

$$j = \frac{U}{S} \quad (3.16)$$

$$i = \frac{V}{S} \quad (3.17)$$

Sendo \mathbf{P} o vetor posição do ponto (x, y, z) em relação ao centro do sistema de coordenadas externo arbitrado e \mathbf{C} o vetor posição do foco da câmera em relação àquele mesmo centro, conforme mostrado na figura 3.3, tem-se:

$$\begin{aligned} \mathbf{P} &= \mathbf{C} + \mathbf{D} \\ \therefore \mathbf{D} &= \mathbf{P} - \mathbf{C} \end{aligned} \quad (3.18)$$

Substituindo na equação 3.15 a 3.18, obtém-se:

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} fH_x & fH_y & fH_z \\ fV_x & fV_y & fV_z \\ a_x & a_y & a_z \end{pmatrix} \cdot \begin{pmatrix} P_x - C_x \\ P_y - C_y \\ P_z - C_z \end{pmatrix} \quad (3.19)$$

$$\therefore \begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} fH_x & fH_y & fH_z & 0 \\ fV_x & fV_y & fV_z & 0 \\ a_x & a_y & a_z & 0 \end{pmatrix} \cdot \begin{pmatrix} P_x - C_x \\ P_y - C_y \\ P_z - C_z \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} fH_x & fH_y & fH_z & -C'_x \\ fV_x & fV_y & fV_z & -C'_y \\ a_x & a_y & a_z & -C'_z \end{pmatrix} \cdot \begin{pmatrix} P_x \\ P_y \\ P_z \\ 1 \end{pmatrix} \quad (3.20),$$

com:

$$\begin{pmatrix} C'_x \\ C'_y \\ C'_z \end{pmatrix} = \begin{pmatrix} fH_x & fH_y & fH_z \\ fV_x & fV_y & fV_z \\ a_x & a_y & a_z \end{pmatrix} \cdot \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} \quad (3.21)$$

Observe-se que a equação 3.20 estabelece a relação desejada entre as coordenadas de um ponto (referidas a um sistema arbitrário) e as coordenadas homogêneas $(U \ V \ S)^T$ do *pixel* correspondente na imagem gerada pela câmera. Desta forma, a matriz de transformação perspectiva é dada por:

$$\tilde{\mathbf{P}} = \begin{pmatrix} fH_x & fH_y & fH_z & -C'_x \\ fV_x & fV_y & fV_z & -C'_y \\ a_x & a_y & a_z & -C'_z \end{pmatrix} \quad (3.22)$$

Designando por $\tilde{\mathbf{m}}$ as coordenadas homogêneas $(U \ V \ S)^T$ e por $\tilde{\mathbf{w}}$ $(P_x \ P_y \ P_z \ 1)^T$, pode-se rescrever 3.20 como:

$$\tilde{\mathbf{m}} = \tilde{\mathbf{P}} \cdot \tilde{\mathbf{w}} \quad (3.23),$$

que é a equação da projeção perspectiva.

3.4 Parâmetros intrínsecos e extrínsecos

Substituindo em 3.22 os valores de \mathbf{H} e de \mathbf{V} , chega-se a:

$$\tilde{\mathbf{P}} = \begin{pmatrix} \frac{f}{\delta u} \cdot h_x + j_0 \cdot a_x & \frac{f}{\delta u} \cdot h_y + j_0 \cdot a_y & \frac{f}{\delta u} \cdot h_z + j_0 \cdot a_z & -\frac{f}{\delta u} \cdot \mathbf{C} \cdot \mathbf{h} - j_0 \cdot \mathbf{C} \cdot \mathbf{a} \\ \frac{f}{\delta v} \cdot v_x + i_0 \cdot a_x & \frac{f}{\delta v} \cdot v_y + i_0 \cdot a_y & \frac{f}{\delta v} \cdot v_z + i_0 \cdot a_z & -\frac{f}{\delta v} \cdot \mathbf{C} \cdot \mathbf{v} - i_0 \cdot \mathbf{C} \cdot \mathbf{a} \\ a_x & a_y & a_z & -\mathbf{C} \cdot \mathbf{a} \end{pmatrix} \quad (3.24)$$

$$\therefore \tilde{\mathbf{P}} = \begin{pmatrix} \frac{f}{\delta u} & 0 & j_0 \\ 0 & \frac{f}{\delta v} & i_0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} h_x & h_y & h_z & -\mathbf{C} \cdot \mathbf{h} \\ v_x & v_y & v_z & -\mathbf{C} \cdot \mathbf{v} \\ a_x & a_y & a_z & -\mathbf{C} \cdot \mathbf{a} \end{pmatrix} \quad (3.25)$$

Observe-se, nesta equação, que $\tilde{\mathbf{P}}$ foi decomposta no produto de duas matrizes; em uma delas, os elementos dependem somente das características internas da câmera utilizada, e não de sua posição e orientação em relação ao sistema de coordenadas externo arbitrado. Estes elementos, $\frac{f}{\delta u}$, $\frac{f}{\delta v}$, j_0 e i_0 , são os denominados **parâmetros intrínsecos da câmera**.

A outra matriz, por sua vez, é formada por elementos que definem a orientação e a posição da câmera com respeito ao sistema de coordenadas externo arbitrado. Estes elementos são denominados **parâmetros extrínsecos da câmera**.

A equação 3.26 resume estas designações.

$$\tilde{\mathbf{P}} = \underbrace{\begin{pmatrix} \frac{f}{\delta u} & 0 & j_0 \\ 0 & \frac{f}{\delta v} & i_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Parâmetros intrínsecos}} \cdot \underbrace{\begin{pmatrix} \mathbf{h}^T & -\mathbf{C} \cdot \mathbf{h} \\ \mathbf{v}^T & -\mathbf{C} \cdot \mathbf{v} \\ \mathbf{a}^T & -\mathbf{C} \cdot \mathbf{a} \end{pmatrix}}_{\text{Parâmetros extrínsecos}} \quad (3.26)$$

Capítulo 4

A calibração da câmera CCD

Foi mencionado no capítulo 3 que os processos de retificação de imagens estéreo e de reconstituição de uma cena a partir de imagens geradas por câmeras digitais requerem a determinação das matrizes de transformação perspectiva associadas a estas câmeras. Esta determinação não pode ser efetuada de forma direta a partir da equação 3.26, pelas seguintes razões:

- 1) Não se conhece de antemão a distância focal $|f|$ necessária para a determinação dos parâmetros intrínsecos (esta distância depende da lente utilizada podendo ser ajustada dentro de uma faixa de valores);
- 2) São desconhecidas as coordenadas do ponto principal, i_0 e j_0 (elas não coincidem necessariamente com o centro das imagens fornecidas pela câmera);
- 3) Mesmo que fossem fornecidos os parâmetros intrínsecos, os vetores \mathbf{h} , \mathbf{v} , \mathbf{a} , \mathbf{C} , que definem a posição e a orientação da câmera em relação ao sistema de coordenadas externo arbitrado, não poderiam ser medidos diretamente com a precisão necessária, de forma que a matriz de parâmetros extrínsecos permaneceria indeterminada.

Assim sendo, a matriz de transformação perspectiva é determinada por um procedimento experimental específico. O processo de **calibração** das câmeras, como é denominado, tem na verdade dois objetivos:

- A determinação da matriz de transformação perspectiva $\tilde{\mathbf{P}}$;
- O cálculo dos parâmetros intrínsecos e extrínsecos a partir da matriz $\tilde{\mathbf{P}}$ obtida.

Cabe ressaltar que, devido ao processo utilizado em sua medição, a matriz de transformação perspectiva é também designada **matriz de calibração**.

Neste capítulo, é descrito o processo de calibração utilizado nesta pesquisa, apresentado em [5]. A seção 4.1 trata do procedimento para medição da matriz de calibração, ao passo que 4.2 é dedicada ao cálculo dos parâmetros intrínsecos e extrínsecos.

4.1 Medição da matriz de transformação perspectiva

A primeira etapa do processo de calibração consiste em obter com a câmera em estudo uma imagem de diversos pontos cujas coordenadas em relação ao sistema ortogonal arbitrado são conhecidas, sendo em seguida determinados os *pixels* correspondentes àqueles pontos. Deve-se dispor, portanto, de um **padrão de calibração**, que nada mais é que uma superfície não-plana (esta condição é explicada adiante) em que são marcados em posições bem conhecidas os pontos a serem utilizados na medição. A figura 4.1 mostra o padrão de calibração utilizado nesta pesquisa; os pontos utilizados na determinação de $\tilde{\mathbf{P}}$ foram os centróides dos 32 quadrados claros; em alguns casos utilizam-se seus vértices, o que resulta em 132 pontos disponíveis para a calibração.

No capítulo 8, são fornecidas as coordenadas dos 32 centróides de referência adotados. São ainda fornecidas as matrizes de calibração obtidas pelo procedimento apresentado neste capítulo.

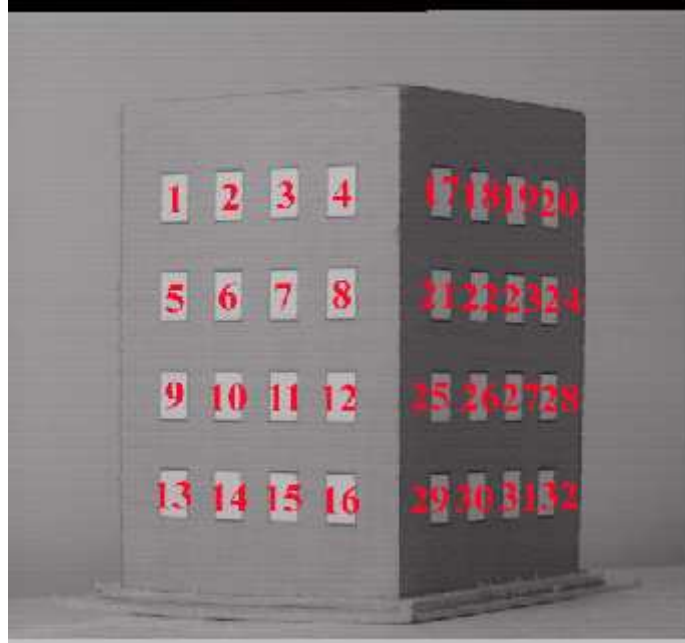


Figura 4.1 : O padrão de calibração empregado.

Efetuada a correspondência entre os 32 *pixels* centrais e as coordenadas dos pontos do padrão, é possível calcular a matriz de transformação perspectiva. Para tanto, considerem-se as equações 3.16, 3.17 e 3.20; aplicando-as para cada ponto (x_m, y_m, z_m) do padrão de calibração, obtêm-se as igualdades:

$$\begin{cases} j_m = \frac{\tilde{\mathbf{P}}_{1,1} \cdot x_m + \tilde{\mathbf{P}}_{1,2} \cdot y_m + \tilde{\mathbf{P}}_{1,3} \cdot z_m + \tilde{\mathbf{P}}_{1,4}}{\tilde{\mathbf{P}}_{3,1} \cdot x_m + \tilde{\mathbf{P}}_{3,2} \cdot y_m + \tilde{\mathbf{P}}_{3,3} \cdot z_m + \tilde{\mathbf{P}}_{3,4}} \\ i_m = \frac{\tilde{\mathbf{P}}_{2,1} \cdot x_m + \tilde{\mathbf{P}}_{2,2} \cdot y_m + \tilde{\mathbf{P}}_{2,3} \cdot z_m + \tilde{\mathbf{P}}_{2,4}}{\tilde{\mathbf{P}}_{3,1} \cdot x_m + \tilde{\mathbf{P}}_{3,2} \cdot y_m + \tilde{\mathbf{P}}_{3,3} \cdot z_m + \tilde{\mathbf{P}}_{3,4}} \end{cases} \quad (4.1)$$

Observe-se que o número de incógnitas deste sistema é 12 (os elementos de $\tilde{\mathbf{P}}$), ao passo que o número de equações obtidas é $2N$, sendo N o número de pontos de coordenadas conhecidas no padrão de calibração. Observe-se também que se trata de um sistema homogêneo; conseqüentemente, há infinitas soluções. Isto já era esperado, já que se a matriz $\tilde{\mathbf{P}}$ for multiplicada por uma constante não-nula qualquer, as razões expressas em 3.16 e 3.17 não se alteram; por este processo, $\tilde{\mathbf{P}}$ pode ser definida, portanto, a menos de uma constante multiplicativa.

Para que se obtenha uma solução única para o sistema, o termo 3,4 da matriz de calibração será adotado igual a 1. Isto significa dizer que será obtida uma matriz $\tilde{\mathbf{P}}'$ igual à matriz $\tilde{\mathbf{P}}$ definida no capítulo 3 dividida por $(-\mathbf{a} \cdot \mathbf{C})$, o valor de $\tilde{\mathbf{P}}_{3,4}$ em 3.24. Assim:

$$\tilde{\mathbf{P}}' = -\frac{1}{\mathbf{a} \cdot \mathbf{C}} \cdot \tilde{\mathbf{P}} \quad (4.2)$$

O sistema a ser resolvido passa a ter 11 incógnitas e $2N$ equações do tipo:

$$\begin{cases} j_m = \frac{\tilde{\mathbf{P}}'_{1,1} \cdot x_m + \tilde{\mathbf{P}}'_{1,2} \cdot y_m + \tilde{\mathbf{P}}'_{1,3} \cdot z_m + \tilde{\mathbf{P}}'_{1,4}}{\tilde{\mathbf{P}}'_{3,1} \cdot x_m + \tilde{\mathbf{P}}'_{3,2} \cdot y_m + \tilde{\mathbf{P}}'_{3,3} \cdot z_m + 1} \\ i_m = \frac{\tilde{\mathbf{P}}'_{2,1} \cdot x_m + \tilde{\mathbf{P}}'_{2,2} \cdot y_m + \tilde{\mathbf{P}}'_{2,3} \cdot z_m + \tilde{\mathbf{P}}'_{2,4}}{\tilde{\mathbf{P}}'_{3,1} \cdot x_m + \tilde{\mathbf{P}}'_{3,2} \cdot y_m + \tilde{\mathbf{P}}'_{3,3} \cdot z_m + 1} \end{cases} \quad (4.3)$$

Para que possa ser solucionado o sistema, deve-se ter portanto:

$$2N \geq 11 \quad (4.4)$$

$$N \geq 6 \quad (4.5)$$

o que significa que o padrão de calibração deve dispor de no mínimo 6 pontos de coordenadas conhecidas.

Rescrevendo as equações do sistema 4.3:

$$\begin{cases} j_m = \tilde{\mathbf{P}}'_{1,1} \cdot x_m + \tilde{\mathbf{P}}'_{1,2} \cdot y_m + \tilde{\mathbf{P}}'_{1,3} \cdot z_m + \tilde{\mathbf{P}}'_{1,4} + \tilde{\mathbf{P}}'_{3,1} \cdot (-j_m x_m) + \tilde{\mathbf{P}}'_{3,2} \cdot (-j_m y_m) + \tilde{\mathbf{P}}'_{3,3} \cdot (-j_m z_m) \\ i_m = \tilde{\mathbf{P}}'_{2,1} \cdot x_m + \tilde{\mathbf{P}}'_{2,2} \cdot y_m + \tilde{\mathbf{P}}'_{2,3} \cdot z_m + \tilde{\mathbf{P}}'_{2,4} + \tilde{\mathbf{P}}'_{3,1} \cdot (-i_m x_m) + \tilde{\mathbf{P}}'_{3,2} \cdot (-i_m y_m) + \tilde{\mathbf{P}}'_{3,3} \cdot (-i_m z_m) \end{cases} \quad (4.6)$$

o que se reduz a:

$$\mathbf{AU} = \mathbf{B} \quad (4.7)$$

com:

$$\mathbf{A} = \begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -j_1 x_1 & -j_1 y_1 & -j_1 z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -i_1 x_1 & -i_1 y_1 & -i_1 z_1 \\ & & & & & & & \vdots & & & \\ x_N & y_N & z_N & 1 & 0 & 0 & 0 & 0 & -j_N x_N & -j_N y_N & -j_N z_N \\ 0 & 0 & 0 & 0 & x_N & y_N & z_N & 1 & -i_N x_N & -i_N y_N & -i_N z_N \end{pmatrix} \quad (4.8)$$

$$\mathbf{U} = \left(\tilde{\mathbf{P}}'_{1,1} \quad \tilde{\mathbf{P}}'_{1,2} \quad \tilde{\mathbf{P}}'_{1,3} \quad \tilde{\mathbf{P}}'_{1,4} \quad \tilde{\mathbf{P}}'_{2,1} \quad \tilde{\mathbf{P}}'_{2,2} \quad \tilde{\mathbf{P}}'_{2,3} \quad \tilde{\mathbf{P}}'_{2,4} \quad \tilde{\mathbf{P}}'_{3,1} \quad \tilde{\mathbf{P}}'_{3,2} \quad \tilde{\mathbf{P}}'_{3,3} \right)^T \quad (4.9)$$

$$\mathbf{B} = \left(j_1 \quad i_1 \quad j_2 \quad i_2 \quad \cdots \quad j_N \quad i_N \right)^T \quad (4.10)$$

Como vimos, trata-se necessariamente de um sistema sobredeterminado (com um número de equações superior ao de incógnitas). Neste caso, deve-se determinar uma solução ótima, que permita minimizar o erro quadrático $E = (\mathbf{AU} - \mathbf{B}) \cdot (\mathbf{AU} - \mathbf{B})^T$. Tal solução é dada por [5]:

$$\mathbf{U} = \left(\mathbf{A}^T \cdot \mathbf{A} \right)^{-1} \cdot \mathbf{A}^T \cdot \mathbf{B} \quad (4.11)$$

4.2 Determinação dos parâmetros intrínsecos e extrínsecos

A partir dos elementos medidos da matriz de calibração, é possível determinar os parâmetros intrínsecos e extrínsecos da câmera. Um algoritmo foi desenvolvido com esta finalidade por Ganapathy e é apresentado em [5].

A seguir, são apresentadas as etapas deste algoritmo. Observe-se que ele é baseado nas equações 3.24 e 4.2.

Algoritmo de Ganapathy

Notação: $\tilde{\mathbf{P}}'_i = (\tilde{\mathbf{P}}'_{i,1} \quad \tilde{\mathbf{P}}'_{i,2} \quad \tilde{\mathbf{P}}'_{i,3})$.

Determinação dos parâmetros da câmera a partir da matriz de calibração $\tilde{\mathbf{P}}'$.

- 1) Determina-se $(\mathbf{a} \cdot \mathbf{C})^2$ a partir de $\tilde{\mathbf{P}}'_3$:

$$(\mathbf{a} \cdot \mathbf{C})^2 = \frac{1}{\tilde{\mathbf{P}}'_3 \cdot (\tilde{\mathbf{P}}'_3)^T} \quad (4.12)$$

- 2) Determina-se j_0 a partir de $\tilde{\mathbf{P}}'_1$ e $\tilde{\mathbf{P}}'_3$:

$$j_0 = \frac{(\mathbf{a} \cdot \mathbf{C})^2}{\tilde{\mathbf{P}}'_1 \cdot (\tilde{\mathbf{P}}'_3)^T} \quad (4.13)$$

- 3) Determina-se i_0 a partir de $\tilde{\mathbf{P}}'_2$ e $\tilde{\mathbf{P}}'_3$:

$$i_0 = \frac{(\mathbf{a} \cdot \mathbf{C})^2}{\tilde{\mathbf{P}}'_2 \cdot (\tilde{\mathbf{P}}'_3)^T} \quad (4.14)$$

- 4) Determinam-se a_x , a_y , a_z a partir de $\tilde{\mathbf{P}}'_{3,1}$, $\tilde{\mathbf{P}}'_{3,2}$, $\tilde{\mathbf{P}}'_{3,3}$ (a suposição de que $\mathbf{a} \cdot \mathbf{C} < 0$ é explicada no item 7):

$$\begin{aligned} a_x &= -\tilde{\mathbf{P}}'_{3,1} \cdot (\mathbf{a} \cdot \mathbf{C}) \\ a_y &= -\tilde{\mathbf{P}}'_{3,2} \cdot (\mathbf{a} \cdot \mathbf{C}) \\ a_z &= -\tilde{\mathbf{P}}'_{3,3} \cdot (\mathbf{a} \cdot \mathbf{C}) \end{aligned} \quad (4.15)$$

- 5) Determina-se $\left| \frac{f}{\delta u} \right|$ a partir de $\tilde{\mathbf{P}}'_1$:

$$\left| \frac{f}{\delta u} \right| = \sqrt{\tilde{\mathbf{P}}'_1 \cdot (\tilde{\mathbf{P}}'_1)^T \cdot (\mathbf{a} \cdot \mathbf{C})^2 - j_0^2} \quad (4.16)$$

- 6) Determina-se $\left| \frac{f}{\delta v} \right|$ a partir de $\tilde{\mathbf{P}}'_2$:

$$\left| \frac{f}{\delta v} \right| = \sqrt{\tilde{\mathbf{P}}'_2 \cdot (\tilde{\mathbf{P}}'_2)^T \cdot (\mathbf{a} \cdot \mathbf{C})^2 - i_0^2} \quad (4.17)$$

- 7) Adotam-se $\mathbf{a} \cdot \mathbf{C} < 0$ e $\frac{f}{\delta u} > 0$ (o que equivale a arbitrar as direções de \mathbf{a} e de \mathbf{u}). Aceita a convenção, supõe-se $\frac{f}{\delta v} > 0$ e calculam-se $h_x, h_y, h_z, v_x, v_y, v_z$ a partir de $\tilde{\mathbf{P}}'_{1,1}, \tilde{\mathbf{P}}'_{1,2}, \tilde{\mathbf{P}}'_{1,3}, \tilde{\mathbf{P}}'_{2,1}, \tilde{\mathbf{P}}'_{2,2}, \tilde{\mathbf{P}}'_{2,3}$:

$$\left. \begin{aligned} h_x &= -\frac{\tilde{\mathbf{P}}'_{1,1} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_x}{\left(\frac{f}{\delta u}\right)} \\ h_y &= -\frac{\tilde{\mathbf{P}}'_{1,2} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_y}{\left(\frac{f}{\delta u}\right)} \\ h_z &= -\frac{\tilde{\mathbf{P}}'_{1,3} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_z}{\left(\frac{f}{\delta u}\right)} \end{aligned} \right| \begin{aligned} h_x &= -\frac{\tilde{\mathbf{P}}'_{1,1} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_x}{\left(\frac{f}{\delta u}\right)} \\ h_y &= -\frac{\tilde{\mathbf{P}}'_{1,2} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_y}{\left(\frac{f}{\delta u}\right)} \\ h_z &= -\frac{\tilde{\mathbf{P}}'_{1,3} \cdot (\mathbf{a} \cdot \mathbf{C}) + j_0 \cdot a_z}{\left(\frac{f}{\delta u}\right)} \end{aligned} \quad (4.18)$$

Em seguida, efetua-se um segundo cálculo de \mathbf{v} partindo-se do produto vetorial de \mathbf{a} por \mathbf{h} :

$$\mathbf{v}_2 = \mathbf{a} \wedge \mathbf{h}$$

Determina-se então o ângulo entre os valores experimentais \mathbf{v} e \mathbf{v}_2 . Se o mesmo for mais próximo de 0 do que de π , a suposição de que $\frac{f}{\delta v} > 0$ é correta. Caso contrário, tem-se $\frac{f}{\delta v} < 0$ e deve-se substituir \mathbf{v} por \mathbf{v}_2 .

- 8) Calculam-se C_x, C_y, C_z a partir de $\mathbf{a} \cdot \mathbf{C}, \tilde{\mathbf{P}}'_{1,4}$ e $\tilde{\mathbf{P}}'_{2,4}$:

$$\begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} = -\frac{\mathbf{a} \cdot \mathbf{C}}{f} \cdot \begin{pmatrix} \mathbf{a}^T \\ \mathbf{h}^T \\ \mathbf{v}^T \end{pmatrix}^{-1} \cdot \begin{pmatrix} -f \\ (j_0 - \tilde{\mathbf{P}}'_{1,4}) \cdot \delta u \\ (i_0 - \tilde{\mathbf{P}}'_{2,4}) \cdot \delta v \end{pmatrix} \quad (4.19)$$

Observe-se que a distância focal f pode ser determinada a partir das razões $f/\delta u$ e $f/\delta v$ desde que sejam conhecidas as dimensões do *pixel*. Estes valores, diretamente relacionados às taxas de amostragem horizontal e vertical da câmera, são em geral fornecidos pelo fabricante da câmera. Para a câmera Pulnix utilizada nesta pesquisa, o tamanho do pixel é dado por $\delta u = 8,6 \mu\text{m}$ e $\delta v = 8,3 \mu\text{m}$.

Capítulo 5

Retificação de Imagens Estéreo

5.1 O problema da correspondência entre imagens estéreo

Será mostrado no capítulo 7 que, conhecidas as coordenadas i e j dos *pixels* associados a um mesmo ponto nas imagens fornecidas por duas ou mais câmeras calibradas, podem ser determinadas de imediato as três coordenadas espaciais deste objeto. A reconstituição completa da cena filmada é portanto efetuada determinando-se, para cada *pixel* de uma imagem, aqueles que nas demais sejam representantes da mesma região da cena, designados seus pontos **homólogos** [10].

A determinação da correspondência ponto a ponto das imagens estéreo é, juntamente com a calibração das câmeras utilizadas (abordada no capítulo 4), o problema central da visão computacional estéreo [10]. Trata-se também da etapa de maior esforço computacional em todo o processo, e que constitui o principal obstáculo à adoção desta técnica em aplicações em tempo real que exijam a utilização de mapas de profundidade densos [7]. Conforme será explicado no capítulo 7, optou-se neste trabalho pelo uso de mapas de profundidade esparsos; as limitações e as vantagens desta abordagem serão explicadas adiante.

Com o objetivo de reduzir a complexidade do problema, é normalmente efetuada uma transformação das imagens, de forma que se possa garantir que pontos correspondentes estejam localizados sempre na mesma linha. Esta transformação, que permite que uma busca a princípio bidimensional de pontos correspondentes seja reduzida a uma busca unidimensional, é denominada **retificação**.

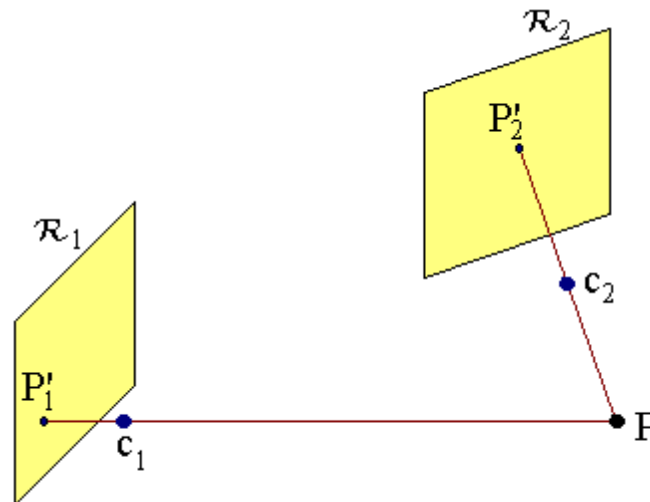


Figura 5.1 : Correspondência entre pontos de imagens estéreo geradas por câmeras em configuração arbitrária.

Para uma visão geral do processo, considere-se inicialmente a figura 5.1, que exemplifica a formação de imagens de um objeto pontual por duas câmeras em posições arbitrárias. As câmeras são caracterizadas pelos centros ópticos c_1 e c_2 e pelos planos de retina R_1 e R_2 .

Nesta figura, exemplifica-se o caso geral de correspondência entre imagens estéreo: a relação entre as coordenadas dos *pixels* correspondentes P'_1 e P'_2 é função da posição do objeto P . Desta forma, não se pode determinar a linha ou a coluna de P'_2 a partir somente da posição de P'_1 .

Já a figura 5.2 mostra o caso em que as duas câmeras apresentam o mesmo plano focal; além disso, a linha que une os pontos principais dos planos de retina são perpendiculares aos dois eixos ópticos e paralela a um dos eixos da imagem. Esta é a denominada configuração canônica [5]; neste caso, embora P'_2 não possa ser determinado somente a partir de P'_1 , sabe-se de antemão que ele se encontra ao longo de uma linha paralela a um dos eixos da imagem (no exemplo, o eixo horizontal). Esta linha, destacada em azul no plano R_2 na figura 5.2 e formada pelos possíveis correspondentes do ponto P'_1 , é a denominada linha epipolar.

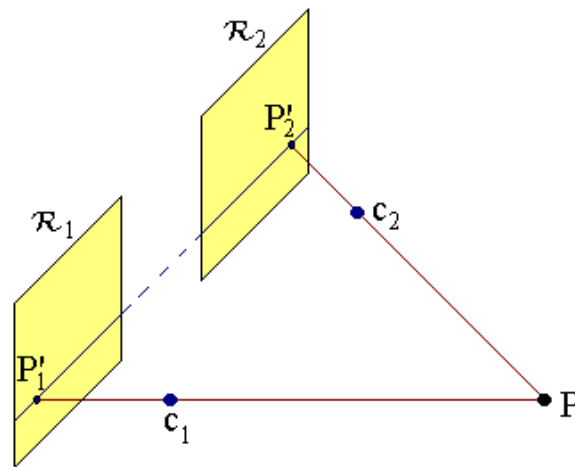


Figura 5.2 : Correspondência entre pontos de imagens estereo geradas por câmeras em configuração canônica.

No caso da figura 5.2, observe-se que a linha epipolar de P_1' não só é horizontal como também passa por P_1' . Em outras palavras, os pontos P_1' e P_2' têm a mesma coordenada j . Nesta situação, reduz-se o conjunto de pontos que possivelmente são homólogos de P_1' àqueles que, na imagem 2, situam-se na mesma linha horizontal que esse ponto.

Durante a retificação, o que se deseja é calcular a partir das imagens fornecidas por um sistema em configuração arbitrária aquelas que seriam fornecidas por câmeras na configuração canônica, de forma a garantir que pontos correspondentes em imagens distintas passem a se situar na mesma linha. Este processo se baseia no fato de que, dispondo-se da imagem fornecida por uma câmera, pode-se calcular a imagem da mesma cena que seria fornecida por uma outra câmera qualquer, desde que mantido o centro focal da configuração original. Na seção 5.7, será demonstrado como esta nova imagem pode ser calculada a partir de uma transformação linear e das matrizes de calibração da câmera original e da câmera hipotética que retifica a imagem. Já em 5.3, serão apresentadas as condições que esta segunda matriz de calibração deve satisfazer para que os pontos correspondentes nas imagens transformadas estejam sempre na mesma linha.

5.2 Transformação do plano de retina

Dada a imagem de uma cena fornecida por uma câmera digital calibrada, pode-se determinar aquela que seria obtida modificando-se a orientação e outros parâmetros de interesse desta câmera (tais como a distância focal e a resolução ao longo dos eixos vertical e horizontal), mas preservando seu centro óptico. Este processo, designado transformação do plano de retina, é utilizado para que se obtenham as imagens estéreo que seriam fornecidas por câmeras em configuração canônica, a partir daquelas disponíveis, obtidas com um sistema em configuração arbitrária.

A figura 5.3 ilustra como esta transformação é utilizada, durante a retificação. Os dois planos de retina originais, \mathcal{R}_1 e \mathcal{R}_2 , são reprojetaados num mesmo plano, garantindo que as novas linhas epipolares sejam paralelas a um dos eixos das imagens.

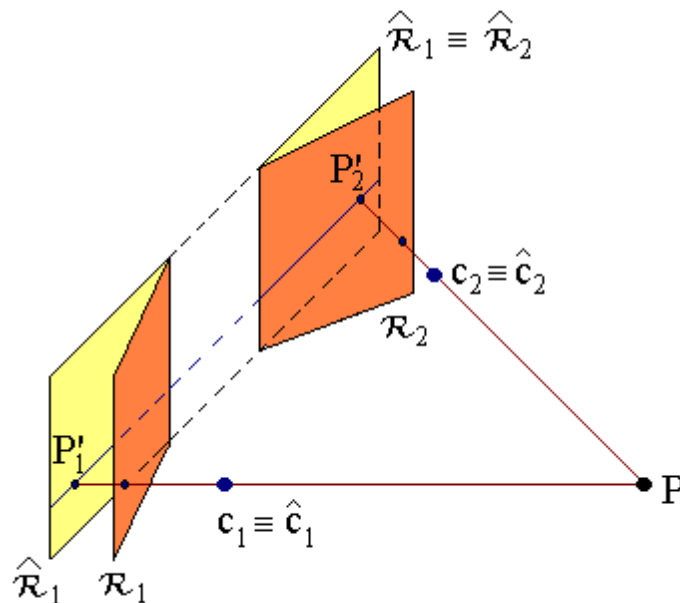


Figura 5.3 : Transformação dos planos de retina de duas imagens estéreo, durante o processo de retificação.

Para o cálculo da imagem \mathbf{I}_n obtida pela transformação do plano de retina, considerem-se as matrizes de calibração $\tilde{\mathbf{P}}_o = (\mathbf{P}_o | \mathbf{p}_o)$ e $\tilde{\mathbf{P}}_n = (\mathbf{P}_n | \mathbf{p}_n)$, referentes, respectivamente, a uma câmera do sistema original e à correspondente no sistema em configuração canônica (diz-se que a matriz $\tilde{\mathbf{P}}_n$ retifica a imagem \mathbf{I}_o gerada pela câmera de matriz $\tilde{\mathbf{P}}_o$). Se $\tilde{\mathbf{w}}$ é um ponto arbitrário em coordenadas homogêneas e $\tilde{\mathbf{m}}_o$ e $\tilde{\mathbf{m}}_n$ são as coordenadas das imagens de $\tilde{\mathbf{w}}$ geradas por $\tilde{\mathbf{P}}_o$ e $\tilde{\mathbf{P}}_n$, tem-se [3]:

$$\begin{cases} \tilde{\mathbf{m}}_o = \tilde{\mathbf{P}}_o \cdot \tilde{\mathbf{w}} \\ \tilde{\mathbf{m}}_n = \tilde{\mathbf{P}}_n \cdot \tilde{\mathbf{w}} \end{cases} \quad (5.1)$$

Mas, pela equação do raio óptico de \mathbf{m}_o (reta que une esse ponto no plano de retina ao centro óptico), tem-se:

$$\mathbf{w} = \mathbf{c}_o + \lambda \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \quad (5.2)$$

com λ uma constante real.

Substituindo na segunda equação de (5.1), vem:

$$\tilde{\mathbf{m}}_n = \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \mathbf{c}_o + \lambda \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \\ 1 \end{pmatrix} \quad (5.3)$$

$$\therefore \tilde{\mathbf{m}}_n = \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \mathbf{c}_o \\ 1 \end{pmatrix} + \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \lambda \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \\ 0 \end{pmatrix} \quad (5.4)$$

Pela hipótese de que o centro óptico da câmera é mantido durante a transformação, tem-se $\mathbf{c}_o = \mathbf{c}_n$; logo:

$$\tilde{\mathbf{m}}_n = \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \mathbf{c}_n \\ 1 \end{pmatrix} + \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \lambda \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \\ 0 \end{pmatrix} \quad (5.5)$$

Pela equação do centro óptico:

$$\begin{aligned} \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \mathbf{c}_n \\ 1 \end{pmatrix} &= 0 \\ \therefore \tilde{\mathbf{m}}_n &= \tilde{\mathbf{P}}_n \cdot \begin{pmatrix} \lambda \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \\ 0 \end{pmatrix} \end{aligned} \quad (5.6)$$

Como $\tilde{\mathbf{P}}_n = (\mathbf{P}_n | \mathbf{p}_n)$, tem-se:

$$\tilde{\mathbf{m}}_n = \lambda \cdot \mathbf{P}_n \cdot \mathbf{P}_o^{-1} \cdot \tilde{\mathbf{m}}_o \quad (5.7)$$

A equação (5.7) permite calcular, a partir da imagem \mathbf{I}_o , a correspondente \mathbf{I}_n no sistema em configuração canônica; a transformação é linear e dada por $\mathbf{P}_n \cdot \mathbf{P}_o^{-1}$.²

Para que seja realizada a retificação de um par de imagens estéreo, é necessário ainda que se obtenham as duas matrizes de calibração $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$, referentes às duas câmeras em configuração canônica. A próxima seção trata do cálculo destas matrizes, a partir de $\tilde{\mathbf{P}}_{o1}$ e $\tilde{\mathbf{P}}_{o2}$.

5.3 Cálculo das matrizes de calibração retificadoras de um par de imagens estéreo

A etapa inicial da retificação de imagens consiste no cálculo de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$. Nesta pesquisa, adotou-se para tanto o algoritmo proposto em [3] (o programa em MatLab fornecido é apresentado no apêndice C). Nesta abordagem, são impostas condições aos elementos destas matrizes de tal forma a garantir que os novos centros ópticos coincidam com os da configuração original, bem como que pontos homólogos nas novas imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} estejam na mesma linha. São ainda impostas outras restrições a estes elementos para que seja obtida uma solução única.

² O fato de ser desconhecida a constante λ é irrelevante, já que são adotadas coordenadas homogêneas. Com efeito, para o cálculo das coordenadas j e i do “pixel” correspondente ao ponto \mathbf{w} filmado, são usadas as equações 3.16 e 3.17, ou seja, $j=U/S$ e $i=V/S$, com $\mathbf{m}_n = (U \ V \ S)^T$. Estas razões independem do valor de λ .

São ao todo oito as condições propostas em [3] (a justificativa para cada uma delas é apresentada no apêndice C):

1. Os centros ópticos das câmeras descritas por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem coincidir, respectivamente, com os das câmeras $\tilde{\mathbf{P}}_{o1}$ e $\tilde{\mathbf{P}}_{o2}$;
2. $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem apresentar o mesmo plano focal;
3. Pontos homólogos nas imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} devem estar sempre na mesma linha (o que garante ainda que as novas linhas epipolares serão paralelas a um dos eixos das imagens);
4. As novas matrizes de calibração devem apresentar a normalização $\|\mathbf{a}_3\| = \|\mathbf{b}_3\| = 1$ (onde \mathbf{a}_3 e \mathbf{b}_3 representam a terceira linha de \mathbf{P}_{n1} e de \mathbf{P}_{n2} , respectivamente);
5. O plano focal de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ será paralelo à intersecção dos planos focais de $\tilde{\mathbf{P}}_{o1}$ e $\tilde{\mathbf{P}}_{o2}$;
6. Os planos projetados em $i=0$ e em $j=0$ nas duas imagens retificadas devem ser perpendiculares entre si (ortogonalidade dos sistemas de referência retificados);
7. Os pontos principais i_0 e j_0 das matrizes $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem ser nulos;
8. As resoluções horizontal e vertical das câmeras $\tilde{\mathbf{P}}_{o1}$ e $\tilde{\mathbf{P}}_{o2}$ devem ser matidas na nova configuração representada por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$.

Estas condições conduzem a quatro sistemas de equações cuja resolução sucessiva permite obter os elementos de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$. Sendo

$$\tilde{\mathbf{P}}_{n1} = \left(\begin{array}{c|c} \mathbf{a}_1^T & a_{14} \\ \mathbf{a}_2^T & a_{24} \\ \mathbf{a}_3^T & a_{34} \end{array} \right) \text{ e } \tilde{\mathbf{P}}_{n2} = \left(\begin{array}{c|c} \mathbf{b}_1^T & b_{14} \\ \mathbf{b}_2^T & b_{24} \\ \mathbf{b}_3^T & b_{34} \end{array} \right) \quad (5.8)$$

tem-se:

$$\begin{cases} \mathbf{a}_3^T \cdot \mathbf{c}_1 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot \mathbf{c}_2 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot (\mathbf{f}_1 \wedge \mathbf{f}_2) = 0 \\ \|\mathbf{a}_3\| = 1 \end{cases} \quad \begin{cases} \mathbf{a}_2^T \cdot \mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{c}_2 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_2\| = \alpha_v \end{cases} \quad \begin{cases} \mathbf{a}_1^T \cdot \mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_1\| = \alpha_u \end{cases} \quad \begin{cases} \mathbf{b}_1^T \cdot \mathbf{c}_2 + b_{14} = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{b}_1\| = \alpha_u \end{cases} \quad (5.9)$$

onde:

- \mathbf{f}_1 e \mathbf{f}_2 são os vetores unitários perpendiculares aos planos focais das câmeras originais (constituem a terceira linha de \mathbf{P}_{o1} e \mathbf{P}_{o2} , respectivamente);
- α_u e α_v representam a distância focal da câmera \mathbf{P}_{o1} em número de *pixels* no eixo horizontal e vertical, respectivamente, ou seja:

$$\alpha_u = \frac{f_1}{\delta u} \quad (5.10)$$

$$\alpha_v = \frac{f_1}{\delta v} \quad (5.11)$$

Cabe ressaltar que os termos \mathbf{b}_2 , b_{24} , \mathbf{b}_3 , b_{34} não são explicitados nas equações 5.9. Isto ocorre porque, em decorrência das oito condições apresentadas:

$$\begin{pmatrix} \mathbf{b}_2 & | & b_{24} \\ \mathbf{b}_3 & | & b_{34} \end{pmatrix} = \begin{pmatrix} \mathbf{a}_2 & | & a_{24} \\ \mathbf{a}_3 & | & a_{34} \end{pmatrix}. \quad (5.12)$$

No apêndice C, demonstra-se de que forma as oito condições apresentadas conduzem aos sistemas de equações 5.9, bem como à igualdade 5.12 (esta demonstração foi proposta em [3]).

5.4 Retificação das imagens do padrão de calibração

O padrão de calibração descrito no capítulo 4 foi utilizado não apenas durante o procedimento experimental de calibração das câmeras mas também para teste do algoritmo de retificação empregado. Como há 32 pontos de referência (centróides dos quadrados claros) cujas posições são bem determinadas, pôde-se comparar suas coordenadas i e j nas imagens fornecidas pela câmera CCD em duas posições distintas (não em configuração canônica) com aquelas das imagens retificadas. O objetivo foi verificar se as coordenadas i de pontos correspondentes nas duas imagens tornaram-se idênticas após a retificação, ou seja, se os pontos homólogos, antes em linhas diferentes, passaram a se situar em linhas comuns nas duas imagens.

Outra verificação do algoritmo de retificação será ainda feita no capítulo 8; a reconstituição das coordenadas espaciais dos 32 pontos de referência a partir das imagens retificadas mostrará que as novas matrizes $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ calculadas segundo 5.9 representam bem a transformação perspectiva realizada pelo sistema hipotético de duas câmeras em configuração canônica.

A figura 5.4 apresenta as imagens do padrão de calibração fornecidas pela câmera CCD em duas posições distintas, de forma que os planos focais adotados formam um ângulo de 160° entre si. Conforme se observa desta figura, os pontos de referência homólogos estão em linhas e colunas distintas nas duas imagens; estas coordenadas, calculadas com o auxílio do programa *centroid.m* fornecido no apêndice B, são apresentadas na tabela 5.1 e na tabela 5.2 .

Já na figura 5.5, são mostradas as imagens do padrão de calibração retificadas, calculadas com o uso do programa *retifica.m* (detalhes da utilização deste são apresentados no capítulo 7). Observe-se que os centróides homólogos apresentam neste caso a mesma coordenada i (ver tabela 5.3 e tabela 5.4), o que significa que eles se situam sempre na mesma linha. Cabe ressaltar que o processo de retificação gerou uma faixa escura à esquerda da segunda imagem apresentada na figura 5.5; isto se deveu ao fato de estar sendo representada uma região do espaço que seria filmada pelo sistema representado pelas transformações perspectivas $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$, mas que não foi amostrada pelo sistema real $\tilde{\mathbf{P}}_{o1}$, $\tilde{\mathbf{P}}_{o2}$ (os *pixels* correspondentes na imagem foram então representados com o nível de cinza 0). Esta faixa foi, no entanto, omitida na figura 5.5, para facilitar a visualização.

Também se observa na figura 5.5 que a retificação das imagens do padrão de calibração levou à formação de padrões em nível de cinza 0 (linhas escuras que cortam as imagens); isto ocorreu em decorrência da taxa de amostragem finita das câmeras digitais. Assim, estes padrões representam pontos que seriam filmados pela configuração canônica representada por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$, mas que não foram amostrados pelo sistema real.

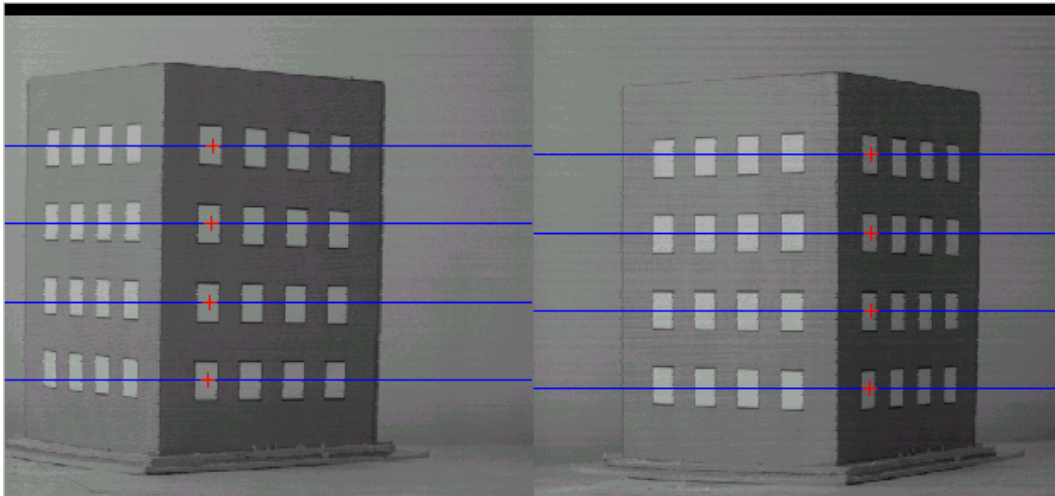


Figura 5.4 : Imagens não - retificadas do padrão de calibração fornecidas pela câmera CCD em duas posições distintas, com um ângulo de 160° entre os planos focais. Observar que os pontos homólogos destacados nas duas imagens não estão alinhados.

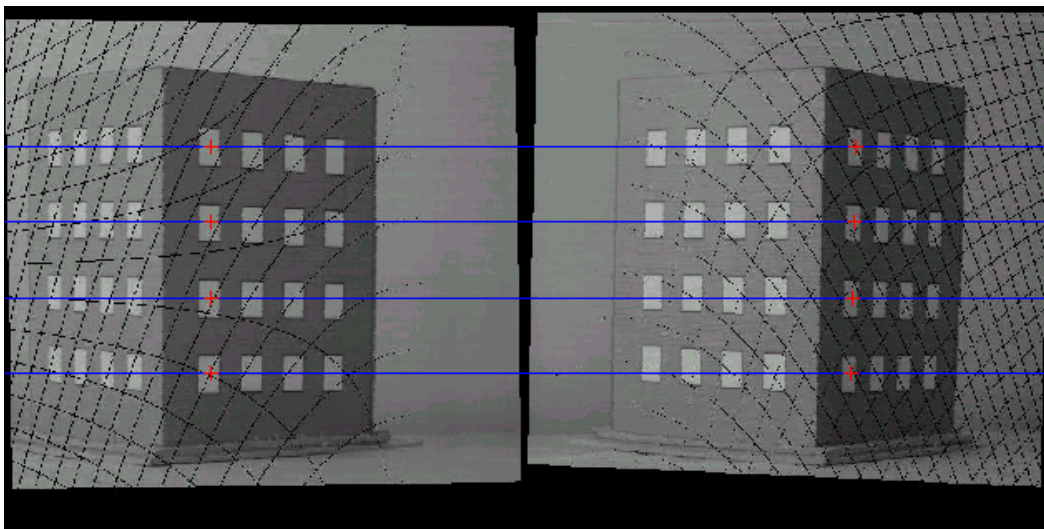


Figura 5.5 : Imagens retificadas do padrão de calibração. Observar que os pontos homólogos destacados nas duas imagens estão alinhados.

Ponto de referência	Coordenadas Calculadas a Partir			
	da Matriz de Transformação		Do Programa Centroid.m	
	<i>i</i>	<i>j</i>	<i>i</i>	<i>j</i>
1	140.97	48.69	140.77	49.23
2	139.47	73.94	139.40	74.02
3	137.91	100.26	138.04	99.96
4	136.29	127.73	136.54	127.44
5	212.83	47.76	212.75	47.86
6	212.84	72.92	212.60	72.66
7	212.84	99.16	212.96	98.62
8	212.85	126.53	213.04	126.25
9	284.35	46.83	284.35	46.84
10	285.84	71.91	285.74	71.78
11	287.40	98.06	287.52	97.66
12	289.02	125.34	288.99	125.00
13	355.53	45.90	355.92	46.31
14	358.49	70.90	358.59	70.88
15	361.57	96.96	361.48	96.93
16	364.79	124.15	364.39	124.20
17	137.88	201.28	138.26	201.37
18	141.06	244.73	141.16	244.87
19	144.15	286.82	144.12	286.80
20	147.14	327.63	146.99	327.44
21	214.88	199.88	214.86	200.32
22	216.85	243.24	216.86	243.54
23	218.75	285.25	218.68	285.39
24	220.60	325.98	220.46	325.98
25	291.49	198.49	291.47	199.0
26	292.29	241.76	291.91	242.09
27	292.98	283.69	292.98	283.77
28	293.69	324.34	293.47	324.31
29	367.70	197.11	367.50	197.39
30	367.26	240.29	367.06	240.29
31	366.84	282.14	367.16	281.82
32	366.43	322.71	366.97	322.34

Tabela 5.1 : Coordenadas dos pontos de referência do padrão de calibração, na imagem gerada pela câmera CCD na posição 1 (a numeração dos pontos é feita de acordo com a figura 4.1, e a posição das câmeras de acordo com a figura 7.3).

Ponto de referência	Coordenadas Calculadas a Partir			
	da Matriz de Transformação		Do Programa Centroid.m	
	<i>i</i>	<i>j</i>	<i>i</i>	<i>j</i>
1	151.40	128.07	151.25	128.45
2	149.51	168.69	149.57	168.74
3	147.56	210.56	147.78	210.06
4	145.54	253.76	146.02	253.18
5	225.19	127.97	224.89	128.15
6	224.43	168.49	224.07	168.36
7	223.64	210.26	223.73	209.90
8	222.83	253.38	222.69	252.90
9	298.64	127.87	298.53	127.80
10	298.99	168.29	298.96	168.19
11	299.36	209.95	299.24	209.78
12	299.74	252.92	299.54	252.52
13	371.74	127.77	372.35	127.97
14	373.20	168.08	373.45	168.34
15	374.71	209.65	374.68	209.83
16	376.27	252.51	375.84	252.49
17	146.77	327.73	147.26	327.56
18	149.92	355.87	150.10	356.10
19	152.95	382.82	153.02	383.10
20	155.85	408.68	155.79	408.65
21	223.57	327.13	223.48	327.40
22	225.08	225.21	225.01	355.80
23	226.53	382.12	226.36	382.38
24	227.91	407.92	227.55	407.92
25	300.00	326.54	299.47	327.08
26	299.88	354.56	300.15	355.01
27	299.76	381.42	299.76	381.44
28	299.65	407.18	299.41	406.98
29	376.06	325.94	376.03	326.29
30	374.32	353.91	374.45	353.97
31	372.65	380.72	372.78	380.29
32	371.05	406.44	371.47	405.71

Tabela 5.2 : Coordenadas dos pontos de referência do padrão de calibração, na imagens geradas pela câmera CCD na posição 2 (a numeração dos pontos é feita de acordo com a figura 4.1, e a posição das câmeras de acordo com a figura 7.3).

Ponto de referência	Coordenadas Calculadas a Partir			
	da Matriz de Transformação		Do Programa Centroid.m	
	<i>i</i>	<i>j</i>	<i>i</i>	<i>j</i>
1	395.68	54.22	395.72	54.61
2	396.96	80.79	396.77	80.78
3	398.29	108.38	397.98	107.92
4	399.68	137.06	399.89	136.57
5	312.82	54.38	321.81	54.28
6	312.70	80.88	322.02	80.49
7	312.59	108.4	321.92	107.57
8	312.46	137.01	321.37	136.49
9	248.83	54.54	247.79	54.55
10	246.83	80.98	246.91	80.66
11	245.28	108.43	245.49	107.72
12	243.66	136.96	244.06	136.61
13	175.21	54.69	174.64	55.13
14	172.34	81.07	172.81	81.00
15	169.35	108.45	169.88	108.12
16	166.25	136.91	166.08	137.04
17	397.32	213.38	396.86	213.46
18	393.67	258.10	393.95	258.13
19	390.15	301.17	389.95	300.94
20	386.76	342.66	387.01	342.50
21	319.08	213.13	319.27	213.41
22	316.91	257.74	317.19	257.86
23	314.82	300.70	315.06	300.83
24	321.81	342.11	312.59	341.96
25	241.24	212.88	241.48	213.29
26	240.54	257.38	241.03	257.67
27	239.87	300.24	240.09	300.22
28	239.22	341.55	239.47	341.38
29	163.82	212.63	164.23	212.80
30	164.57	257.06	164.62	257.19
31	165.30	299.79	164.60	299.60
32	166.00	341.00	165.69	340.22

Tabela 5.3 : Coordenadas dos pontos de referência do padrão de calibração, na imagem 1 retificada (a numeração dos pontos é feita de acordo com a figura 4.1, e a posição das câmeras de acordo com a figura 7.3).

Ponto de referência	Coordenadas Calculadas a Partir			
	da Matriz de Transformação		Do Programa Centroid.m	
	<i>i</i>	<i>j</i>	<i>i</i>	<i>j</i>
1	395.68	820.54	395.79	821.00
2	396.96	861.63	397.09	861.66
3	398.29	904.30	398.36	903.86
4	399.68	948.66	398.96	947.98
5	312.82	818.80	322.33	818.96
6	312.71	859.76	321.75	859.56
7	312.59	902.29	321.19	901.83
8	312.46	946.49	321.11	946.06
9	248.83	817.07	248.20	816.95
10	246.83	857.89	247.25	857.82
11	245.28	900.28	245.70	899.73
12	243.66	944.32	243.64	943.67
13	175.21	815.35	174.89	815.52
14	172.34	856.04	172.50	856.25
15	169.35	898.28	169.73	898.54
16	166.25	942.17	167.19	941.96
17	397.32	1025.33	396.64	1025.20
18	393.67	1054.70	393.59	1054.98
19	390.15	1082.97	390.27	1083.15
20	386.76	1110.22	386.80	1110.01
21	319.08	1022.95	319.24	1023.30
22	316.91	1052.29	316.87	1052.79
23	314.82	1080.54	314.92	1080.77
24	321.81	1107.76	313.24	1107.74
25	241.24	1020.59	241.30	1020.96
26	240.54	1049.90	240.47	1050.23
27	239.87	1078.12	239.98	1078.13
28	239.22	1105.32	239.57	1104.98
29	163.82	1018.23	163.81	1018.42
30	164.57	1047.51	164.83	1047.47
31	165.30	1075.71	165.17	1075.28
32	166.00	1102.89	165.56	1101.90

Tabela 5.4 : Coordenadas dos pontos de referência do padrão de calibração, na imagem 2 retificada (a numeração dos pontos é feita de acordo com a figura 4.1, e a posição das câmeras de acordo com a figura 7.3).

Capítulo 6

Detecção de Bordas

Uma imagem contém uma grande quantidade e variedade de informações. Estas informações podem ser:

- Presença ou ausência de um objeto;
- A orientação do objeto;
- As cores do objeto na imagem, que podem fornecer informações sobre sua temperatura (se for usado um sensor infravermelho para coletá-la);
- Localização do objeto no espaço (objetivo deste trabalho); etc.

Uma imagem pode ter uma ou mais das características acima, de forma que para um dado processamento existe redundância de informações que aumenta o esforço computacional.

No sistema desenvolvido o tipo de informação que se deseja extrair é a localização do objeto no espaço. Como demonstrado em capítulos anteriores, a extração deste tipo de informação a partir de imagens é conseguido utilizando-se um sistema estéreo.

O passo mais importante no cálculo das coordenadas estéreo é a obtenção dos pontos correspondentes, ou seja, que ponto da imagem da direita corresponde a um determinado ponto da imagem da esquerda. Este processamento é responsável pelo maior esforço computacional.

Uma maneira de diminuir este esforço é reduzir o número de pontos que serão utilizados durante o processo de correspondência. Para se obter as coordenadas 3D de um objeto é necessário apenas calcular as coordenadas dos extremos, logo só estes pontos precisam entrar no processo de correspondência.

A detecção dos pontos que fazem parte dos extremos do objeto foi realizada utilizando-se a técnica de detecção de bordas de Sobel. A explicação detalhada desta técnica está nas seções seguintes. Porém é necessário entender algumas técnicas de processamento de imagens primeiramente.

6.1 Thresholding

Thresholding é um processo de binarização de uma imagem utilizando um limiar. Este método produz uma imagem, em que aparece a silhueta do objeto de interesse em preto e o fundo em branco. Na verdade, é produzida, na saída do operador, uma imagem binária onde o bit 0 representa um destes dois níveis cinza e o bit 1 o outro. No MatLab o bit 0 representa o preto e o bit 1 representa o branco, esta convenção de cor será usada deste ponto em diante. No caso de imagens em nível de cinza, imagem original, a faixa de valores de intensidade vai de 0 (branco) até 1 (preto).

Neste texto será utilizado uma letra maiúscula, por exemplo F, para denominar uma imagem e a letra minúscula correspondente, f, para denominar a matriz de valores de intensidade de cinza correspondente. Em todos os exemplos será utilizado uma imagem em níveis de cinza, que é o tipo de imagem utilizado no sistema, mas as idéias podem ser estendidas para imagens coloridas.

Considerando a imagem F, figura 6.1, e adotando a configuração do MatLab de escala de cinza, que varia de 0 (preto) até 1 (branco), obtêm-se a matriz f que representa esta imagem.

Utilizando-se o Programa 6.1 (Apêndice B), aplicou-se o *thresholding* para valores de limiar variando de 0,1 a 0,9, com incrementos de 0,1, e obtiveram-se as imagens mostradas na figura 6.3. Os programas que geraram todas as outras imagens exemplos deste capítulo se encontram também no Apêndice B.

Na imagem F original, distingue-se a forma do anel, já na figura 6.3 percebe-se que para valores de limiar menores que 0,5 este anel não é perceptível. Logo a escolha de valores de limiar muito próximos do valor 0 causa perda de informação sobre o objeto. Em compensação para valores de limiar muito próximos de 1, verifica-se um alto nível de ruído, que para valores menores é filtrado. Este alto nível de ruído pode causar, numa etapa posterior de processamento uma errônea identificação do objeto.

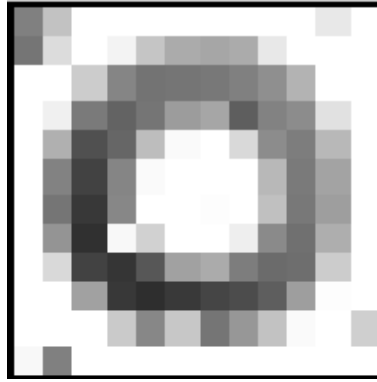


Figura 6.1: Imagem de um anel. Esta imagem será a base de todos os exemplos desenvolvidos neste capítulo.

$$f = \begin{bmatrix} 0,50 & 0,76 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 0,91 & 0,91 \\ 0,46 & 0,86 & 1,00 & 0,95 & 0,77 & 0,67 & 0,65 & 0,67 & 0,91 & 1,00 & 1,00 & 1,00 \\ 1,00 & 1,00 & 0,80 & 0,52 & 0,45 & 0,46 & 0,47 & 0,50 & 0,56 & 0,70 & 1,00 & 1,00 \\ 1,00 & 0,94 & 0,48 & 0,39 & 0,46 & 0,61 & 0,65 & 0,37 & 0,51 & 0,55 & 0,88 & 0,88 \\ 1,00 & 0,69 & 0,32 & 0,40 & 0,74 & 0,98 & 1,00 & 0,85 & 0,55 & 0,49 & 0,72 & 0,72 \\ 1,00 & 0,51 & 0,26 & 0,52 & 0,98 & 1,00 & 1,00 & 1,00 & 0,72 & 0,48 & 0,64 & 0,64 \\ 1,00 & 0,46 & 0,22 & 0,53 & 1,00 & 1,00 & 0,99 & 1,00 & 0,75 & 0,47 & 0,62 & 0,62 \\ 1,00 & 0,58 & 0,19 & 0,97 & 0,82 & 1,00 & 1,00 & 0,93 & 0,54 & 0,44 & 0,68 & 0,68 \\ 1,00 & 0,85 & 0,26 & 0,21 & 0,34 & 0,63 & 0,67 & 0,49 & 0,42 & 0,43 & 0,80 & 0,80 \\ 1,00 & 1,00 & 0,63 & 0,22 & 0,18 & 0,23 & 0,27 & 0,30 & 0,37 & 0,62 & 0,99 & 0,99 \\ 1,00 & 1,00 & 1,00 & 0,79 & 0,53 & 0,78 & 0,46 & 0,59 & 0,76 & 0,98 & 1,00 & 1,00 \\ 0,97 & 0,50 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 & 1,00 \end{bmatrix}$$

Figura 6.2: Matriz correspondente a imagem F.

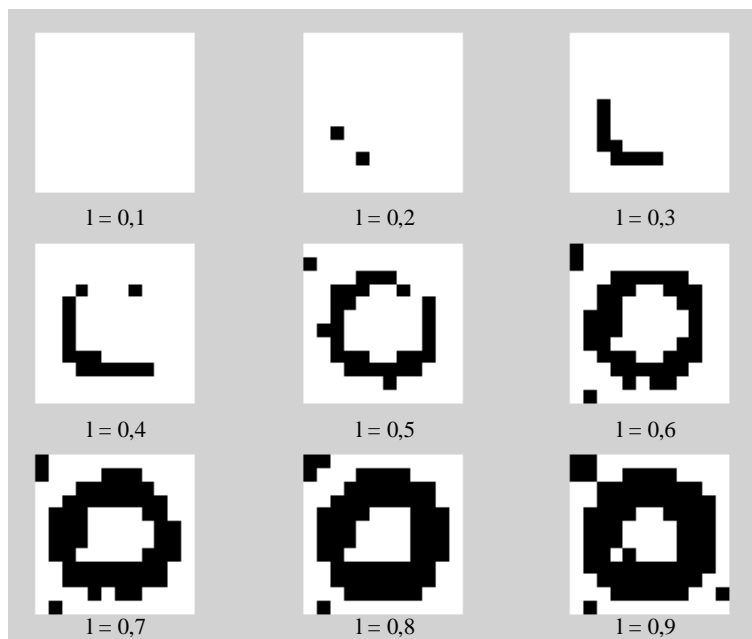


Figura 6.3 : Imagens de saída do operador *thresholding* para diferentes valores de limiar *l*.

Analisando as imagens da figura 6.3, percebe-se que o valor ideal de limiar estaria entre 0,7 e 0,8. Porém esta faixa de valores só é válida para esta imagem específica, sendo que para outra imagem esta faixa pode ser diferente.

Mesmo estando o limiar dentro desta faixa, a imagem de saída não é uma silhueta perfeita da imagem. A razão é que o *thresholding* elimina apenas parte do ruído da imagem, que surge devido distorções causadas pelo sensor fotosensível, pelo sistema de digitalização da imagem e reflexão da luz dentre outras.

6.2 Significado Matemático da Detecção de Bordas

Geralmente uma imagem apresenta um gradiente de iluminação, onde uma região apresenta-se mais escura que outra devido à não uniformidade da iluminação do objeto real. Neste tipo de imagem a utilização de *threshold* não leva a obtenção da silhueta do objeto sendo necessário outro tipo de processamento.

Neste caso um processo adequado para a identificação de objetos em níveis de cinza é a detecção de bordas na imagem, reduzindo-a a um conjunto de linhas que formam um modelo *wire-frame*. Este processo localiza as fronteiras entre as regiões da imagem que diferem entre si de acordo com uma certa medida de homogeneidade pelo cálculo da taxa de variação dos níveis de cinza. Quando esta taxa tem um valor alto, significa que naquela região existe uma borda.

Matematicamente, pode-se considerar a imagem como uma função de duas variáveis, onde a localização (x,y) na imagem tem intensidade $I(x,y)$ de luminosidade. Como a imagem é armazenada em uma matriz de pixels discretos, considera-se a função de intensidade luminosa como uma função contínua no espaço. O gráfico da figura 6.4 representa a função de luminosidade ao longo de uma linha de uma matriz de imagem hipotética. Defini-se a borda como o ponto em que a função de intensidade apresenta uma inclinação máxima.

Derivando a função de luminosidade encontra-se o gráfico da figura 6-5. Neste caso, a borda se encontra no ponto em que a função derivada de $I(x,y)$ apresenta valor máximo.

Pelo cálculo da segunda derivada obtêm-se a figura 6.6. A borda encontra-se no ponto em que a função da segunda derivada de $I(x,y)$ passa pelo zero.

6.3 Operadores DX e DY

Para derivar a matriz da imagem utilizam-se os operadores de diferenças parciais, que medem a taxa de variação de nível de cinza, um no sentido horizontal e outro no sentido vertical. Esses operadores digitais são análogos as derivadas parciais em cálculo clássico. Matematicamente, eles são definidos

$$\begin{aligned} [DX(f)](i, j) &= f(i, j) - f(i, j-1) \\ [DY(f)](i, j) &= f(i, j) - f(i+1, j) \end{aligned} \quad (6.1)$$

onde i representa a coluna da matriz e j a linha.

DX é a diferença do nível de cinza entre um *pixel* e o seu adjacente a esquerda. DY é a diferença do nível de cinza entre um *pixel* e o seu adjacente da linha de baixo.

O operador DX não é definido na primeira coluna da matriz, não há coluna anterior para subtraí-la. O operador DY não é definido na última linha da matriz. Logo a matriz de saída destes operadores será menor que a matriz original.

Aplicou-se os operadores DX e DY a matriz f , e obteve-se as matrizes g e h mostradas na figura 6.7 e as imagens G e H mostradas na figura 6.8.

Uma breve análise da matriz g mostra que o operador DX revela as bordas verticais, o operador atribui um alto valor de cinza aos pontos onde existe borda. Os valores negativos indicam a direção do contraste de luminosidade da matriz original f (a imagem F é mais escura à esquerda).

Análise idêntica a anterior mostra que DY revela a borda existente na horizontal, o operador atribui um alto valor de cinza aos pontos onde existe borda. Agora os valores negativos indicam que a imagem F é mais escura na base do que no topo.

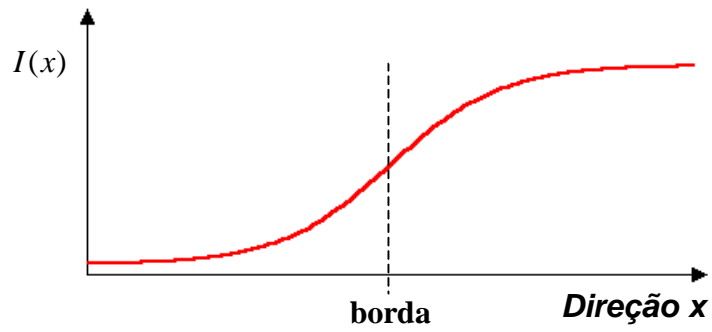


Figura 6.4 : Gráfico da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.

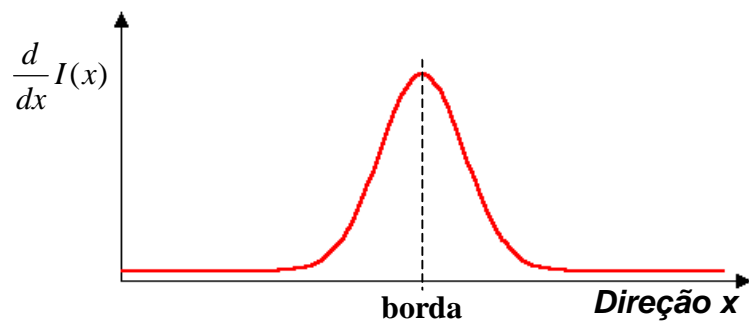


Figura 6.5 : Gráfico da derivada da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.

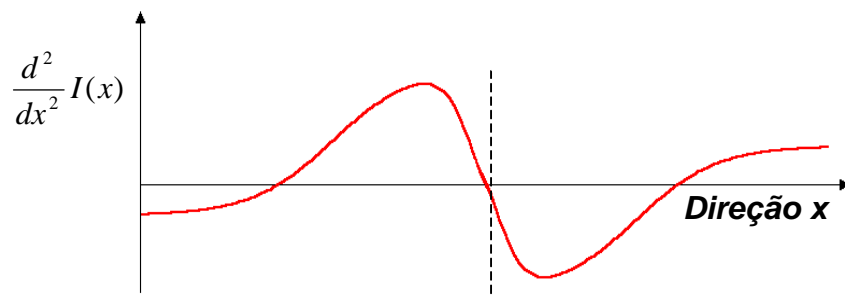


Figura 6.6 : Gráfico da segunda derivada da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.

$$g = DX(f) = \begin{bmatrix} 0,26 & 0,24 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & -0,09 & 0,09 & 0,09 \\ 0,40 & 0,14 & -0,05 & -0,18 & -0,10 & -0,02 & 0,02 & 0,24 & 0,09 & 0,00 & 0,00 & 0,00 \\ 0,00 & -0,20 & -0,28 & -0,07 & 0,01 & 0,01 & 0,03 & 0,06 & 0,14 & 0,30 & 0,00 & 0,00 \\ -0,06 & -0,46 & -0,09 & 0,07 & 0,15 & 0,04 & -0,28 & 0,14 & 0,04 & 0,33 & 0,12 & 0,12 \\ -0,31 & -0,37 & 0,08 & 0,34 & 0,24 & 0,02 & -0,1 & -0,30 & -0,06 & 0,23 & 0,28 & 0,28 \\ -0,49 & -0,25 & 0,26 & 0,46 & 0,02 & 0,00 & 0,00 & -0,28 & -0,24 & 0,16 & 0,36 & 0,36 \\ -0,54 & -0,24 & 0,31 & 0,47 & 0,00 & -0,01 & 0,01 & -0,25 & -0,28 & 0,15 & 0,38 & 0,38 \\ -0,42 & -0,39 & 0,78 & -0,15 & 0,18 & 0,00 & -0,07 & -0,39 & -0,10 & 0,24 & 0,32 & 0,32 \\ -0,15 & -0,59 & -0,05 & 0,13 & 0,29 & 0,04 & -0,18 & -0,07 & 0,01 & 0,37 & 0,20 & 0,20 \\ 0,00 & -0,37 & -0,41 & -0,04 & 0,05 & 0,04 & 0,03 & 0,07 & 0,25 & 0,37 & 0,01 & 0,01 \\ 0,00 & 0,00 & -0,21 & -0,26 & 0,25 & -0,32 & 0,13 & 0,17 & 0,22 & 0,02 & -0,19 & -0,19 \\ -0,47 & 0,50 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix}$$

$$h = DY(f) = \begin{bmatrix} 0,04 & -0,10 & 0,00 & 0,05 & 0,23 & 0,33 & 0,35 & 0,33 & 0,09 & 0,00 & -0,09 & 0,00 \\ -0,54 & -0,14 & 0,20 & 0,43 & 0,32 & 0,21 & 0,18 & 0,17 & 0,35 & 0,30 & 0,00 & 0,00 \\ 0,00 & 0,06 & 0,32 & 0,13 & -0,01 & -0,15 & -0,18 & 0,13 & 0,05 & 0,15 & 0,12 & 0,00 \\ 0,00 & 0,25 & 0,16 & -0,01 & -0,28 & -0,37 & -0,35 & -0,48 & -0,04 & 0,06 & 0,16 & 0,00 \\ 0,00 & 0,18 & 0,04 & -0,12 & -0,24 & -0,02 & 0,00 & -0,15 & -0,17 & 0,01 & 0,08 & 0,00 \\ 0,00 & 0,05 & 0,03 & -0,01 & -0,02 & 0,00 & 0,01 & 0,00 & -0,03 & 0,01 & 0,02 & 0,00 \\ 0,00 & -0,12 & -0,07 & -0,44 & 0,18 & 0,00 & -0,01 & 0,07 & 0,21 & 0,03 & -0,06 & 0,00 \\ 0,00 & -0,27 & -0,37 & 0,76 & 0,48 & 0,37 & 0,33 & 0,44 & 0,12 & 0,01 & -0,12 & 0,00 \\ 0,00 & -0,15 & -0,37 & -0,01 & 0,16 & 0,40 & 0,40 & 0,19 & 0,05 & -0,19 & -0,19 & 0,00 \\ 0,00 & 0,00 & -0,37 & -0,57 & -0,35 & -0,55 & -0,19 & -0,29 & -0,39 & -0,36 & -0,01 & 0,19 \\ 0,03 & 0,50 & 0,00 & -0,21 & -0,47 & -0,22 & -0,54 & -0,41 & -0,24 & -0,02 & 0,00 & -0,19 \end{bmatrix}$$

Figura 6.7: Matrizes resultantes da aplicação dos operadores DX e DY sobre a matriz *f*.

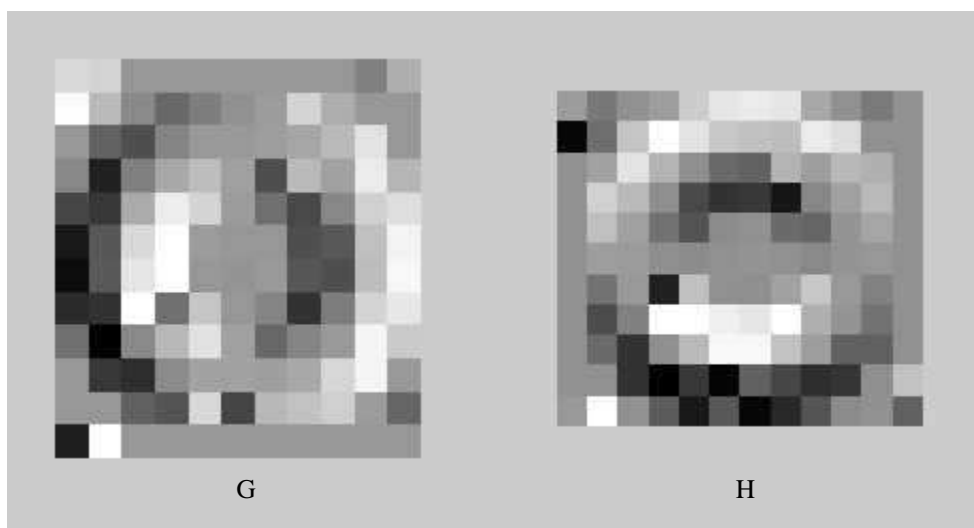


Figura 6.8: Matrizes resultantes da aplicação dos operadores DX e DY sobre a matriz *f*.

Os operadores DX e DY são dependentes da direção da variação de luminosidade. Sendo esta a sua maior desvantagem, já que a direção das bordas em uma imagem independem da direção e em muitos casos ocorrem curvas.

Para superar esta desvantagem utiliza-se um operador que combina as saídas dos operadores DX e DY , resultando num operador independente da direção.

Define-se gradiente como sendo o par ordenado, em que o primeiro elemento é o valor do *pixel* (i, j) da matriz de saída de DX , e o segundo é o valor do *pixel* equivalente (i, j) na matriz de saída DY , logo

$$\text{Grad}(f(i, j)) = [DX(f(i, j)), DY(f(i, j))] \quad (6.2)$$

O $\text{grad}(f)$ é um vetor bidimensional que define o módulo e a direção da variação de intensidade luminosa para cada *pixel* em f . Vários tipos de normas podem ser utilizadas para se obter a magnitude do vetor intensidade, três delas serão discutidas a seguir.

Considere o vetor $V = (x, y)$. A norma l_∞ é definida como o valor máximo em modulo do par (x, y) , matematicamente

$$\|V\|_\infty = \max\{|x|, |y|\} \quad (6.3)$$

A norma l_1 é definida como a soma dos módulos de x e y , matematicamente

$$\|V\|_1 = |x| + |y| \quad (6.4)$$

A norma l_2 é definida como a raiz quadrada da soma dos quadrados de x e y , matematicamente

$$\|V\|_2 = \sqrt{x^2 + y^2} \quad (6.5)$$

Define-se o operador GRADMAG0 como sendo aquele que utiliza a norma l_∞ para calcular a magnitude do gradiente, decorrente da aplicação dos operadores DX e DY. O diagrama deste operador está mostrado na figura 6.9 .

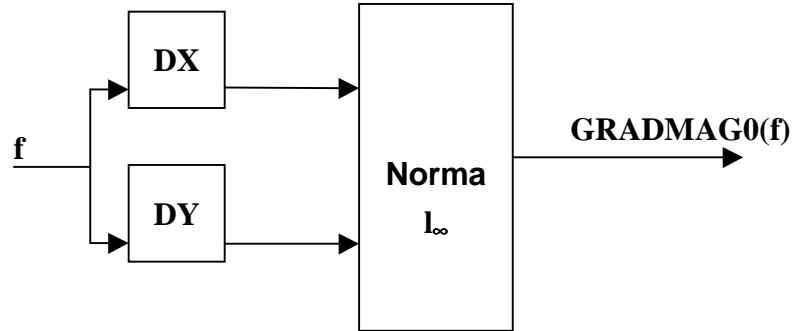


Figura 6.9 : Diagrama em blocos do operador GRAMADMAG0

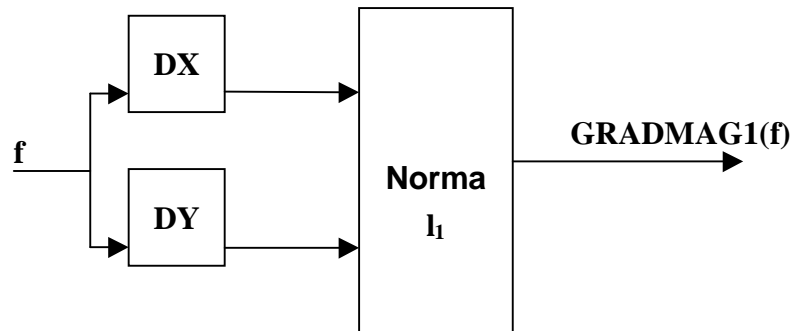


Figura 6.10 : Diagrama em blocos do operador GRAMADMAG1

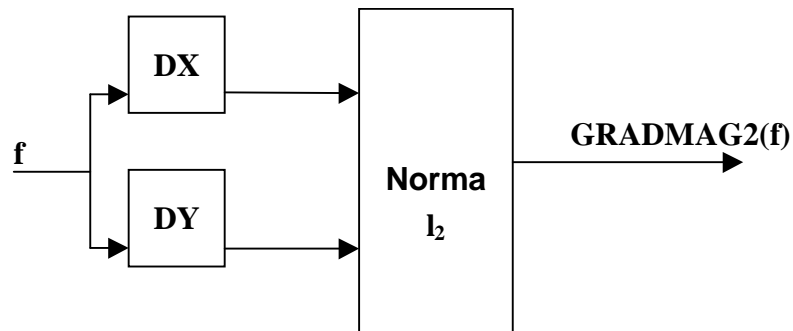


Figura 6.11 : Diagrama em blocos do operador GRAMADMAG2

Define-se operador GRADMAG1 como sendo aquele que utiliza a norma l_1 para calcular a magnitude do gradiente, decorrente da aplicação dos operadores DX e DY. O diagrama deste operador está mostrado na figura 6.10 .

Define-se operador GRADMAG2 como sendo aquele que utiliza a norma l_2 para calcular a magnitude do gradiente, decorrente da aplicação dos operadores DX e DY. O diagrama deste operador está mostrado na figura 6.11.

Como foi dito anteriormente o DX não é definido na primeira coluna e portanto tem uma coluna a menos que a imagem original. O operador DY, por outro lado não é definido na última linha tendo, então, uma linha a menos que a matriz f. Assim as matrizes de saída destes operadores apresentam tamanhos e índices diferentes. Uma maneira de resolver o problema é acrescentar uma coluna inicial em DX(f) e uma linha final DY(f) com valores indefinidos. Exemplos de valores indefinidos seriam valores fora da escala de luminosidade e o zero.

As matrizes a, b e c resultantes, respectivamente, da aplicação dos operadores GRADMAG0, GRADMAG1 e GRADMAG2 sobre o par [DX,DY] estão mostradas na figura 6.13.

As matrizes a, b e c não apresentam a estrutura de *wire-frame* , que é desejado no processo de detecção de bordas. Para se conseguir tal objetivo, aplica-se o *thresholding* após o processamento dos processadores discutidos anteriormente. A figura 6.12 mostra as imagens resultantes da aplicação do *threshold* de limiar igual a 0,3 nas matrizes a, b e c.



Figura 6. 12: Imagens A, B e C referentes as matrizes a, b e c com limiar de *thresholding* igual a 0,3.

$$\begin{aligned}
 a = \text{GRADMAG0}(f) &= \begin{bmatrix} 0,04 & 0,26 & 0,24 & 0,05 & 0,23 & 0,33 & 0,35 & 0,33 & 0,09 & 0,00 & 0,09 & 0,09 \\ 0,54 & 0,40 & 0,20 & 0,43 & 0,32 & 0,21 & 0,18 & 0,17 & 0,35 & 0,30 & 0,00 & 0,00 \\ 0,00 & 0,06 & 0,32 & 0,28 & 0,07 & 0,15 & 0,18 & 0,13 & 0,06 & 0,15 & 0,30 & 0,00 \\ 0,00 & 0,25 & 0,46 & 0,09 & 0,28 & 0,37 & 0,35 & 0,48 & 0,14 & 0,06 & 0,33 & 0,12 \\ 0,00 & 0,31 & 0,37 & 0,12 & 0,34 & 0,24 & 0,02 & 0,15 & 0,30 & 0,06 & 0,23 & 0,28 \\ 0,00 & 0,49 & 0,25 & 0,26 & 0,46 & 0,02 & 0,01 & 0,00 & 0,28 & 0,24 & 0,16 & 0,36 \\ 0,00 & 0,54 & 0,24 & 0,44 & 0,47 & 0,00 & 0,01 & 0,07 & 0,25 & 0,28 & 0,15 & 0,38 \\ 0,00 & 0,42 & 0,39 & 0,78 & 0,48 & 0,37 & 0,33 & 0,44 & 0,39 & 0,10 & 0,24 & 0,32 \\ 0,00 & 0,15 & 0,59 & 0,05 & 0,16 & 0,40 & 0,40 & 0,19 & 0,07 & 0,19 & 0,37 & 0,20 \\ 0,00 & 0,00 & 0,37 & 0,57 & 0,35 & 0,55 & 0,19 & 0,29 & 0,39 & 0,36 & 0,37 & 0,19 \\ 0,03 & 0,50 & 0,00 & 0,21 & 0,47 & 0,25 & 0,54 & 0,41 & 0,24 & 0,22 & 0,02 & 0,19 \\ 0,00 & 0,47 & 0,50 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix} \\
b = \text{GRADMAG1} &= \begin{bmatrix} 0,04 & 0,36 & 0,24 & 0,05 & 0,23 & 0,33 & 0,35 & 0,33 & 0,09 & 0,00 & 0,18 & 0,19 \\ 0,54 & 0,54 & 0,34 & 0,48 & 0,50 & 0,31 & 0,20 & 0,19 & 0,59 & 0,39 & 0,00 & 0,00 \\ 0,00 & 0,06 & 0,52 & 0,41 & 0,08 & 0,16 & 0,19 & 0,16 & 0,11 & 0,29 & 0,42 & 0,00 \\ 0,00 & 0,31 & 0,62 & 0,10 & 0,35 & 0,52 & 0,39 & 0,76 & 0,18 & 0,10 & 0,49 & 0,12 \\ 0,00 & 0,49 & 0,43 & 0,20 & 0,58 & 0,26 & 0,02 & 0,30 & 0,47 & 0,07 & 0,31 & 0,28 \\ 0,00 & 0,54 & 0,29 & 0,27 & 0,48 & 0,02 & 0,01 & 0,00 & 0,31 & 0,25 & 0,18 & 0,36 \\ 0,00 & 0,66 & 0,27 & 0,75 & 0,65 & 0,00 & 0,02 & 0,08 & 0,46 & 0,31 & 0,21 & 0,38 \\ 0,00 & 0,69 & 0,46 & 1,54 & 0,63 & 0,55 & 0,33 & 0,51 & 0,51 & 0,11 & 0,36 & 0,32 \\ 0,00 & 0,30 & 0,96 & 0,06 & 0,29 & 0,69 & 0,44 & 0,37 & 0,12 & 0,20 & 0,56 & 0,20 \\ 0,00 & 0,00 & 0,74 & 0,98 & 0,39 & 0,60 & 0,23 & 0,32 & 0,46 & 0,61 & 0,38 & 0,20 \\ 0,03 & 0,50 & 0,00 & 0,42 & 0,73 & 0,47 & 0,86 & 0,54 & 0,41 & 0,24 & 0,02 & 0,38 \\ 0,00 & 0,47 & 0,50 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix} \\
c = \text{GRADMAG2}(f) &= \begin{bmatrix} 0,04 & 0,28 & 0,24 & 0,05 & 0,23 & 0,33 & 0,35 & 0,33 & 0,09 & 0,00 & 0,13 & 0,09 \\ 0,54 & 0,42 & 0,24 & 0,43 & 0,37 & 0,23 & 0,18 & 0,17 & 0,42 & 0,31 & 0,00 & 0,00 \\ 0,00 & 0,06 & 0,38 & 0,31 & 0,07 & 0,15 & 0,18 & 0,13 & 0,08 & 0,21 & 0,32 & 0,00 \\ 0,00 & 0,26 & 0,49 & 0,09 & 0,29 & 0,40 & 0,35 & 0,56 & 0,15 & 0,07 & 0,37 & 0,12 \\ 0,00 & 0,36 & 0,37 & 0,14 & 0,42 & 0,24 & 0,02 & 0,21 & 0,34 & 0,06 & 0,24 & 0,28 \\ 0,00 & 0,49 & 0,25 & 0,26 & 0,46 & 0,02 & 0,01 & 0,00 & 0,28 & 0,24 & 0,16 & 0,36 \\ 0,00 & 0,55 & 0,24 & 0,54 & 0,50 & 0,00 & 0,01 & 0,07 & 0,33 & 0,28 & 0,16 & 0,38 \\ 0,00 & 0,50 & 0,40 & 1,08 & 0,50 & 0,41 & 0,33 & 0,45 & 0,41 & 0,10 & 0,27 & 0,32 \\ 0,00 & 0,21 & 0,70 & 0,05 & 0,21 & 0,49 & 0,40 & 0,26 & 0,09 & 0,19 & 0,42 & 0,20 \\ 0,00 & 0,00 & 0,52 & 0,70 & 0,35 & 0,55 & 0,19 & 0,29 & 0,40 & 0,44 & 0,37 & 0,19 \\ 0,03 & 0,50 & 0,00 & 0,30 & 0,54 & 0,33 & 0,63 & 0,43 & 0,29 & 0,22 & 0,02 & 0,26 \\ 0,00 & 0,47 & 0,50 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 & 0,00 \end{bmatrix}
 \end{aligned}$$

Figura 6.13 : Matrizes resultantes da aplicação dos operadores GRADMAG0, GRADMAG1 e GRADMAG2 sobre a matriz f

Os operadores DX e DY podem ser vistos como máscaras, onde DX e DY são dados pelas matrizes:

$$DX = G1 = [-1 \quad 1] \qquad DY = G2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad (6.6)$$

Levando em consideração este dado e a utilização do *threshold*, a detecção de bordas pode ser descrita pelo diagrama em blocos da figura 6.14.

As máscaras G1 e G2 determinam uma medida de gradiente na implementação do detector de bordas. Assim diferentes formas de detecção podem ser conseguidas variando-se as matrizes G1 e G2. Exemplos de máscaras utilizadas na detecção de bordas são: a máscara de Prewitt e a máscara de Sobel.

As máscaras de Prewitt são dadas por:

$$P1 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad P2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (6.7)$$

As máscaras de Sobel são dadas por:

$$S1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad P2 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (6.8)$$

A figura 6.15 mostra as figuras de saída para cada uma das combinações de máscaras e normas. Para obtenção destas imagens foi utilizado um limiar de *threshold* igual a 1.

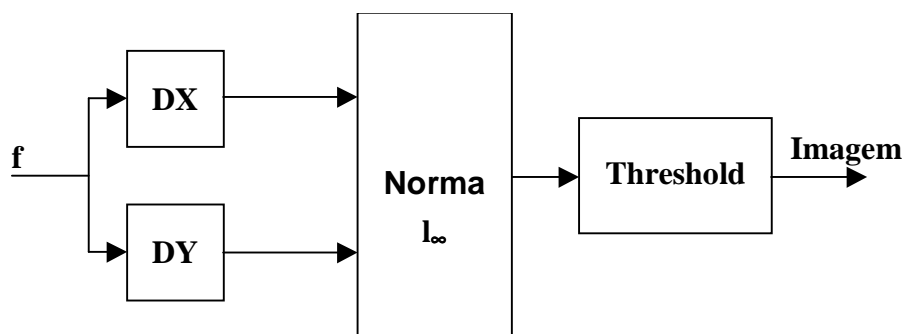


Figura 6.14 : Diagrama em blocos de um processo de detecção de bordas.

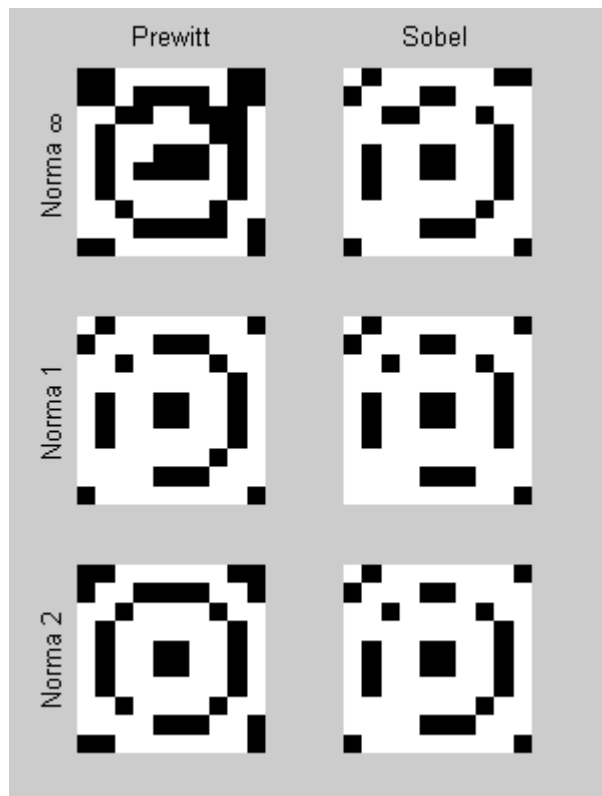


Figura 6.15 : Imagens de saída para cada uma das combinações de máscara (Prewitt e Sobel) e norma (l_∞ , l_1 e l_2).

Para o sistema desenvolvido, utilizou-se a técnica de detecção bordas de Sobel, com Norma l_2 e limiar de threshold igual a 0.03, figura 6.16.

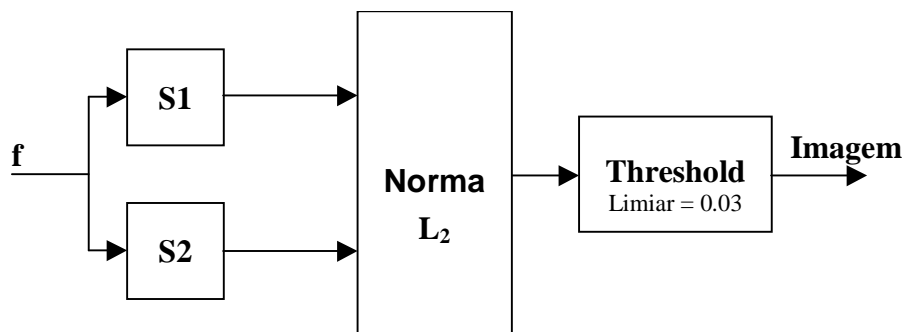


Figura 6.16 : Diagrama em blocos do processo de detecção de bordas implementado.

Capítulo 7

Metodologia

7.1 Aquisição de Imagens

Para a realização da fase experimental do projeto, foram construídos dois padrões de pontos de referência, para a calibração da câmera utilizada, e 4 prismas de bases diferentes, para teste dos algoritmos implementados. Estes objetos, construídos com papelão e revestidos em plástico adesivo, são mostrados na figura 7. 1 .



Figura 7 . 1: Objetos utilizados durante a parte experimental. A partir da central superior no sentido horário: padrão com trinta e dois pontos, padrão com seis pontos, cubo, prisma de base pentagonal, prisma com base hexagonal e prisma com base octogonal.

As imagens dos padrões de calibração e dos sólidos foram adquiridas utilizando-se uma câmera CCD monocromática da Pulnix, Série TM-6, conectada a um PC pela placa frame grabber óculos controlada pelo “*driver*” ODTCX. As imagens foram gravadas em formato TIFF pelo aplicativo ODCI, executado em ambiente DOS. No PC, com *clock* de 300 MHz e 128 MB de RAM, foram implementados os algoritmos utilizados.

O sistema foi construído sobre uma mesa de formica branca, sobre a qual fixaram-se dois papelões quadriculados (5 cm x 5 cm) com fita crepe, para posicionamento da câmera e do objeto a ser filmado. A figura 7.2 e a Tabela 7.1 descrevem a disposição destes elementos sobre a mesa.

Deve-se ressaltar que foi utilizada apenas uma câmera durante o experimento; assim, para a simulação do sistema estéreo, alternou-se a câmera entre as posições A e B. Em cada posição existia uma base de papelão, figura 7.3, que evitava o deslocamento da câmera em relação às posições em que se efetuou a calibração.

Inicialmente havia sido utilizada uma base em forma de L, que no entanto não permitia uma boa fixação da câmera. Esta falta de precisão levou a erros durante a reconstituição, tornando necessária a mudança desta base para outra de forma retangular.

Ponto	X(mm)	Y(mm)
1	191	35
2	163	28
3	162	29
4	190	34

Ponto	X(mm)	Y(mm)
5	183	75
6	150	68
7	154	68
8	183	73

Tabela 7.1: Posicionamento dos vértices da câmera em relação ao referencial O, para as duas situações consideradas.

Dois aspectos devem ser destacados acerca das imagens geradas pelo aplicativo ODCI. Em primeiro lugar, embora pelo modelo da câmera escura de orifício (figura 3.1) deva haver uma inversão da imagem em relação ao objeto real, o programa realiza uma nova inversão para que a imagem resultante tenha a mesma orientação do objeto com respeito à câmera. Durante todo o projeto foram utilizadas as imagens neste mesmo formato, sem a preocupação de manter a orientação prevista pelo modelo da câmera escura de orifício. Conforme discutido no capítulo 8, isto não prejudica o funcionamento do sistema implementado já que a calibração é também efetuada com este tipo de imagem, resultando em matrizes de transformação que mapeiam corretamente as coordenadas tridimensionais em *pixels* nesta configuração.

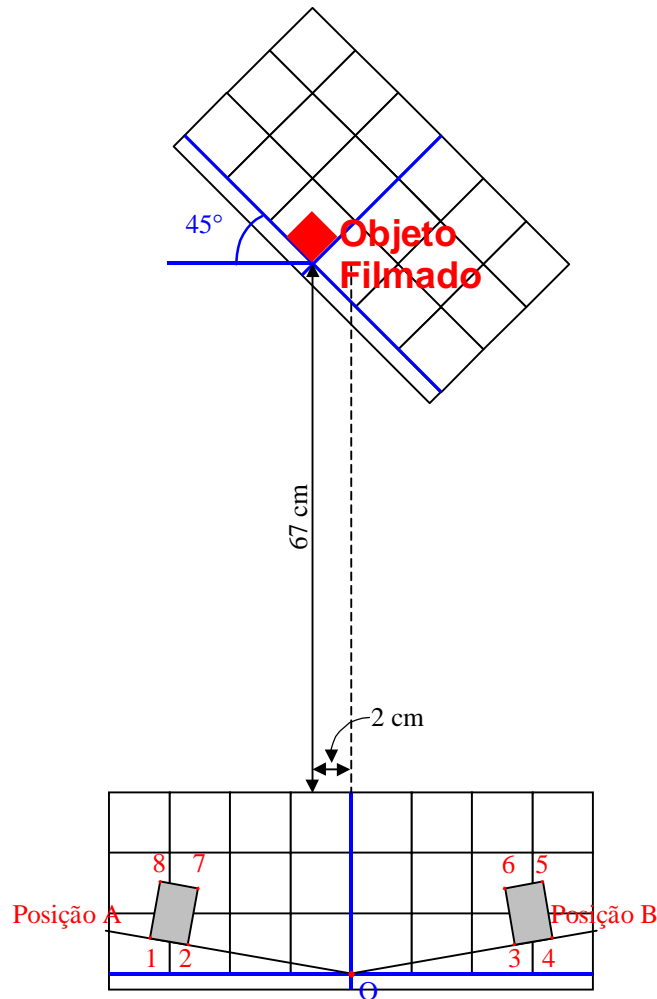


Figura 7.2 : Vista superior da mesa de trabalho. A posições A e B indicam as possíveis posições que a câmera poderia ocupar.

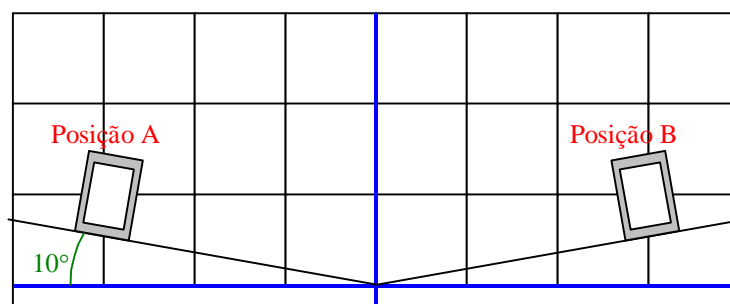


Figura 7.3: Base de papelão que firmava a câmera na mesa.

Além disso, o formato TIFF empregado pelo ODCI não é reconhecido pelo MATLAB 5.0, em que foram implementados todos os algoritmos. Tornou-se necessário assim converter os arquivos gerados para outro formato. Optou-se pelo JPEG, que tem a vantagem de compactar as imagens gerando arquivos menores.

7.1 Calibração da câmera

Dois programas foram elaborados para a calibração da câmera utilizada. O primeiro deles, *centroid.m*, permite o cálculo das coordenadas dos centróides de cada quadrado claro nas imagens do padrão de calibração (figura 4.1). Conforme discutido no capítulo 4, estes centróides são utilizados como pontos de referência durante a calibração propriamente dita.

A figura 7.4 mostra uma janela de operação deste programa. São calculadas as médias aritméticas das coordenadas j e i de cada *pixel* no interior do quadro em azul cujos níveis de cinza estão acima de um limiar, estabelecido de forma a diferenciar os quadrados claros do fundo escuro. As dimensões e a posição de cada quadro e o limiar são definidos pelo usuário.

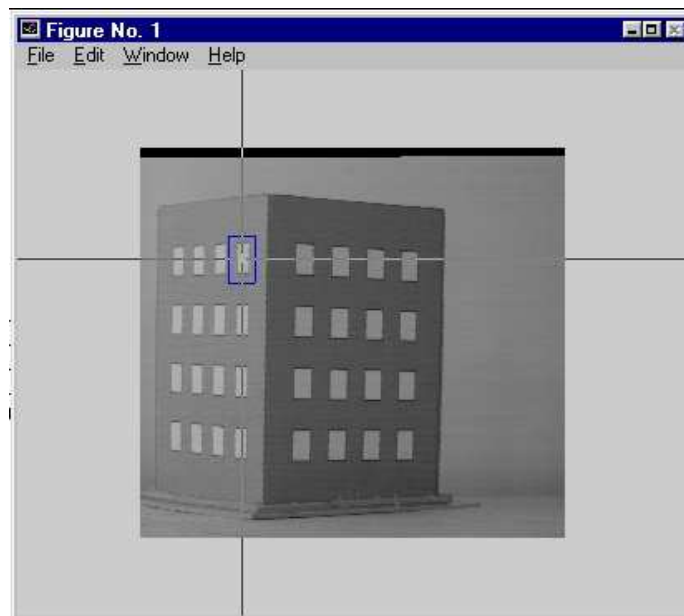


Figura 7.4: Janela de operação do programa *centroid.m*.

Nas tabelas 5.1 e 5.2, são fornecidas as coordenadas dos 32 centróides do padrão de calibração, nas imagens originais fornecidas pelo ODCI e nas retificadas.

O outro programa, *calib.m*, permite o cálculo da matriz de transformação perspectiva a partir dos pontos de referência fornecidos por *centroid.m*, utilizando a equação 4.3. Calcula ainda os parâmetros intrínsecos e extrínsecos da câmera com o uso do algoritmo de Ganapathy (seção 4.3).

7.2 Retificação das imagens estéreo

Para o cálculo das matrizes de calibração retificadoras, utilizou-se o programa *rectify.m* fornecido em [3], que permite solucionar os sistemas de equações 5.3.

Foi elaborado ainda, nesta pesquisa, o programa *retifica.m*, que recebe como entrada as matrizes retificadoras e as imagens estéreo originais, geradas pelo ODCI, e efetua a transformação 5.2. São fornecidas como saída, portanto, as imagens retificadas a serem utilizadas no cálculo do mapa de correspondências.

7.3 Correspondência manual das imagens estéreo

Conforme será mostrado na seção, o cálculo das coordenadas espaciais de um ponto pode ser efetuado de forma imediata desde que se conheçam os *pixels* correspondentes nas duas imagens geradas pela câmera calibrada. Para efeito de teste dos algoritmos de calibração e de reconstituição (seção 7.5) implementados, foi elaborado o programa *demoxyz.m*, que reconstitui as coordenadas x , y , z de um ponto indicado pelo usuário nas imagens estéreo.

Uma janela de operação deste programa é mostrada na figura 7.5. Além dos *pixels* correspondentes, o usuário deve fornecer as matrizes de calibração das câmeras geradoras de cada imagem, calculadas por *calib.m*. No exemplo, é efetuada a reconstituição de um dos pontos de referência do padrão; a comparação dos valores obtidos para cada um dos 32 pontos com os valores conhecidos utilizados na construção do padrão permite avaliar os algoritmos mencionados.

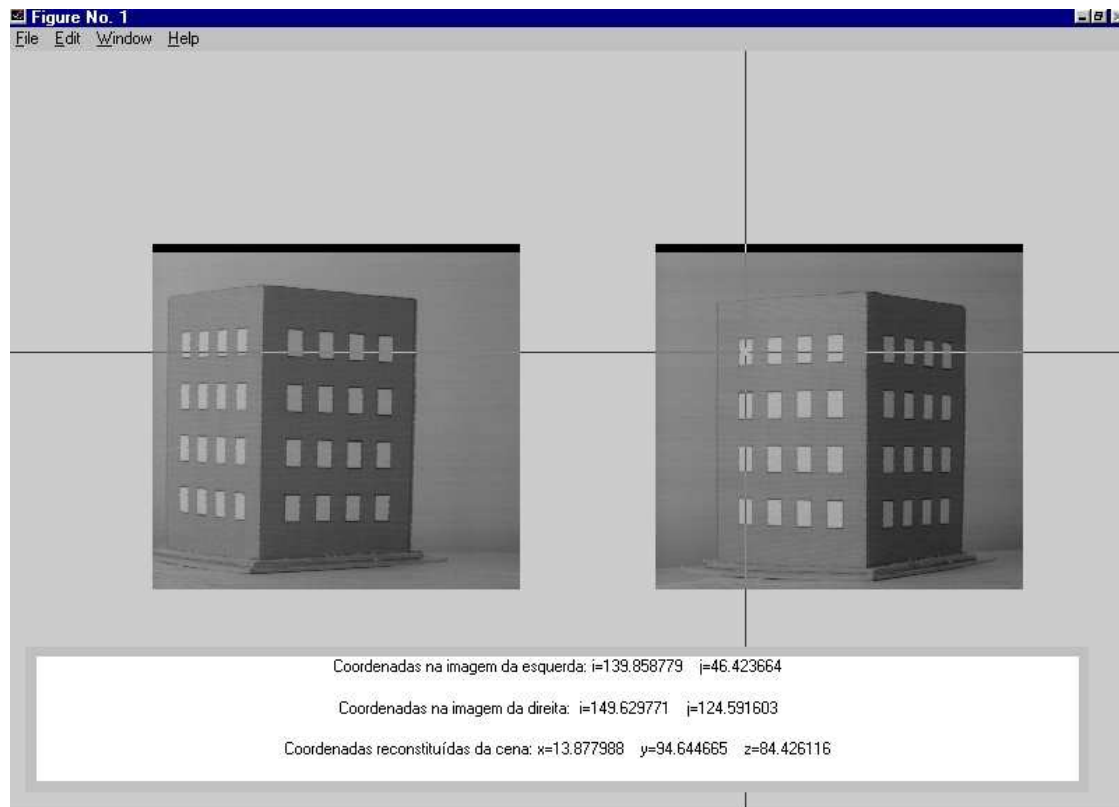


Figura 7.5: Janela de operação do programa *demoxyz.m*. No exemplo é efetuada a correspondência manual de pontos nas imagens não retificadas do padrão de calibração.

Observe-se que este programa permite ainda que se avalie se as matrizes retificadoras calculadas por *rectify.m* representam bem a transformação perspectiva realizada pelo sistema hipotético de duas câmeras em configuração canônica. Basta, neste caso, que se forneçam como entradas da função as imagens estéreo retificadas (fornecidas por *retifica.m*) e as matrizes retificadoras.

7.4 Correspondência automática das imagens estéreo

A técnica adotada para a correspondência automática das imagens estéreo consiste inicialmente num pré-processamento, destinado à detecção das bordas presentes, e na retificação das imagens binarizadas obtidas. Os pontos homólogos podem então ser determinados por uma varredura das linhas horizontais; como eles estão alinhados, graças à retificação, pode ser efetuada uma associação dos pontos presentes em linhas comuns nas duas imagens, de acordo com a seqüência das colunas. A figura 7.6 ilustra este processo.

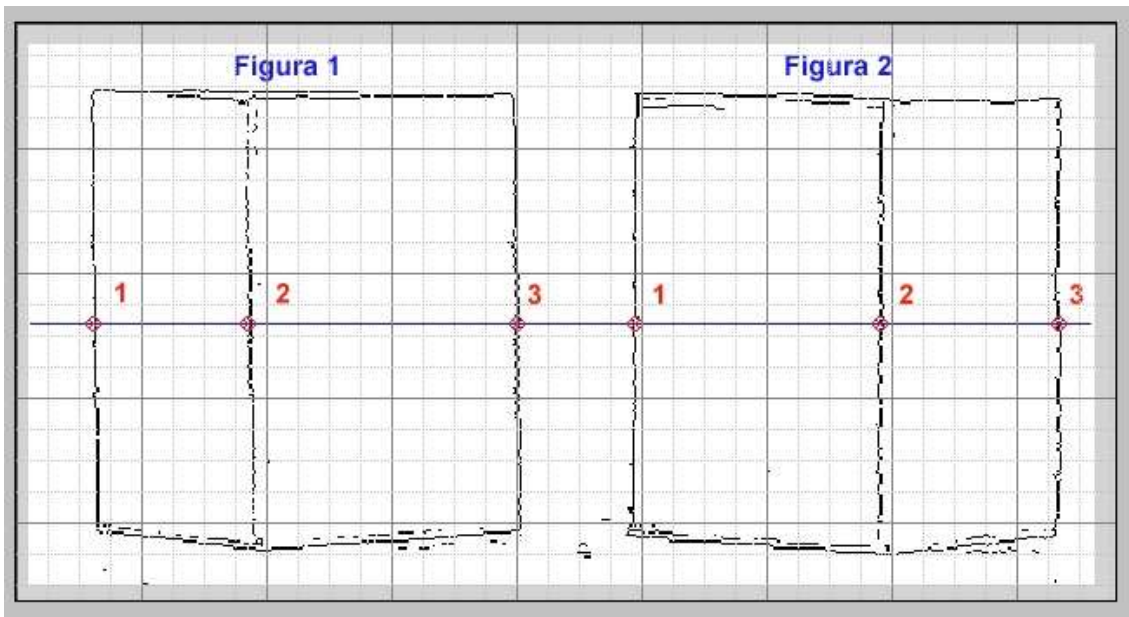


Figura 7.6: Correspondência automática de imagens estéreo retificadas.

Um problema deste procedimento surge quando o número de pontos presentes numa linha horizontal de um imagem não coincide com o da outra. Optou-se por descartar a linha correspondente nesta situação; conforme será mostrado junto aos resultados experimentais, ainda assim pôde ser efetuada a reconstituição de grande parte das pontos presentes nas arestas dos sólidos testados, sem que fosse afetada a estimativa da distância de cada um deles ao centro óptico da câmera nas duas posições adotadas.

O programa *corresp.m* implementa o algoritmo descrito. Ele recebe como entrada duas imagens estéreo binarizadas e retorna um mapa de correspondências esparso. A partir deste mapa, podem ser reconstituídos não somente pontos presentes nas bordas detectadas das imagens mas também pontos internos a elas; basta para tanto que sejam calculadas as retas que unem pontos de borda reconstituídos e que, em cada imagem retificada, se encontram na mesma horizontal. Por este procedimento, as linhas epipolares são reprojctadas no espaço tridimensional, levando à reconstituição de pontos não presentes no mapa de correspondências (conforme será mostrado nos resultados experimentais, utilizou-se neste projeto este procedimento).

Cabe ressaltar que a detecção das bordas das imagens, na fase de pré-processamento, levou a uma redução significativa do tempo de processamento, já que o número de pontos a serem analisados é cerca de 10% do total presente nas imagens fornecidas pelo ODCI. Na conclusão, é efetuada uma comparação dos tempos de processamento atingidos nesta pesquisa com aqueles apresentados em [7], em que se adotam mapas de profundidade densos calculados a partir de informação de cor em imagens RGB.

7.5 Reconstituição das imagens tridimensionais

Dado um elemento fotossensível \mathbf{f} no plano de retina de uma câmera qualquer, tudo o que se sabe acerca do objeto P correspondente filmado, cuja luz refletida o sensibiliza, é que ele está situado no raio óptico de \mathbf{f} , ou seja, na reta que une o foco da câmera àquele elemento. Conforme discutido no capítulo 2, isto significa que há infinitas soluções para o problema de determinar as coordenadas espaciais de P a partir de uma única imagem.

Se, no entanto, são conhecidas as imagens de P em duas imagens estéreo, sua posição pode ser determinada inequivocamente a partir da intersecção dos raios ópticos correspondentes às câmeras geradoras das duas imagens (observe-se, por exemplo, a figura 5.1).

Assim, sendo \mathbf{w} o vetor posição de P e $(u', v')^T$ e $(u'', v'')^T$ as coordenadas das imagens deste ponto em duas imagens estéreo, tem-se, do cálculo da intersecção dos raios ópticos [3]:

$$\mathbf{w} = \mathbf{A}^{-1} \cdot \mathbf{y} \quad (7.1)$$

onde:

$$A = \begin{pmatrix} (\mathbf{a}_1 - u' \mathbf{a}_3)^T \\ (\mathbf{a}_2 - v' \mathbf{a}_3)^T \\ (\mathbf{b}_1 - u'' \mathbf{b}_3)^T \\ (\mathbf{b}_2 - v'' \mathbf{b}_3)^T \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} -a_{14} + u' a_{34} \\ -a_{24} + v' a_{34} \\ -b_{14} + u'' b_{34} \\ -b_{24} + v'' b_{34} \end{pmatrix} \quad (7.2)$$

$$\tilde{\mathbf{P}}_{n1} = \left(\begin{array}{c|c} \mathbf{a}_1^T & a_{14} \\ \mathbf{a}_2^T & a_{24} \\ \mathbf{a}_3^T & a_{34} \end{array} \right) \text{ e } \tilde{\mathbf{P}}_{n2} = \left(\begin{array}{c|c} \mathbf{b}_1^T & b_{14} \\ \mathbf{b}_2^T & b_{24} \\ \mathbf{b}_3^T & b_{34} \end{array} \right), \quad (7.3)$$

O programa *reconsti.m*, implementado nesta pesquisa, efetua a reconstituição das coordenadas espaciais utilizando as equações anteriores e tendo como entrada o mapa de profundidade fornecido por *corresp.m*.

Capítulo 8

Resultados Experimentais

8.1 Resultados da calibração da câmera CCD

Para que pudessem ser efetuados os testes de validação do sistema estéreo implementado, foi inicialmente realizada a calibração da câmera CCD nas duas posições descritas no capítulo 7.

As imagens apresentadas na figura 5.4 foram utilizadas para a calibração; as coordenadas dos 32 *pixels* de referência, utilizadas como entrada do programa *calib.m*, foram calculadas com o auxílio do programa *centroid.m* e são apresentadas na tabela 5.1. As correspondentes coordenadas espaciais x , y , z de cada ponto de referência, também utilizadas em *calib.m*, foram determinadas durante a construção do padrão de calibração e são fornecidas na tabela 8.3.

Como resultado da calibração, foram obtidas as seguintes matrizes de projeção perspectiva, para as duas posições adotadas e considerando x , y , z dados em *mm*:

$$\tilde{\mathbf{P}}_1 = \begin{pmatrix} 2,443 & -1,3118 & 0,0453 & 137,811 \\ 0,2808 & 0,2401 & -4,0005 & 465,5910 \\ 8,265 \cdot 10^{-4} & 0,0011 & -1,3425 \cdot 10^{-4} & 1 \end{pmatrix}$$

$$\tilde{\mathbf{P}}_2 = \begin{pmatrix} 1,865 & -2,0576 & -0,0108 & 304,567 \\ 0,3342 & 0,2238 & -4,0165 & 475,5197 \\ 0,0011 & 8,1297 \cdot 10^{-4} & -1,2811 \cdot 10^{-4} & 1 \end{pmatrix}$$

Os parâmetros intrínsecos e extrínsecos das duas câmeras, calculados em *calib.m* a partir de $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$ com a utilização do algoritmo descrito em 4.2, são apresentados nas tabelas 8.1 e 8.2.

	Valor medido para a câmara na posição 1	Valor medido para a câmara na posição 2
Distância focal em número de pixels na horizontal ($f/\delta u$)	1953,98	1966,97
Distância focal em número de pixels na vertical ($f/\delta v$)	2808,45	2825,63
Coordenada horizontal do ponto principal (j_0)	526,28	547,00
Coordenada vertical do ponto principal (i_0)	268,88	227,09

Tabela 8.1: Parâmetros intrínsecos da câmara CCD obtidos durante a fase de calibração, nas duas posições adotadas.

	Valor medido para a câmara na posição 1			Valor medido para a câmara na posição 2		
	x (mm)	y (mm)	z (mm)	x (mm)	y (mm)	z (mm)
Eixo a	0,588	0,803	-0,0955	0,810	0,580	-0,0913
Eixo v	-0,0323	-0,0946	-0,995	-0,0704	-0,0586	-0,996
Eixo h	0,808	-0,588	0,0296	0,583	-0,813	0,00662
Foco (C)	-380,03	-600,84	54,03	-594,97	-391,52	46,98

Tabela 8.2: Parâmetros extrínsecos da câmara CCD obtidos durante a fase de calibração, nas duas posições adotadas.

Uma avaliação dos resultados da calibração será efetuada na seção 8.2 ; a comparação entre as coordenadas conhecidas dos pontos de referência do padrão com aquelas reconstituídas a partir das imagens estéreo permitirá determinar se as matrizes $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$ obtidas modelam bem a câmara em cada posição. Algumas observações, no entanto, podem ser efetuadas desde já:

1) Os parâmetros intrínsecos mantiveram-se aproximadamente constantes de uma posição da câmera para a outra, conforme esperado. Apenas a coordenada i_0 do ponto principal apresentou uma maior variação (da ordem de 15,54%), em virtude da imprecisão do padrão de calibração e de erros instrumentais.

2) As razões $f/\delta u$ e $f/\delta v$ apresentaram valores distintos entre si; isto significa que a câmera Pulnix utilizada apresenta resoluções distintas nos eixos horizontal e vertical ($\delta u \neq \delta v$). Este fato pode ser confirmado a partir da figura 8.1, em que as arestas verticais do cubo aparentemente são maiores do que as arestas horizontais, e pelos dados dos fabricantes: $\delta u = 8,6\mu$ e $\delta v = \mu$.

3) O eixo v medido coincidiu aproximadamente com o simétrico do eixo z , nas duas posições adotadas. O fato destes eixos terem apresentado a mesma direção deveu-se ao posicionamento da câmera, com sua base inferior paralela ao suporte do padrão de calibração; como o eixo v e o eixo z escolhido são, respectivamente, perpendiculares à base da câmera e ao suporte do padrão, isto implica que eles devam ser paralelos entre si. Já a orientação contrária destes eixos deveu-se à utilização, durante a fase de calibração, de imagens do padrão sem a usual inversão causada pelo modelo da câmera escura de orifício (ver figura 3.1), conforme destacado no capítulo 7. Isto fez com que o eixo v medido tivesse o sentido contrário ao da configuração real, em que as imagens são invertidas em relação aos objetos filmados³.

4) A distância do foco da câmera à origem do sistema de coordenadas adotado resultou em 712,99 mm para a posição 1 e em 713,78 mm para a posição 2 (valores calculados a partir das coordenadas x , y , z do foco listadas na tabela 8.2 usando a relação $d = \sqrt{x^2 + y^2 + z^2}$). Estes resultados se aproximam da distância do padrão de calibração aos dois pontos de apoio da câmera, na configuração descrita no capítulo 7.

5) As coordenadas z dos pontos focais correspondentes às duas posições adotadas diferiram em cerca de 13%. Esta diferença é atribuída a erros instrumentais e à imprecisão dos suportes utilizados para a fixação da câmera, que fez com que ela não ficasse na mesma altura nas duas situações consideradas.

³ Esta inversão do sentido do eixo v não constitui um erro experimental; apenas se optou, neste trabalho, por utilizar as imagens no formato em que são gravadas pelo aplicativo ODCI (ver capítulo [Metodologia]), sem refletir a inversão dos objetos causada pela câmera escura de orifício. As matrizes de calibração obtidas mapeiam corretamente as coordenadas espaciais dos objetos em *pixels* nestas imagens.

8.2 Reconstituição de objetos a partir da correspondência manual das imagens estéreo

Para a avaliação dos resultados da calibração da câmera, foi efetuada a reconstituição dos 32 pontos de referência do padrão a partir de suas imagens estéreo não retificadas (figura 5.4). Os pontos homólogos nas duas imagens foram, neste primeiro teste, determinados manualmente a partir do programa *demoxyz.m*. As coordenadas espaciais medidas são apresentadas na tabela 8.3, juntamente com os valores conhecidos determinados durante a construção do padrão (a numeração dos pontos é feita de acordo com a figura 4.1).

Observa-se nesta tabela que os valores de x , y , z fornecidos pelo programa *demoxyz.m* aproximam-se satisfatoriamente dos valores conhecidos; a maior discrepância observada foi de 3,30 mm (cerca de 6% do valor real), no caso da coordenada x do ponto 26. Esta diferença é aceitável, tendo em vista a imprecisão na determinação manual dos pontos correspondentes (ver figura 7.5).

O mesmo teste foi realizado utilizando ainda as imagens retificadas do padrão, geradas pelo programa *retifica.m*. O objetivo foi testar o algoritmo de retificação empregado, bem como avaliar se as matrizes de calibração retificadoras calculadas representam bem a transformação perspectiva realizada pelo sistema hipotético de duas câmeras em configuração canônica. Os resultados, apresentados na tabela 8.4, foram também satisfatórios, o que assegura que se possa utilizar o programa *retifica.m* para a correspondência automática das imagens estéreo, conforme o algoritmo discutido no apêndice B.

Ponto de referência	Coordenadas conhecidas			Coordenadas reconstituídas com o programa <i>demoxyz.m</i> a partir das imagens não retificadas do padrão		
	x (mm)	y (mm)	z (mm)	x (mm)	y (mm)	z (mm)
1	14	93	84	12.30	93.15	83.94
2	14	73	84	12.35	72.91	84.01
3	14	53	84	12.18	53.14	83.81
4	14	33	84	12.34	33.27	83.60
5	14	93	64	11.38	92.36	63.71
6	14	73	64	13.09	73.38	63.65
7	14	53	64	11.31	52.35	63.77
8	14	33	64	15.03	36.76	63.77
9	14	93	44	12.09	92.89	43.97
10	14	73	44	13.20	74.33	43.67
11	14	53	44	13.03	54.39	43.67
12	14	33	44	12.25	32.85	43.70
13	14	93	24	12.85	93.45	23.27
14	14	73	24	12.94	73.09	23.36
15	14	53	24	12.77	53.14	23.46
16	14	33	24	12.97	33.16	23.58
17	34	13	84	33.07	13.55	83.75
18	54	13	84	52.25	12.49	83.74
19	74	13	84	72.96	12.73	83.54
20	94	13	84	91.82	12.23	83.75
21	34	13	64	32.30	12.79	64.05
22	54	13	64	53.12	12.75	63.75
23	74	13	64	72.17	11.92	63.65
24	94	13	64	94.15	14.18	63.86
25	34	13	44	31.54	12.03	43.55
26	54	13	44	50.70	10.94	43.63
27	74	13	44	72.05	13.34	43.95
28	94	13	44	93.33	13.32	43.50
29	34	13	24	32.30	12.25	23.71
30	54	13	24	52.65	12.72	23.66
31	74	13	24	72.97	13.55	23.60
32	94	13	24	93.87	14.25	23.27

Tabela 8.3 : Resultado da reconstituição dos pontos de referência do padrão de calibração, a partir da correspondência manual de suas imagens estéreo não retificadas (a numeração dos pontos é feita de acordo com a figura 4.1).

Ponto de referência	Coordenadas conhecidas			Coordenadas reconstituídas com o programa <i>demoxyz.m</i> a partir das imagens retificadas do padrão		
	x (mm)	y (mm)	z (mm)	x (mm)	y (mm)	z (mm)
1	14	93	84	12,92	94,15	84,17
2	14	73	84	12,59	73,65	84,08
3	14	53	84	12,28	53,95	84,26
4	14	33	84	14,01	35,46	84,27
5	14	93	64	11,91	92,49	64,13
6	14	73	64	12,62	73,78	64,18
7	14	53	64	11,25	52,25	63,90
8	14	33	64	13,99	35,40	64,01
9	14	93	44	13,03	94,61	44,32
10	14	73	44	12,64	73,89	43,90
11	14	53	44	11,26	52,30	44,06
12	14	33	44	12,98	33,74	44,20
13	14	93	24	12,01	92,92	24,17
14	14	73	24	12,69	74,06	24,33
15	14	53	24	12,82	53,46	23,78
16	14	33	24	11,45	32,63	24,10
17	34	13	84	33,56	14,12	84,04
18	54	13	84	52,58	12,78	84,39
19	74	13	84	71,71	11,95	84,46
20	94	13	84	94,06	14,40	84,71
21	34	13	64	33,62	14,05	64,35
22	54	13	64	52,69	12,71	64,37
23	74	13	64	73,09	13,52	64,39
24	94	13	64	94,35	14,40	64,40
25	34	13	44	33,68	13,98	44,55
26	54	13	44	53,94	14,25	44,47
27	74	13	44	71,54	12,44	44,41
28	94	13	44	92,78	13,32	44,58
29	34	13	24	33,71	13,90	24,13
30	54	13	24	54,03	14,14	23,65
31	74	13	24	73,47	13,46	24,31
32	94	13	24	94,33	14,98	24,34

Tabela 8.4 : Resultado da reconstituição dos pontos de referência do padrão de calibração, a partir da correspondência manual de suas imagens estéreo retificadas (a numeração dos pontos é feita de acordo com a figura 4.1).

Em seguida, foi realizada a reconstituição dos vértices dos demais objetos construídos (cubo e prismas de bases pentagonal, hexagonal e octogonal), a partir da correspondência manual das imagens estéreo. Os resultados obtidos são apresentados no item B das figuras 8.1 a 8.4, em uma representação bidimensional dos eixos x , y , z e sobrepondo os vértices calculados (em verde) aos modelos conhecidos dos objetos (em vermelho).

Embora os objetos reconstituídos tenham se sobreposto bem aos modelos, cabe ressaltar que as imagens dos prismas de base pentagonal e de base octogonal mostraram-se deslocados em relação à posição real (os modelos foram desenhados de forma a compensar esta diferença); no primeiro caso, o deslocamento foi de 5 mm no eixo z e, no segundo, de 6 mm em z , de 10 mm em x e de 49 mm em y . Isto se deveu a um deslocamento constatado da câmera em relação à posição de calibração, durante a aquisição das imagens. Conforme destacado em Capítulo 7, não se dispunha de um dispositivo específico para fixação da câmera nas posições desejadas.

No caso do cubo e do prisma de base hexagonal, entretanto, não se observou este deslocamento.

8.3 Reconstituição de objetos a partir da correspondência automática das imagens estéreo

Com o uso do algoritmo de correspondência proposto no apêndice B (programa *corresp.m*), foi efetuada a reconstituição das coordenadas espaciais de pontos do cubo e dos prismas de bases pentagonal, hexagonal e octogonal. As figuras 8.1 a 8.4 mostram as imagens estéreo adquiridas, suas bordas detectadas e uma representação bidimensional dos eixos x , y , z com os pontos reconstituídos.

Observa-se, nestas figuras, que estes pontos, em azul, sobrepõem-se bem aos modelos conhecidos de cada objeto, em vermelho. Nota-se também que as superfícies dos modelos não são inteiramente cobertas pelos pontos reconstituídos. Isto se deve ao fato de que, no algoritmo proposto, são efetuadas as correspondências apenas quando as linhas epipolares (horizontais) interceptam as bordas das duas imagens retificadas num mesmo número de pontos (ver seção 7.4). Os pontos descartados neste processo correspondem às regiões dos modelos não cobertas pelas imagens reconstituídas.

A adoção de um mapa de profundidade esparso não deve prejudicar, no entanto, a estimativa das distâncias dos objetos filmados e, assim, a navegação de robôs com base nos pontos reconstituídos por este algoritmo. A próxima seção trata das distâncias a cada objeto estimadas a partir destes pontos.

Assim como ocorrido com os objetos reconstituídos a partir da correspondência manual das imagens estéreo, também se observou o deslocamento dos prismas de base pentagonal e octogonal reconstituídos em relação à posição real, pelas mesmas razões já apontadas.

8.4 Estimativa das distâncias dos objetos filmados

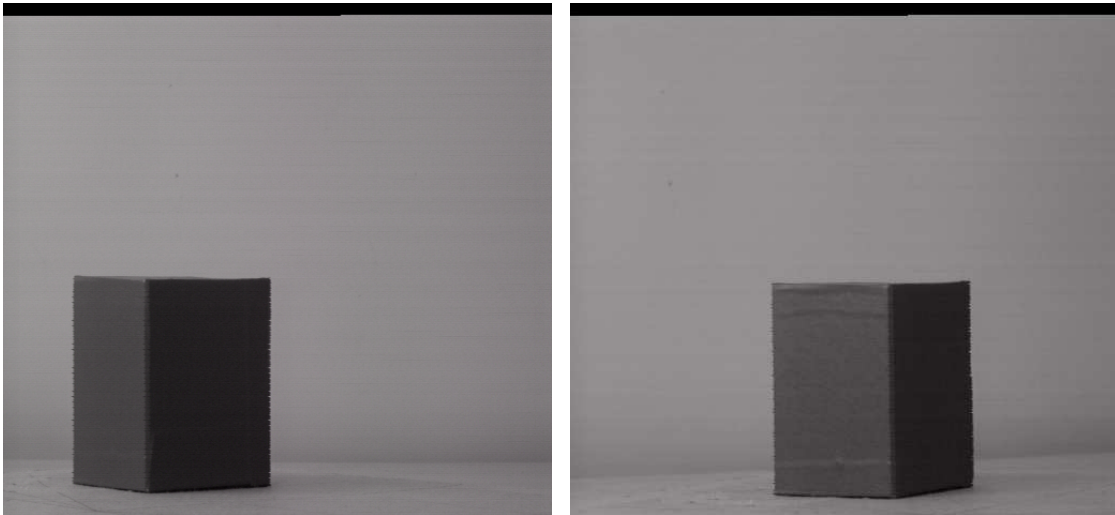
As distâncias dos objetos filmados em relação aos focos da câmera nas duas posições adotadas foram estimadas a partir das coordenadas espaciais de seus pontos reconstituídos, com o uso do programa *distcam.m*. A tabela resume estas distâncias.

Com exceção do caso do prisma de base octogonal, os resultados foram próximos do esperado (720 mm para a câmera na posição 1 e 730 mm para a posição 2), de acordo com o posicionamento das câmeras apresentado no capítulo 7. O maior erro verificado nos 4 primeiros casos foi de 5% do valor real. Este erro é aceitável, tendo em vista que não se dispunha de um dispositivo que garantisse um posicionamento da câmera idêntico ao usado durante a fase de calibração.

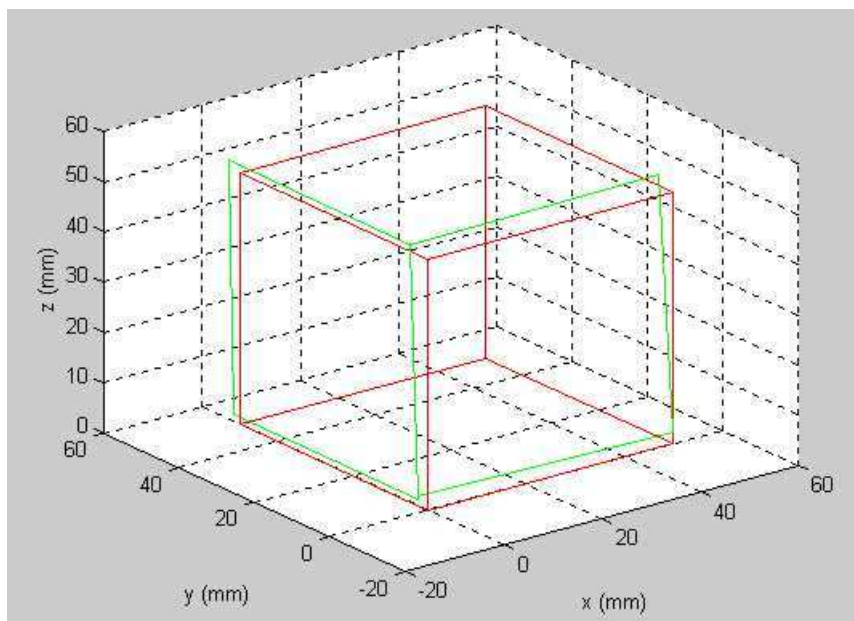
Quanto ao caso do prisma de base octogonal, o que se verificou foi que a distâncias calculadas foram cerca de 80 mm menores que os valores previstos. Este erro foi atribuído a um deslocamento da mesa de suporte da câmera durante a aquisição das imagens correspondentes. Este fato também explica o deslocamento dos objetos reconstituídos em relação às posições reais, conforme relatado nas seções 8.2 e 8.3.

Objeto	Distância ao foco 1 (mm)	Distância ao foco 2 (mm)
Padrão de calibração	712,9949	713,7795
Cubo	710,6759	710,6011
Prisma de base pentagonal	700,3547	697,8286
Prisma de base hexagonal	702,6882	699,3268
Prisma de base octogonal	626,7602	643,8842

Tabela 8.5: Distâncias calculadas dos objetos aos focos da câmera nas duas posições consideradas.



(a)



(b)

Figura 8.1 : Reconstituição das coordenadas espaciais de um cubo, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho.

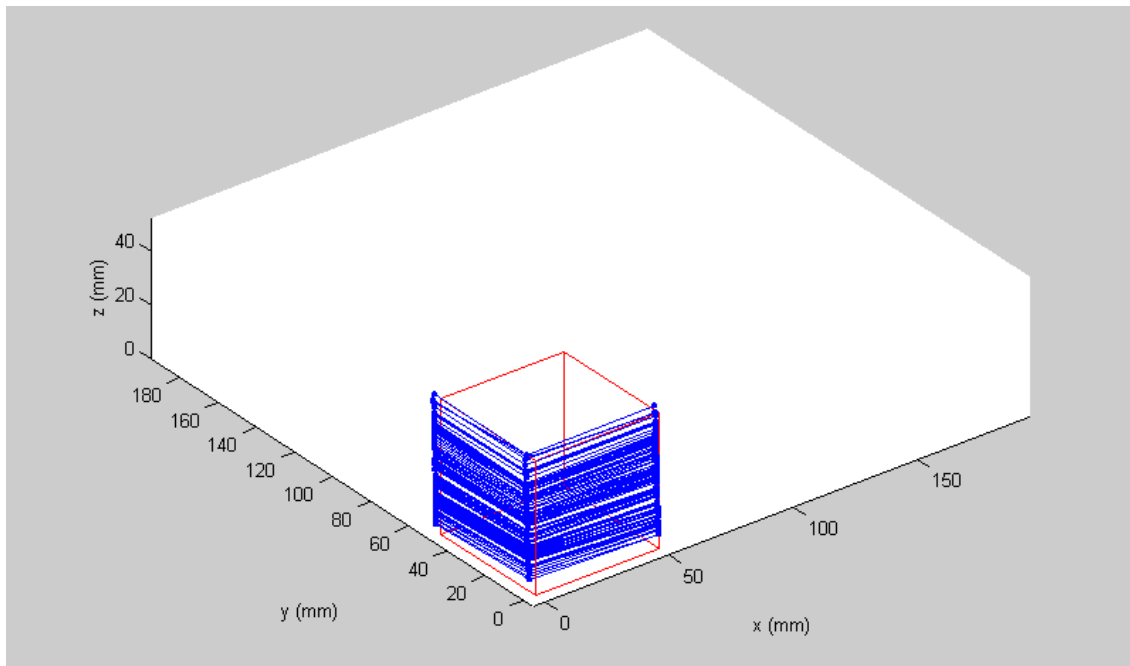
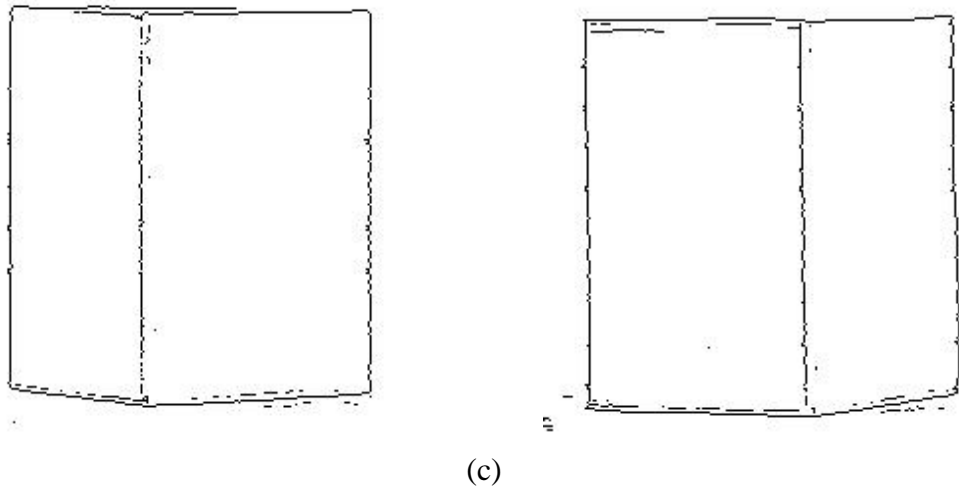
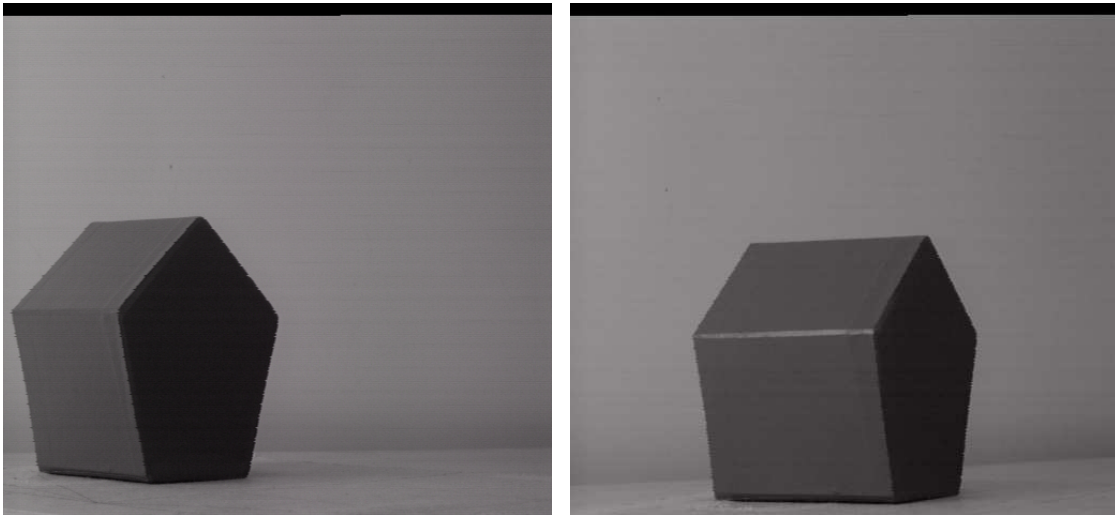
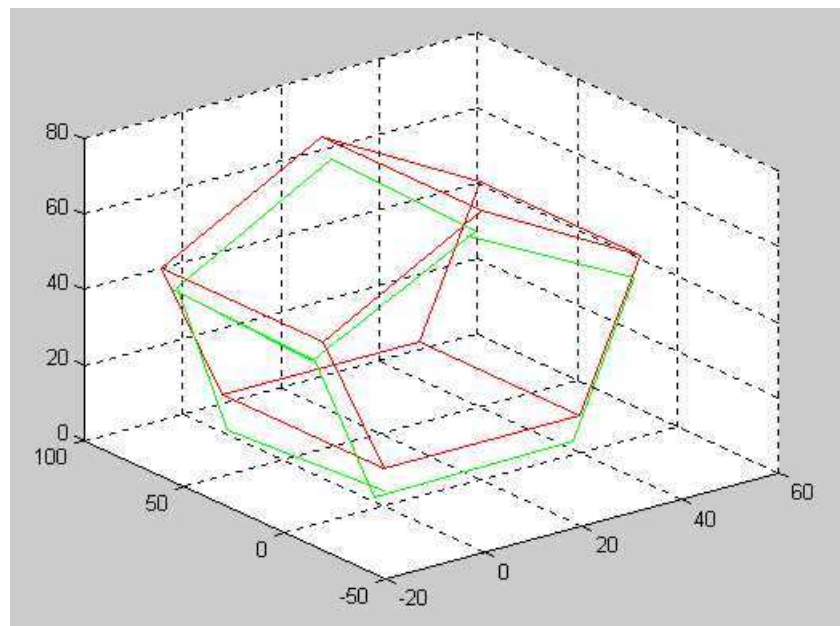


Figura 8.1 : Reconstituição das coordenadas espaciais de um cubo, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho.



(a)



(b)

Figura 8.2 : Reconstituição das coordenadas espaciais de um prisma de base pentagonal, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho.

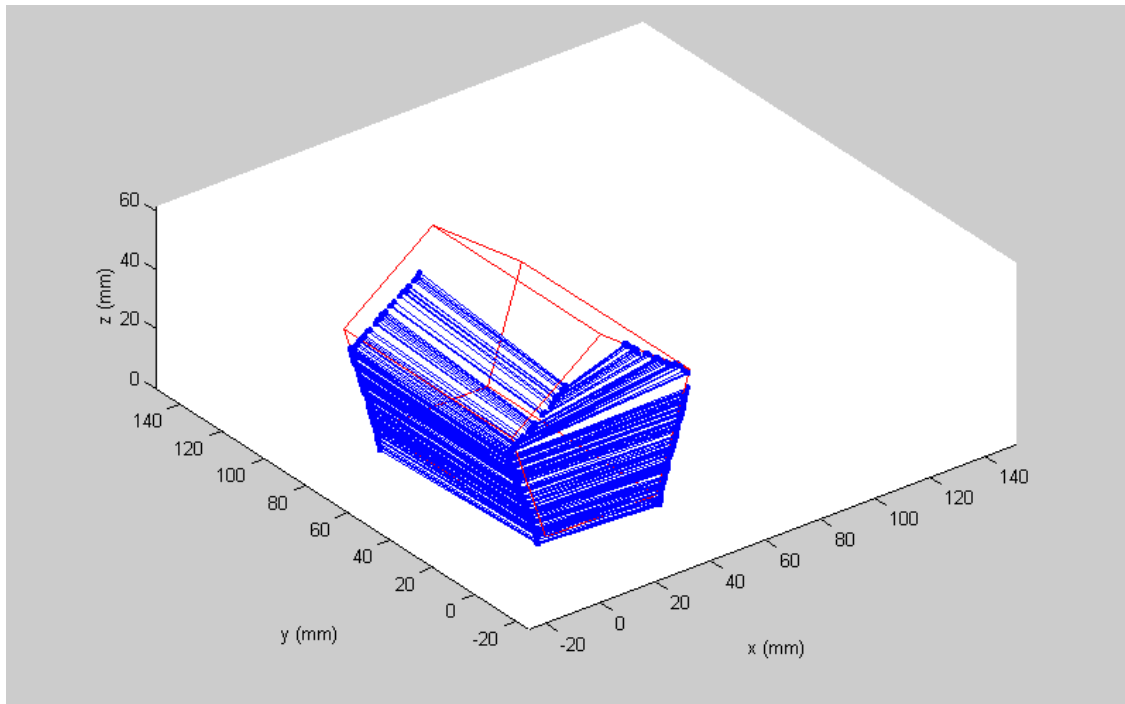
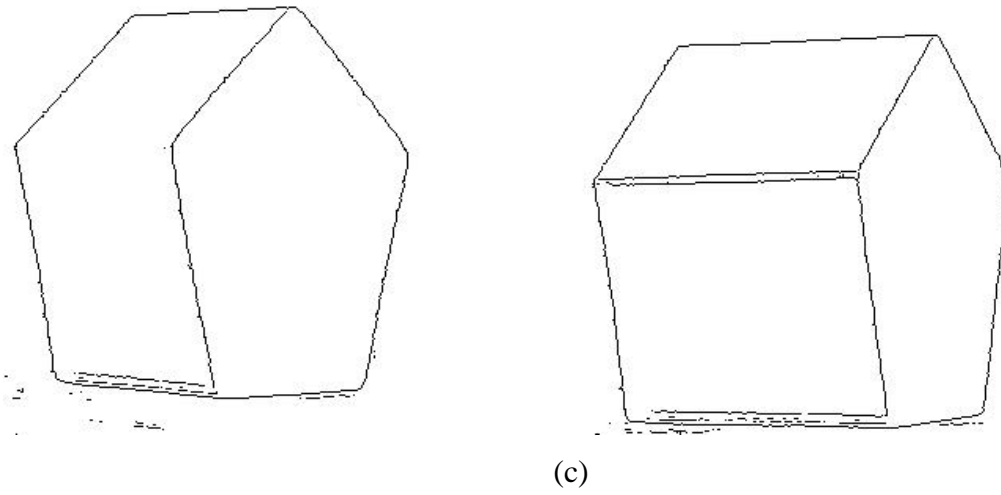
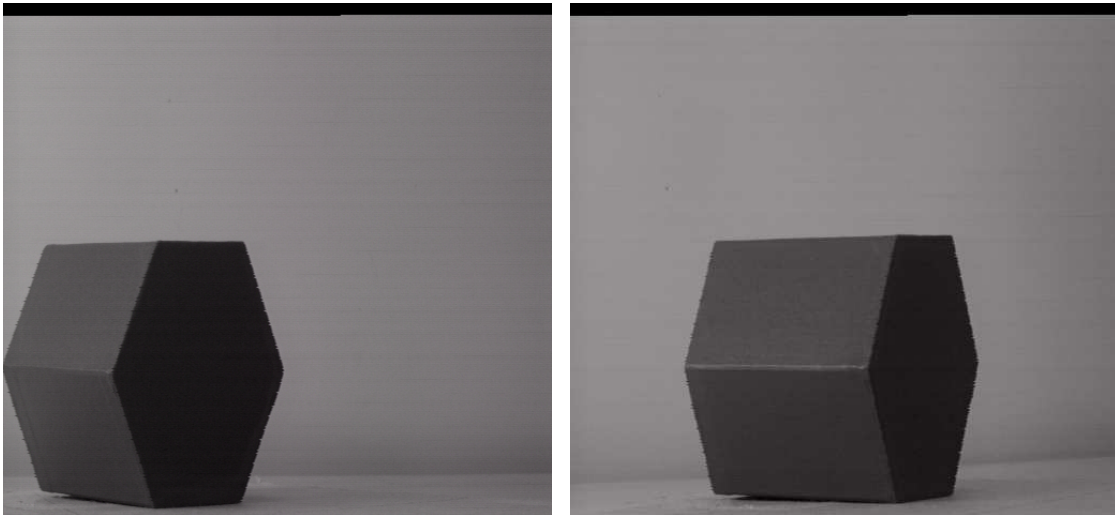
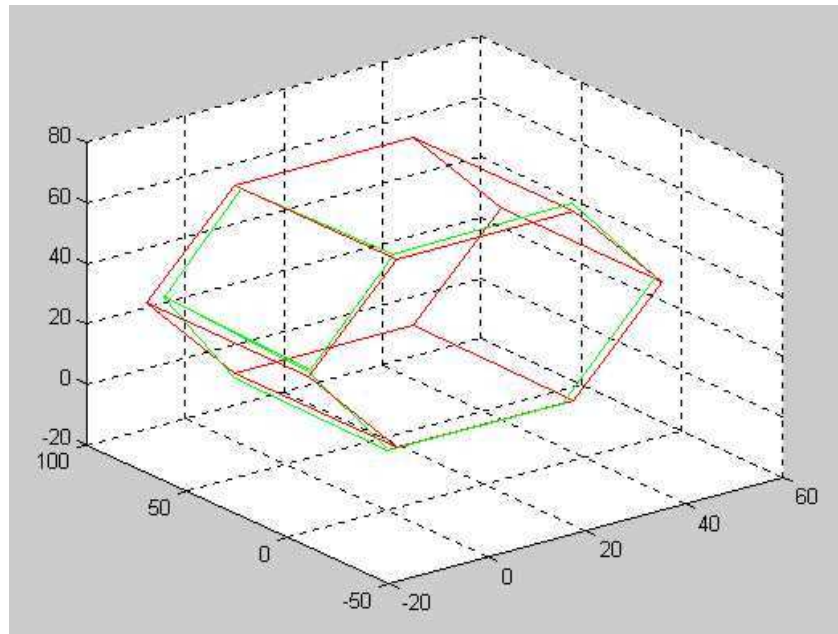


Figura 8.2 : Reconstituição das coordenadas espaciais de um prisma de base pentagonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho.

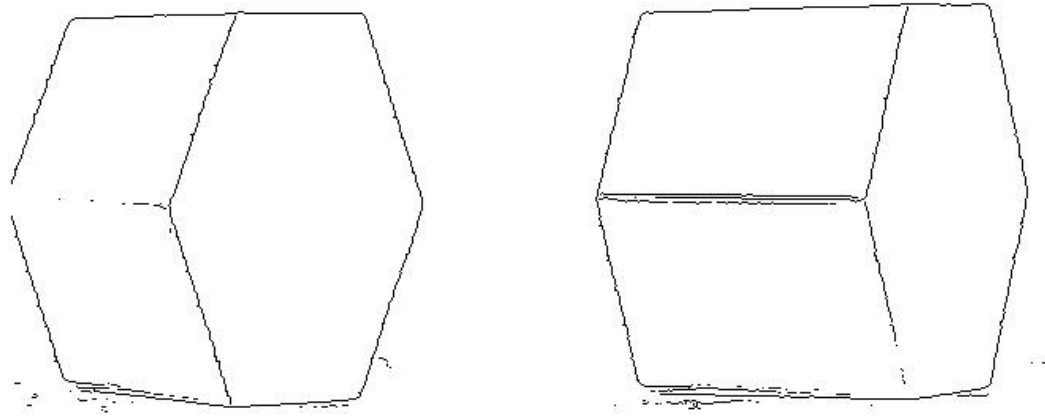


(a)

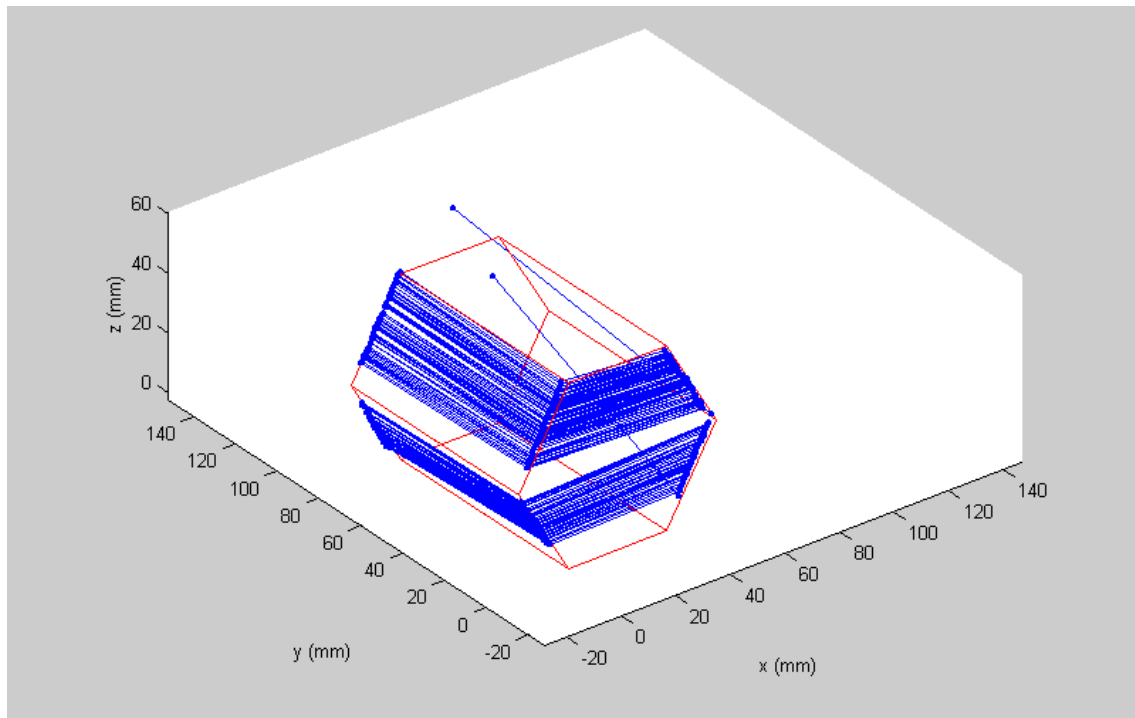


(b)

Figura 8.3: Reconstituição das coordenadas espaciais de um prisma de base hexagonal, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho.

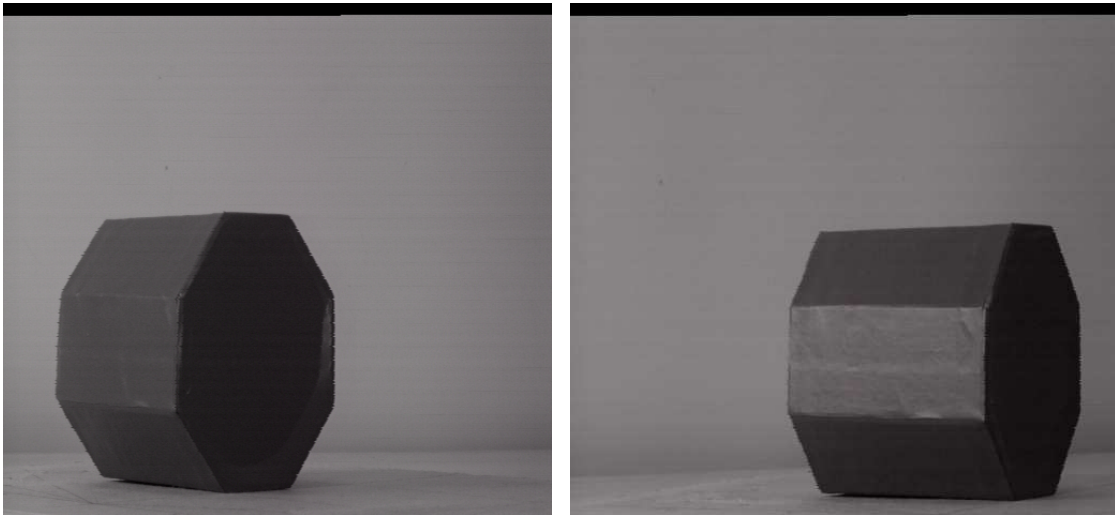


(c)

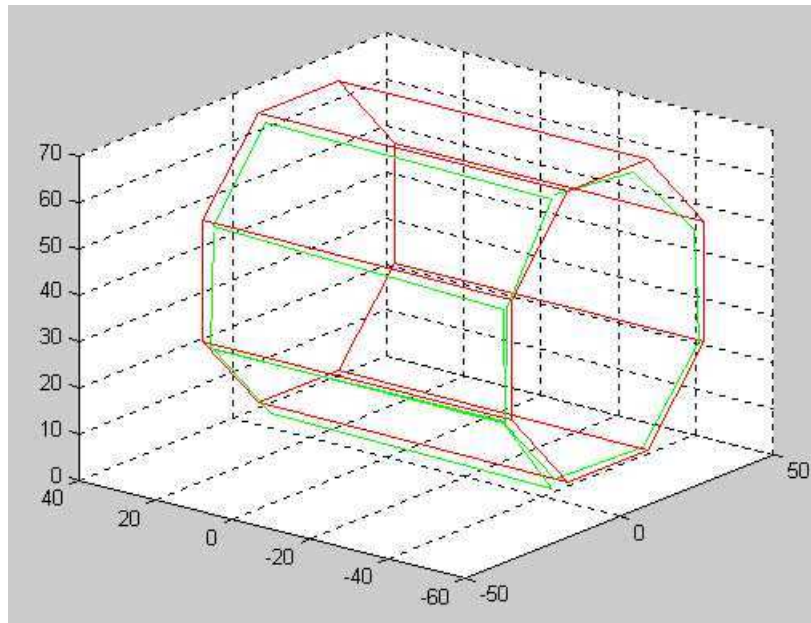


(d)

Figura 8.3: Reconstituição das coordenadas espaciais de um prisma de base hexagonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho.

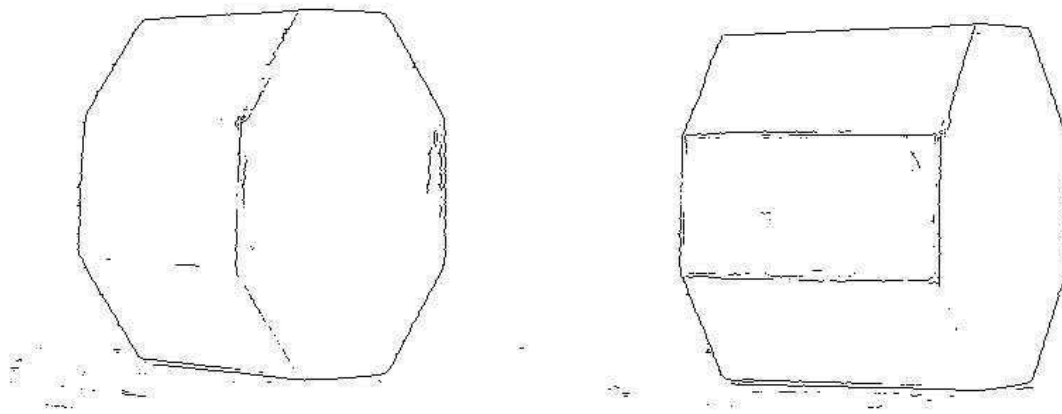


(a)

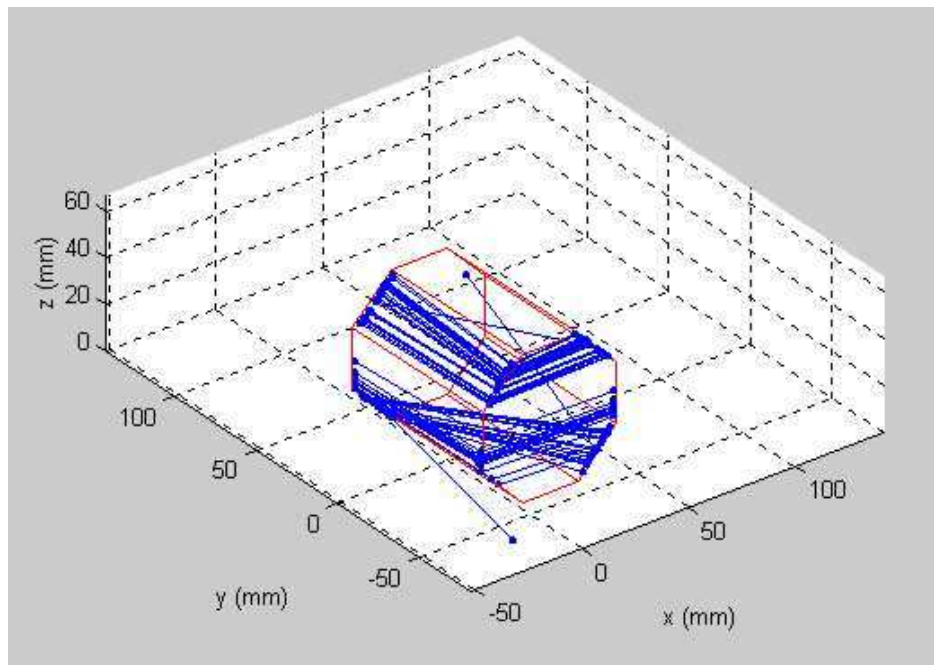


(b)

Figura 8.4: Reconstituição das coordenadas espaciais de um prisma de base octogonal, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho.



(c)



(d)

Figura 8.4: Reconstituição das coordenadas espaciais de um prisma de base octogonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho.

Capítulo 9

Conclusão

Nesta pesquisa, foi desenvolvido um sistema de visão computacional que efetua a reconstituição da estrutura tridimensional de objetos a partir de imagens estéreo. Na abordagem adotada, é inicialmente realizada a aquisição de imagens de um objeto por uma câmera CCD calibrada colocada em duas posições distintas. Em seguida, efetua-se um pré-processamento que consiste na detecção das bordas presentes; a retificação das imagens binarizadas obtidas permite a determinação de um mapa de correspondências esparso e, assim, o cálculo das três coordenadas espaciais das arestas do objeto.

Os resultados obtidos com esta técnica mostraram-se satisfatórios. Observou-se que os pontos reconstituídos de 4 prismas construídos para teste sobrepuseram-se bem aos modelos conhecidos das arestas destes sólidos. A distância dos objetos à câmera, em cada posição, pôde assim ser estimada com erros inferiores a 5% na maioria dos casos, o que se mostra suficiente para a navegação de robôs. Em outro caso, um deslocamento da câmera em relação às posições para as quais ela foi calibrada, ocorrido durante a movimentação da mesma para a aquisição das duas imagens, resultou em erros da ordem de 12%.

Cabe ressaltar que este deslocamento da câmera em relação às posições de calibração só representou um problema porque seu valor era desconhecido, já que se tratou de um erro experimental. De fato, no caso da navegação de robôs, é imprescindível que as câmeras sejam movimentadas; conhecendo-se cada deslocamento, no entanto, podem ser calculadas as novas matrizes de projeção perspectiva a partir da equação 3.26 e dos parâmetros intrínsecos e extrínsecos determinados durante a calibração, o que elimina o erro verificado.

Uma forma de melhor validar os algoritmos implementados consiste em utilizar duas câmeras fixas, o que evita que haja deslocamento em relação às posições de calibração durante a aquisição das duas imagens de um mesmo objeto. Outra alternativa é a instalação da câmera em um braço de robô de precisão, cujos deslocamentos possam ser bem determinados permitindo assim o cálculo das novas matrizes de projeção perspectiva.

Outro aspecto relevante da técnica adotada diz respeito à redução do esforço computacional a partir da realização da correspondência apenas de pontos estratégicos nas duas imagens, determinados durante o pré-processamento. Observa-se em [7], por exemplo, que o cálculo de mapas de profundidade densos utilizando informação de cor requer, em uma arquitetura Sun SPARC 10-40, 12,85 s para imagens em resolução 256 x 256 *pixels* e 116,61 s para imagens em resolução PAL (752 x 566 *pixels*); a alternativa proposta em [7] é a adoção de processadores em paralelo, que permitiu reduzir estes tempos respectivamente para 0,48s e 3,98s. Neste trabalho, a adoção de mapas de profundidade esparsos, por meio da detecção de bordas, permitiu que todo o processo de reconstituição utilizando imagens 480 x 512 *pixels* fosse realizado em cerca de 3s num microcomputador PC 300 MHZ com 128MB de memória RAM (a correspondência das imagens em si pôde ser efetuada em menos de 1s).

Uma desvantagem desta abordagem, no entanto, é que ela assume que os objetos filmados possam ser bem representados pelas bordas calculadas durante a fase de pré-processamento. Assim, embora tenha se mostrado viável para o caso dos prismas reconstituídos, bem limitados por suas arestas, ela não pode ser empregada no caso de imagens mais complexas, de objetos irregulares.

Como sugestão para trabalhos futuros, há a possibilidade de se instalar duas câmeras em um pequeno robô conectado a um PC por um enlace de microondas. A navegação deste robô em um ambiente de laboratório, cercado de objetos poliédricos, pode então ser feita utilizando as coordenadas tridimensionais reconstituídas a partir de imagens estéreo enviadas periodicamente ao PC pelo robô. Conforme já ressaltado, a eficiência deste método irá depender da precisão do posicionamento do robô, já que terá de ser efetuado um novo cálculo das matrizes de projeção perspectiva sempre que houver um deslocamento das câmeras.

Referências Bibliográficas

- [1] CALÇADA, C. S. E SAMPAIO, J. L., *Física Clássica – Óptica e Ondas*, Atual Editora LTDA., 1991, São Paulo.
- [2] FAUGERAS, O., *Three-Dimensional Computer Vision – A Geometric Viewpoint*, chapter 3, The MIT Press, Massachusetts, 1996.
- [3] FUSIELLO, A., TRUCCO, E. AND VERRI, A., “Rectification with unconstrained stereo geometry”, 1998.
- [4] GEIGER, D., “Inteligência Artificial: Além dos Limites do Homem”, Revista Galileu, número 88, páginas 34 a 40, Globo, São Paulo, 1998
- [5] GREWE, L. L. AND KAK, A. C., “Stereo Vision”, Robot Vision Laboratory, School of Electrical Engineering, Purdue University, West Lafayette, Indiana.
- [6] KOSCHAN, A., “A Comparative Study On Color Edge Detection”, Institute for Technical Informatics, FR 3-11, Technical University Berlin, Franklinstr. 28-29.
- [7] KOSCHAN, A. AND RODEHORST, V., “Towards Real-Time Stereo Employing Parallel Algorithms For Edge-Based And Dense Stereo Matching”, Proc. of the IEEE Workshop on Computer Architectures for Machine Perception CAMP’95, 18-20 September 1995, Como, Italy.

-
- [8] LAVEAU, S. AND FAUGERAS, O., “3-D Scene Representation as a Collection of Images and Fundamental Matrices”, INRIA – Institut National de Recherche en Informatique et Automatique, N° 2205, Février 1994.
- [9] MCKERROW, P. J., *Introduction to Robotics*, chapter 10, Addison-Wesley Publishers, 1995.
- [10] SCHERER, S. AND WERTH, P., “Adaptive Local Description Matching”, to appear in Proceedings 22nd Workshop of AAPR 1998, Illmitz, Austria, April 29, 1998.
- [11] SHOMAR, W. J. AND YOUNG, T. Y., “Three-Dimensional Shape Recovery from Line Drawings”, Department of Electrical and Computer Engineering, University of Miami, Coral Gables, Florida.

Apêndice A

Sensor

Qualquer ser vivo, seja qual for o seu tamanho e complexidade, precisa obter informações sobre o meio em que vive de modo a garantir a sua sobrevivência. Estas informações são utilizadas na obtenção de alimentos, manutenção ou alteração da taxa de metabolismo do organismo e decisão sobre o momento de reprodução. Estas informações permitem que os seres vivos sejam autônomos e não necessitem de outro “ser” que tomem estas decisões por eles.

Dentre os seres vivos, os animais pluricelulares mais evoluídos (insetos, répteis, aves e mamíferos) apresentam a capacidade de se locomoverem dentro do seu meio ambiente e tomarem decisões sobre o modo de interação com ele. Por apresentarem uma maior autonomia em relação ao meio, estes animais necessitam de uma maior quantidade de informações sobre o mesmo, que são fornecidas pelos órgãos de sentido (tato, paladar, olfato, audição e visão).

Desde o início de sua existência o Homem têm criado ferramentas que o auxiliem na execução de suas tarefas. Com o passar dos séculos, estas ferramentas têm sido

aprimoradas de modo a torná-las autônomas, ou seja independentes do homem para executar determinada tarefa. Esta independência se refere a capacidade de tomar decisões inerentes ao processo da tarefa e obter energia, gasta na realização, de fontes que não utilizem energia muscular humana.

Para pequenas tarefas, como moer milho, o homem há alguns séculos conseguiu construir máquinas independentes do esforço humano. Porém, máquinas com a capacidade de tomar decisões capazes de realizar grandes tarefas, como explorar a cratera de um vulcão, só começaram a ser construídas nos meados do século XX. Surgindo, assim, os primeiros robôs, mecanismos autônomos capazes de realizar tarefas complexas sem a intervenção humana.

Para conseguir a desejada autonomia, estes mecanismos, como os seres vivos, necessitam obter informações sobre o meio ambiente em que se encontram. Estas informações são obtidas através de sensores que realizam a mesma função dos órgãos dos sentidos dos seres vivos. Os sensores são capazes de fornecer aos robôs o sentido do “tato”- detetores de pressão e calor – “olfato”, “paladar”- detecção de substâncias químicas existentes no meio – “audição”- reconhecimento de voz e sonar – e “visão”- reconhecimento de padrões e navegação. Sendo o último o assunto deste trabalho.

A visão fornece uma quantidade razoável de informações sobre o meio:

- Luminosidade;
- Presença, distância, forma e tamanho de corpos existentes no meio; e
- Reconhecimento e distinção de um corpo em especial dentre um conjunto de corpos.

Os seres vivos utilizam apenas um tipo de “sensor” – olhos – para obtenção de todas estas informações. Os robôs, ao contrário, podem utilizar um ou mais tipos diferentes de sensores para obtenção destes dados.

A câmera de vídeo é um sensor capaz de fornecer a maioria destas informações quando é utilizada apenas uma e todas quando utilizado um sistema estéreo (com duas ou mais câmeras). Chama-se a atenção para o fato de que a câmera não fornece estas informações prontas, como num termômetro, sendo necessário um processamento. O uso da visão estéreo para obtenção das informações sobre presença, distância, forma e tamanho de corpos existentes no meio é o escopo dentro da visão computacional que é abordado neste texto.

A seguir será feita uma breve descrição sobre sensores.

A.1 Sensores na Robótica

O sucesso na aplicação de um robô numa determinada tarefa depende exclusivamente da sua capacidade de medir parâmetros internos do seu sistema e externos referentes ao ambiente. Esta capacidade é determinada pela quantidade e diversidade de sensores existentes no robô.

Em robótica, os sensores são usados para:

- Medir variáveis de controle;
- Localização de objetos;
- Evitar situações de falha;
- Evitar colisões;
- Monitorização da interação com o ambiente;
- Monitorização das mudanças no ambiente que interfiram no processo; e
- Inspeção dos resultados do processo.

Ainda neste campo, os sensores são classificados em dois tipos: sensores internos e sensores externos. Os primeiros são usados para medir parâmetros do robô que possam afetar o bom funcionamento do seu sistema. O segundo são usados para medir características do sistema e a posição relativa do robô nele.

A.2 Sensores Internos

Parâmetros internos que podem ser medidos são: posição, movimento e esforço de todas as partes do robô (braços, articulações, etc.), temperatura e umidade interna. O conhecimento destes parâmetros permite evitar estragos em qualquer parte do robô que o parariam temporariamente ou permanentemente. As medidas de posição e movimento são relativas a um “frame” e exemplificadas nos próximos parágrafos.

As medidas de posição e movimento, neste caso, são referentes ao movimento de partes do robô, como um braço, e não movimento dele como um todo no ambiente.

Os robôs são modelados em dois espaços: junta e cartesiano. A posição, velocidade e aceleração podem ser medidos em ambos os espaços. O espaço cartesiano descreve o movimento total do sistema, por exemplo a posição do “punho” de um braço mecânico,

enquanto o espaço de juntas descreve se preocupa com o movimento de cada articulação individualmente.

Um exemplo de **sensor de medida de posição** no espaço cartesiano foi desenvolvido por Mactarlane e Donath(1984) [10], figura A.1. O método rastreia o punho do braço de um robô utilizando um scanner a laser. Uma fina cortina de luz é produzida por lentes cilíndricas que focam o laser num espelho plano rotativo que varre o feixe através do volume de interesse. Este feixe de luz é detectado por fotodiodos colocados em posições fixas, que funcionam como referencia, e nas articulações do sistema. A velocidade de varredura é fixa e função da velocidade de rotação do espelho. A diferença de tempo existente entre a detecção do feixe pelo diodo fixo e pelo diodo da articulação é proporcional a distância entre eles no sentido da varredura. O uso de vários lasers em direções diferentes permite obter a posição do punho do robô.

No espaço de juntas a medida de posição pode ser feita utilizando um potenciômetro e de velocidade utilizando um tacômetro. A aceleração pode ser medida através de sistemas de balança de torque e medidores de pressão.

Os **sensores de força** existem em ambas classes de sensores, internos e externos, sendo a sua classificação dependente da utilização. Existem dois métodos comuns utilizados na confecção destes sensores: deflexão de um elemento elástico e variação da resistência de acordo com a variação da força aplicada.

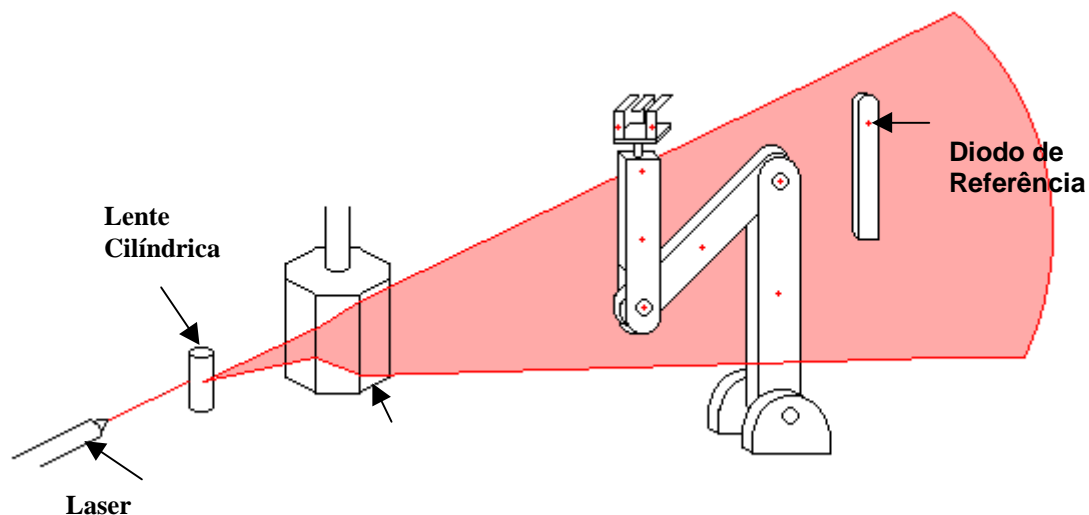


figura A.1 : Sensor de posição de Mactarlane e Donath

A.3 Sensores Externos

Sensores externos medem a interação do robô com o meio. Existe uma infinidade de tipos deste sensor que são usados para as mais variáveis funções como o tato, proximidade, velocidade, som, umidade, composição gasosa, pressão atmosférica, temperatura, etc. Uma observação válida é que a classificação de sensores como internos e externos refere-se ao uso do que a função ou o modelo de sensor.

Os **sensores de tato** são usados para dois propósitos: detectar a presença de toque e detectar as características do objeto tocado. O primeiro é usado quando se deseja saber que parte do robô está tocando algo e o segundo quando se deseja levantar características sobre o objeto que está sendo tocado.

Sensores de proximidade detectam um objeto perto do sensor sem toca-lo fisicamente. Dois princípios são utilizados na sua construção: modificação de um sinal emitido e distúrbio de alguma característica do ambiente.

A modificação do sinal emitido ocorre quando parte do sinal emitido é bloqueado ou refletido na direção de um receptor. **Sensores de proximidade ópticos** utilizam este princípio. Um exemplo é o sensor constituído de um emissor e receptor de luz como mostrado na figura A.2. Este sensor detecta a presença do objeto pela reflexão da do feixe de luz na direção do receptor que ocorre quando o objeto se encontra na região de sensibilidade do sensor.

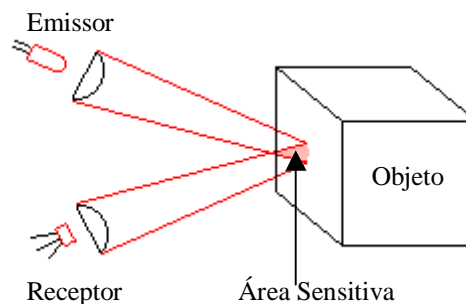


figura A.2 : Sensor de proximidade

Sensores de profundidade medem a distância existente entre o sensor e um objeto na cena e se baseia em dois princípios físicos: tempo de propagação de um pulso e triangulação.

O **sensor de tempo de propagação** emite um pulso e mede o tempo que leva desta emissão até o retorno do mesmo. A metade deste tempo de propagação multiplicado pela velocidade do pulso no meio de propagação fornece a distância existente entre o sensor e o objeto. O exemplo mais comum deste tipo de sensor é o radar.

Sensores de triangulação medem a distância pelo uso de princípios geométricos. A triangulação requer a detecção do objeto a partir de dois pontos de visão e o conhecimento da distância dos pontos em questão. Estes dois pontos pode ser obtidos movimentando um simples sensor ou pelo uso de vários sensores distribuídos no espaço. Um exemplo é o sistema em que um dos sensores emite um feixe de luz que é refletido e detectado pelo segundo sensor, quando isto ocorre, os ângulos de emissão e recepção do feixe de luz são armazenados, e a distância calculada. O sistema de visão estéreo, que é o objeto principal deste trabalho, é um sensor passivo de triangulação.

Outro tipo de sensor de profundidade é o **sonar** que se baseia no fenômeno de eco das ondas sonoras. O princípio é o mesmo utilizado nos sensores de tempo de vôo: o tempo de retorno é proporcional a distância. A diferença é que o sonar utiliza ondas sonoras ao invés de ondas eletromagnéticas.

Apêndice B

Listagens dos Programas

B. 1 Programa de Calibração da Câmera CCD

```
function c=centroid(I,pixels_h,pixels_v,criterio,limiar)

% c=centroid(I,pixels_h,pixels_v,criterio,limiar);
%
% Calcula os centróides de diferentes janelas de uma imagem. Esta é
% inicialmente exibida na tela, sendo que a posição de cada janela é
% definida pelo usuário com o auxílio do "mouse" (botão esquerdo). Os
% dois "pixels" que definem cada centróide podem ser armazenados em
% uma linha da matriz c, bastando que se pressione a tecla espaço
% quando desejado (um sinal sonoro indica que os valores foram
% armazenados). Após o cálculo do último centróide desejado, deve ser
% pressionada a tecla 'ESC'.
%
% I: imagem em níveis de cinza.
% pixels_h: número de pixels presentes numa linha horizontal de cada
%           janela cujo centróide deve ser calculado;
% pixels_v: número de pixels presentes numa linha vertical de cada
%           janela cujo centróide deve ser calculado;
% limiar: valor mínimo ou máximo de intensidade de um ponto em I
%         para que o mesmo seja incluído no cálculo dos centroides.
% criterio: vale '>' ou '<' e estabelece se o valor limiar é mínimo ou
%           máximo, respectivamente.
button=-1;
a_registrar=0;
armazenados=0;
a=round(pixels_h/2);
b=round(pixels_v/2);
som=(0:50000);
som=sin(2*pi*445*som/100000);
imshow(I,256);
while(button~=27)
    [x,y,button]=ginput(1);
    if(button==1)
        x=round(x);
```

```

y=round(y);
A=I(y-b:y+pixels_v-b-1,x-a:x+pixels_h-a-1);
if(critério(1,1)=='>')
    [i,j]=find(A>=limiar);
else
    [i,j]=find(A<=limiar);
end
i=sum(i)/size(i,1);
j=sum(j)/size(j,1);
i=i+y-b-1;
j=j+x-a+1;
imshow(I,256);
X=[x-a;x-a];
Y=[y-b;y+pixels_v-b-1];
line(X,Y);
X=[x+pixels_h-a-1;x+pixels_h-a-1];
Y=[y-b;y+pixels_v-b-1];
line(X,Y);
X=[x-a;x+pixels_h-a-1];
Y=[y-b;y-b];
line(X,Y);
X=[x-a;x+pixels_h-a-1];
Y=[y+pixels_v-b-1;y+pixels_v-b-1];
line(X,Y);
'centroide'
i,j
a_registrar=1;
else
    if((button==32)&(a_registrar==1))
        a_registrar=0;
        armazenados=armazenados+1;
        c(armazenados,1)=i;
        c(armazenados,2)=j;
        sound(som,100000);
    end
end
end
close(1);

```

```

function [T,p]=calib(xyz_cena,ij_imag)

%[T,p]=calib(xyz_cena,ij_imag);
%
% Calcula a matriz de transformação perspectiva T e os parâmetros p
% ( ver relação abaixo ) de uma câmera, devendo para tanto ser
% fornecidos:
% *as coordenadas x,y,z de pelo menos 6 pontos não coplanares de uma
% cena arbitrária ( coordenadas estas referenciadas a um sistema
% também arbitrário).Cada linha da matriz xyz_cena deve ser formada
% pelas coordenadas de um único desses pontos.
% *os índices i e j dos pixels correspondentes àqueles pontos na
% imagem gerada pela câmera. Na matriz ij_imag, cada linha l deve
% conter os índices associados ao ponto cujas coordenadas são os
% elementos da linha l de xyz_cena.
%
% A função calib então retorna:
% *T: matriz de calibração ou de transformação perspectiva (3x4), com
% os elementos normalizados para que se tenha T(3,4)=1.
% *p: matriz-coluna que contém os parâmetros da câmera, na seguinte

```

```

% ordem: f/du, f/dv, hx, hy, hz, ax, ay, az, vx, vy, vz, Cx, Cy, Cz,
% i0,j0.
%
if((~(size(xyz_cena,1)==size(ij_imag,1)))|(~(size(xyz_cena,2)==3))|(~(
size(ij_imag,2)==2)))
    'Entradas invalidas.'
else
    %Cálculo de T
    %Determinacao da matriz U que minimiza os elementos de
    %E=(A*U-B)'*(A*U-B)
    %Determinacao de A e de B
    n=size(xyz_cena,1);
    k=1;
    while(k<=n)
        A(2*k-1,1:3)=xyz_cena(k,1:3);
        A(2*k-1,4)=1;
        A(2*k-1,5:8)=zeros(1,4);
        A(2*k-1,9:11)=xyz_cena(k,1:3)*(ij_imag(k,2))*(-1);
        A(2*k,1:4)=zeros(1,4);
        A(2*k,5:7)=xyz_cena(k,1:3);
        A(2*k,8)=1;
        A(2*k,9:11)=xyz_cena(k,1:3)*(ij_imag(k,1))*(-1);
        B(2*k-1,1)=ij_imag(k,2);
        B(2*k,1)=ij_imag(k,1);
        k=k+1;
    end
    %Cálculo de U
    U=((inv((A')*A))*(A'))*B;
    %Determinacao de T
    T(1,1:4)=(U(1:4,1))';
    T(2,1:4)=(U(5:8,1))';
    T(3,1:3)=(U(9:11,1))';
    T(3,4)=1;

    %Determinacao dos parametros da camera (pelo algoritmo apresentado
    %por Ganapathy)
    %1) Determina-se (a*C)^2 a partir de T3*T3.
    V1=T(3,1:3);
    aC2=1/(V1*V1');

    %2) Determina-se j0 a partir de T1*T3.
    V2=T(1,1:3);
    j0=aC2*(V1*V2');

    %3) Determina-se i0 a partir de T2*T3.
    V2=T(2,1:3);
    i0=aC2*(V1*V2');

    %4) Determinam-se ax,ay,az a partir de T(3,1), T(3,2), T(3,3) (a
    %suposicao de que A*C<0 e' explicada no item 7).
    ax=T(3,1)*sqrt(aC2);
    ay=T(3,2)*sqrt(aC2);
    az=T(3,3)*sqrt(aC2);

    %5) Determina-se o modulo de f/du a partir de T1*T1.
    V1=T(1,1:3);
    fdu=sqrt(V1*(V1')*aC2-j0^2);

    %6) Determina-se o modulo de f/dv a partir de T2*T2.

```

```

        fdv=sqrt(V2*(V2')*aC2-i0^2);

%7) Adotam-se C*a<0 e f/du>0 (o que equivale a arbitrar as
%direcoes de a e de u). Aceita a convencao, supoe-se f/dv>0 e
%calculam-se hx,hy,hz,vx,vy,vz a partir de T(1,1), T(1,2),
%T(1,3), T(2,1), T(2,2), T(2,3).
        hx=(T(1,1)*sqrt(aC2)-j0*ax)/fdv;
        hy=(T(1,2)*sqrt(aC2)-j0*ay)/fdv;
        hz=(T(1,3)*sqrt(aC2)-j0*az)/fdv;
        vx=(T(2,1)*sqrt(aC2)-i0*ax)/fdv;
        vy=(T(2,2)*sqrt(aC2)-i0*ay)/fdv;
        vz=(T(2,3)*sqrt(aC2)-i0*az)/fdv;

        %Em seguida, efetua-se um segundo cálculo de vx,vy,vz
        %partindo-se de v=a*h (produto vetorial).
        vx2=ay*hz-az*hy;
        vy2=az*hx-ax*hz;
        vz2=ax*hy-ay*hx;

        %Determina-se entao o angulo entre os valores experimentais
        %[vx vy vz] e [vx2 vy2 vz2]. Se o mesmo for mais proximo de
        %0 do que de pi, a suposicao de que f/dv>0 e' correta (e,
para
        %v, podem se adotados tanto [vx vy vz] como [vx2 vy2 vz2]).
Caso
        %contrario, tem-se f/dv<0 e deve-se adotar, para v, [vx2 vy2
vz2]
        %ou [-vx -vy -vz].
        V1=[vx vy vz];
        V2=[vx2 vy2 vz2];
        alfa=acos( (V1*(V2')) / (sqrt(sum(V1.*V1))*sqrt(sum(V2.*V2)))
);
        if(alfa>abs(alfa-pi))
            fdv=-fdv;
            V1=(-1)*V1;
        end

%8) Calculam-se Cx,Cy,Cz a partir de a*C, T(1,4) e T(2,4).
        A=[ax ay az;hx hy hz;vx2 vy2 vz2];
        B=[-1;(j0-T(1,4))/fdv;(i0-T(2,4))/fdv]*sqrt(aC2);
        C=inv(A)*B; %C=[Cx;Cy;Cz]

        %Determina-se a matriz-coluna p.
p=[fdv;fdv;hx;hy;hz;ax;ay;az;vx2;vy2;vz2;C(1,1);C(2,1);C(3,1);i0;j0];
end

```

```

function T=matcal(p);

%T=matcal(p);
%
% Calcula a matriz de calibração ou de transformação perspectiva T de
% uma câmera, dados seus parâmetros p, na seguinte ordem: f/du, f/dv,
% hx, hy, hz, ax, ay, az, vx, vy, vz, Cx, Cy, Cz, i0, j0 (p deve ser
% matriz coluna). Os elementos de T são normalizados de forma que se
% tenha T(3,4)=1.
% A definição dos parâmetros p encontra-se em calib.m.

```

```

fdu=p(1,1);
fdv=p(2,1);
hx=p(3,1);
hy=p(4,1);
hz=p(5,1);
ax=p(6,1);
ay=p(7,1);
az=p(8,1);
vx=p(9,1);
vy=p(10,1);
vz=p(11,1);
Cx=p(12,1);
Cy=p(13,1);
Cz=p(14,1);
i0=p(15,1);
j0=p(16,1);

%Alguns produtos internos utilizados
aC=ax*Cx+ay*Cy+az*Cz;
Ch=hx*Cx+hy*Cy+hz*Cz;
Cv=vx*Cx+vy*Cy+vz*Cz;

T(1,1)=(fdu*hx+j0*ax)/(-aC);
T(1,2)=(fdu*hy+j0*ay)/(-aC);
T(1,3)=(fdu*hz+j0*az)/(-aC);
T(1,4)=(fdu*Ch+j0*aC)/aC;

T(2,1)=(fdv*vx+i0*ax)/(-aC);
T(2,2)=(fdv*vy+i0*ay)/(-aC);
T(2,3)=(fdv*vz+i0*az)/(-aC);
T(2,4)=(fdv*Cv+i0*aC)/aC;

T(3,1)=ax/(-aC);
T(3,2)=ay/(-aC);
T(3,3)=az/(-aC);
T(3,4)=1;

```

B. 2 Programa de Retificação

```

function c = extp(a,b)
% calcula o produto externo a x b

c = [0 -a(3) a(2)
     a(3) 0 -a(1)
     -a(2) a(1) 0] * b;

```

```

function [T1,T2,Pn1,Pn2] = rectify(Po1,Po2)
% RECTIFY compute rectification matrices in homogeneous coordinate
%
% [T1,T2,Pn1,Pn2] = rectify(Po1,Po2) computes the rectified
% projection matrices "Pn1" and "Pn2", and the transformation
of the
% retinal plane "T1" and "t2" (in homogeneous coordinate)
which
% perform rectification. The arguments are the two old
projection
% matrices "Po1" and "Po2".
%
% Andrea Fusiello, MVL 1997 (fusiello@dimi.uniud.it)

% focal lenght
% (extp(a,b) is external product of vectors a,b)
au = norm(extp(Po1(1,1:3)', Po1(3,1:3)'));
av = norm(extp(Po1(2,1:3)', Po1(3,1:3)'));

% optical centers
c1 = - inv(Po1(:,1:3))*Po1(:,4);
c2 = - inv(Po2(:,1:3))*Po2(:,4);

% retinal planes
fl = Po1(3,1:3)';
fr = Po2(3,1:3)';

nn = extp(fl,fr);

% solve the four systems
A = [ [c1' 1]' [c2' 1]' [nn' 0]' ]';
[U,S,V] = svd(A);
a3 = 1/norm(V([1 2 3],4)) * V(:,4);

A = [ [c1' 1]' [c2' 1]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
a2 = norm(av)/norm(V([1 2 3],4)) * V(:,4);

A = [ [c1' 1]' [a2(1:3)' 0]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
a1 = norm(au)/norm(V([1 2 3],4)) * V(:,4);

A = [ [c2' 1]' [a2(1:3)' 0]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
b1 = norm(au)/norm(V([1 2 3],4)) * V(:,4);

% adjustment
H = [
    1 0 0
    0 1 0
    0 0 1 ];

% rectifying projection matrices
Pn1 = H * [ a1 a2 a3 ]';
Pn2 = H * [ b1 a2 a3 ]';

% rectifying image transformation
T1 = Pn1(1:3,1:3)* inv(Po1(1:3,1:3));
T2 = Pn2(1:3,1:3)* inv(Po2(1:3,1:3));

```

```

function
[In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,I2,P
2,If2)

% [In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,
% I2,P2,If2);
%
% Realiza a retificação do par de imagens estéreo I1 e I2.
%
% Entradas:
% P1, P2: matrizes de transformação perspectiva das câmaras geradoras
% das imagens I1 e I2.
% If1 e If2: versões binarizadas das imagens I1 e I2. Somente os pixels
% em que If1 e If2 assumem o valor 1 serão considerados
% durante o processo de retificação; assim, será calculado o
% pixel correspondente a I1(i,j) na imagem retificada In1 se
% e somente se If1(i,j)==1.
%
% Saídas:
% In1 e In2: imagens I1 e I2 retificadas.
% Pn1 e Pn2: matrizes de transformação perspectiva as câmaras em
% configuração canônica(hipotéticas) geradoras das imagens
% In1 e In2.
% Ifn1 e Ifn2: imagens If1 e If2 retificadas.
% k_linhas e k_colunas: valores somados às posições de cada pixel das
% imagens retificadas calculadas, com o objetivo
% de evitar índices nulos ou negativo na
% representação matricial estes valores devem ser
% subtraídos dos índices i, j durante o processo
% de reconstituição da cena a partir das
% imagens estéreo).

Po1=P1/sqrt((P1(3,1))^2+(P1(3,2))^2+(P1(3,3))^2);
Po2=P2/sqrt((P2(3,1))^2+(P2(3,2))^2+(P2(3,3))^2);

[T1,T2,Pn1,Pn2] = rectify(Po1,Po2);

[v1 u1]=find(If1~=0);

un1=round((T1(1,1)*u1+T1(1,2)*v1+T1(1,3))./(T1(3,1)*u1+T1(3,2)*v1+T1(3
,3)));
vn1=round((T1(2,1)*u1+T1(2,2)*v1+T1(2,3))./(T1(3,1)*u1+T1(3,2)*v1+T1(3
,3)));

[v2 u2]=find(If2~=0);

un2=round((T2(1,1)*u2+T2(1,2)*v2+T2(1,3))./(T2(3,1)*u2+T2(3,2)*v2+T2(3
,3)));
vn2=round((T2(2,1)*u2+T2(2,2)*v2+T2(2,3))./(T2(3,1)*u2+T2(3,2)*v2+T2(3
,3)));

imin=min([min(vn1) min(vn2)]);
jmin=min([min(un1) min(un2)]);

if(imin<1)
    k_linhas=1-imin;
    vn1=vn1+k_linhas;
    vn2=vn2+k_linhas;

```

```

else
    k_linhas=0;
end

if(jmin<1)
    k_colunas=1-jmin;
    un1=un1+k_colunas;
    un2=un2+k_colunas;
else
    k_colunas=0;
end

In1=zeros(max(vn1),max(un1));
Ifn1=In1;
In2=zeros(max(vn2),max(un2));
Ifn2=In2;

for l=1:size(un1,1)
    In1(vn1(l,1),un1(l,1))=I1(v1(l,1),u1(l,1));
    Ifn1(vn1(l,1),un1(l,1))=1;
end

for l=1:size(un2,1)
    In2(vn2(l,1),un2(l,1))=I2(v2(l,1),u2(l,1));
    Ifn2(vn2(l,1),un2(l,1))=1;
end

```

B. 3 Programa de Correspondência

```

function [Ic,Jc1,Jc2] = corresp(If1,If2,dmin);

%[Ic,Jc1,Jc2,retas1] = corresp(If1,If2,dmin,r);
%
% Calcula os pontos correspondentes do par de imagens binarizadas If1
% e If2, constituídas pelas bordas das imagens estéreo retificadas
% (linhas epipolares horizontais).
%
% dmin é a mínima distância em pixels entre pontos de uma mesma linha
% de If1 e If2 para a qual a correspondência entre os mesmos deverá
% ser feita de forma automática, de acordo com a ordem em que aparecem
% nas duas imagens.
%
%Saídas:
% Ic é o vetor constituído pelos índices das linhas em que se
% encontram os pontos de uma imagem que possuem correspondentes na
% outra imagem.
% Jc1 é o vetor constituído pelos índices das colunas em que se
% encontram os pontos da imagem If1 que possuem correspondentes na
% imagem If2.

```



```

% Jc2 é o vetor constituído pelos índices das colunas em que se
% encontram os pontos da imagem If2 que possuem correspondentes na
% imagem If1.

linhas=min([size(If1,1) size(If2,1)]);
corresp=0; %número de correspondências já efetuadas

%Alocação de memória
Ic=zeros(size(If1,1)*size(If1,2),1);
Jc1=Ic;
Jc2=Ic;
z=Ic; %matriz de zeros;

for k=1:linhas
    A1=If1(k,:);
    A2=If2(k,:);

    %Verificando se a condição imposta por dmin é satisfeita
    s=1;

    I1=find(A1==1);
    J=I1(2:size(I1,2))-I1(1:(size(I1,2)-1));
    if(min(J)<dmin) s=0; end

    I2=find(A2==1);
    J=I2(2:size(I2,2))-I2(1:(size(I2,2)-1));
    if(min(J)<dmin) s=0; end

    %Determinação dos pontos correspondentes
    if((size(I1)==size(I2))&(s==1))
        Ic((corresp+1):(corresp+size(I1,2)),1)=z(1:size(I1,2),1)+k;
        Jc1((corresp+1):(corresp+size(I1,2)),1)=I1';
        Jc2((corresp+1):(corresp+size(I1,2)),1)=I2';
        corresp=corresp+size(I1,2);
    end
end

Ic=Ic(1:corresp,1);
Jc1=Jc1(1:corresp,1);
Jc2=Jc2(1:corresp,1);

```

```

function xyz_cena=demoxyz(I1,P1,I2,P2,k_linhas,k_colunas);

%xyz_cena=demoxyz(I1,P1,I2,P2,k_linhas,k_colunas);
%
% Realiza a reconstituição de pontos de uma cena filmada, a partir
% das imagens correspondentes no par estéreo I1, I2.
%
% P1,P2: matrizes de projeção perspectiva das câmaras geradoras das
% imagens estéreo I1 e I2, respectivamente.
%k_linhas e k_colunas: valores somados às posições de cada pixel das
% imagens retificadas calculadas, com o objetivo
% de evitar índices nulos ou negativos na
% representação matricial (estes valores são
% subtraídos dos índices i, j durante o processo
% de reconstituição da cena a partir das imagens
% estéreo).

```

```
if (nargin==4)
    k_linhas=0;
    k_colunas=0;
end

H=subplot(1,2,1);
imshow(I1);
subplot(1,2,2);
imshow(I2);

linha1=uicontrol('Style','text');
set(linha1,'Position',[10 10 530 80]);
set(linha1,'Units','normalized');

linha1=uicontrol('Style','text');
set(linha1,'Position',[15 62 520 23]);
set(linha1,'BackgroundColor',[1 1 1]);
set(linha1,'Units','normalized');

linha2=uicontrol('Style','text');
set(linha2,'Position',[15 39 520 23]);
set(linha2,'BackgroundColor',[1 1 1]);
set(linha2,'Units','normalized');

linha3=uicontrol('Style','text');
set(linha3,'Position',[15 16 520 23]);
set(linha3,'BackgroundColor',[1 1 1]);
set(linha3,'Units','normalized');

button=-1;
npontos=0; %número de pontos cujas coordenadas já foram calculadas
esquerda=(0==0);
while(button~=27)
    if(esquerda)
        [je,ie,button]=ginput(1);
        set(linha2,'String','');
        set(linha3,'String','');
        set(linha1,'String',sprintf('Coordenadas na imagem da esquerda:
i=%f    j=%f',ie,je));
    else
        [jd,id,button]=ginput(1);
        set(linha2,'String',sprintf('Coordenadas na imagem da direita:
i=%f    j=%f',id,jd));
        npontos=npontos+1;
        xyz_cena(npontos,1:3)=reconsti([ie je],P1,[id
jd],P2,k_linhas,k_colunas);
        set(linha3,'String',sprintf('Coordenadas reconstituídas da cena:
x=%f    y=%f
z=%f',xyz_cena(npontos,1),xyz_cena(npontos,2),xyz_cena(npontos,3)));
    end
    esquerda=~esquerda;
end

close all;
```

B. 4 Programa de Reconstituição

```

function xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)

%xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)
%
%Efetua o cálculo das três coordenadas dos pontos de uma cena
%arbitrária, a partir das posições dos "pixels" correspondentes
%em duas imagens distintas da mesma. Cada linha em ij_imag1 e
%em ij_imag2 deve corresponder às coordenadas i e j do pixel
%de um dos pontos a serem reconstituídos. P1 e P2 são as matrizes
%de transformação perspectiva das câmeras que geraram cada uma
%das imagens. A matriz xyz_cena retornada pela função contém
%em cada linha as coordenadas x,y,z de um unico ponto.
%k_linhas e k_colunas: valores somados às posições de cada pixel das
imagens retificadas
%
%                               calculadas, com o objetivo de evitar índices
nulos ou negativos
%
%                               na representação matricial (estes valores
serão subtraídos
%
%                               dos índices i,j durante o processo de
reconstituição da cena a
%
%                               partir das imagens estéreo).

ij_imag1=[ij_imag1(:,1)-k_linhas ij_imag1(:,2)-k_colunas];
ij_imag2=[ij_imag2(:,1)-k_linhas ij_imag2(:,2)-k_colunas];

k=size(ij_imag1);

if((k(1,2)~=2)|(k(1,1)~=size(ij_imag2,1)))
    disp('Entradas invalidas');
else
    k=k(1,1);
    xyz_cena=zeros(k,3);
    for j=1:k
        A(1,1:3)=P1(1,1:3)-ij_imag1(j,2)*P1(3,1:3);
        A(2,1:3)=P1(2,1:3)-ij_imag1(j,1)*P1(3,1:3);
        A(3,1:3)=P2(1,1:3)-ij_imag2(j,2)*P2(3,1:3);
        A(4,1:3)=P2(2,1:3)-ij_imag2(j,1)*P2(3,1:3);

        y(1,1:1)=-P1(1,4)+ij_imag1(j,2)*P1(3,4);
        y(2,1:1)=-P1(2,4)+ij_imag1(j,1)*P1(3,4);
        y(3,1:1)=-P2(1,4)+ij_imag2(j,2)*P2(3,4);
        y(4,1:1)=-P2(2,4)+ij_imag2(j,1)*P2(3,4);

        xyz_cena(j,1:3)=((inv((A')*A))*A'*y)';
    end
end
end

```

```
function xyz_cena=recocena(I1,P1,I2,P2,thresh,dmin,tipo_filtro);

%xyz_cena=recocena(I1,P1,I2,P2,thresh,dmin);
%
% Reconstitui os pontos da cena filmada, a partir das imagens estéreo
% I1, I2.
%
% thresh: "threshold" aplicado a cada imagem durante a detecção de
% bordas, utilizando a máscara de Sobel.
%
% dmin: distância mínima entre dois pontos das bordas de cada imagem,
% na mesma horizontal, de forma que a correspondência dos mesmos
% seja ainda efetuada de forma automática.
%
% tipo_filtro: pode ser omitido ou assumir o valor "vertical" ou
% "horizontal; indica o tipo de borda que será enfatizado
% durante a aplicação do filtro de Sobel.w

disp('Cálculo das bordas das imagens estéreo...');
if(nargin<7)
    If1=edge(I1,'sobel',thresh);
    If2=edge(I2,'sobel',thresh);
else
    If1=edge(I1,'sobel',thresh,tipo_filtro);
    If2=edge(I2,'sobel',thresh,tipo_filtro);
end

disp('Retificação das imagens filtradas...');
[In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,I2,P
2,If2);

disp('Cálculo das discrepâncias...');
[Ic,Jc1,Jc2] = corresp(Ifn1,Ifn2,dmin);

disp('Reconstituição dos pontos da cena filmada...');
xyz_cena=reconsti([Ic Jc1],Pn1,[Ic Jc2],Pn2,k_linhas,k_colunas);
x=xyz_cena(:,1);
y=xyz_cena(:,2);
z=xyz_cena(:,3);

I=min(find(Ic>min(Ic)));
k=1;
medinf=find(x<0);
medinf=x(medinf);
medinf=sum(medinf)/size(medinf,1);
medsup=find(x>0);
medsup=x(medsup);
medsup=sum(medsup)/size(medsup,1);
while(size(I,1)>0)
    X=x(k:(I-1));
    Y=y(k:(I-1));
    Z=z(k:(I-1));
    if((max(X)<medsup*4)&(min(X)>medinf*4))
        plot3(X,Y,Z, '.');
        hold on;
        plot3(X,Y,Z);
        hold on;
    end
end
```

```

    k=I;
    I=min(find(Ic>Ic(I)));
end
X=x(k:size(x,1));
Y=y(k:size(x,1));
Z=z(k:size(x,1));
if((max(X)<medsup*4)&(min(X)>medinf*4))
    plot3(X,Y,Z, '.');
    hold on;
    plot3(X,Y,Z);
    hold on;
end
plot3(medsup*4,medsup*4,0, 'w');
plot3(medinf*4,medinf*4,0, 'w');
axis('equal');

```

B. 5 Programa de Cálculo da Distância do Objeto

```

function [d1,d2]=distcam(xyz_cena,P1,P2);

%[d1,d2]=distcam(xyz_cena,P1,P2);
%
% Calcula a distância do objeto xyz_cena (reconstituído por recocena)
% aos centros focais das câmeras representadas pelas matrizes de
% transformação perspectiva P1 e P2.

%Cálculo das coordenadas dos centros focais
c1=-inv(P1(:,1:3))*P1(:,4);
c2=-inv(P2(:,1:3))*P2(:,4);

%Cálculo das distâncias
d1=c1'+zeros(size(xyz_cena));
d1=(d1-xyz_cena).^2;
d1=min(sqrt(sum(d1')));

d2=c2'+zeros(size(xyz_cena));
d2=(d2-xyz_cena).^2;
d2=min(sqrt(sum(d2')));

disp('Distância estimada do objeto ao centro focal da câmera 1:');
texto=sprintf('%d cm',d1/10);
disp(texto);

disp(' ');
disp('Distância estimada do objeto ao centro focal da câmera 2:');
texto=sprintf('%d cm',d2/10);
disp(texto);

```

B. 6 Programas Exemplo do Capítulo 6

```
function T=thresh(S,limiar)
%
% T=thresh(S,l)
%
% Esta função realiza o Threshold de uma imagem em níveis de cinza,
% S, e com limiar l

[x,y]=size(S);
T=zeros(x,y);
for i=1:x
    for j=1:y
        if S(i,j)>=limiar
            T(i,j)=1;
        end
    end
end
```

```
function s=filtro(I,M)

%
% s=filtro(I,M)
%
% Esta função realiza filtra a uma imagem S utilizando uma
máscara M
%
% obs: * A matriz M é uma matriz 3x3

[x,y]=size(I);

% O filtro não é definido nas bordas da matriz, logo não é aplica
na primeira
% e na ultima linha e na primeira e ultima coluna logo a matriz de
saída S terá
% duas linhas e duas colunas a menos I.

s=zeros(x-2,y-2);
for i=2:(x-1)
    for j=2:(y-1)

        % O centro da máscara se encontra em i,j da matriz I.Por ser
menor que I, a
        % coordenada de um ponto em s é igual a coordenada em I
reduzida da unidade

        aux=M.*I(i-1:i+1,j-1:j+1);
        s(i-1,j-
1)=aux(1,1)+aux(1,2)+aux(1,3)+aux(2,1)+aux(2,2)+aux(2,3)+aux(3,1)+a
ux(3,2)+aux(3,3);

    end
end
```

```
function f=dx(I)

%
% f=fx(I)
%
% Calcula a derivada parcial de I na direção x
%

[x,y]=size(I);
f=zeros(x,y-1);
for i=2:y % o operador dx não é definido na
primeira coluna
    f(:,i-1)=I(:,i)-I(:,i-1);
end

function f=dy(I)

%
% f=dy(I)
%
% Calcula a derivada parcial de I na direção y%

[x,y]=size(I);
f=zeros(x-1,y);
for i=1:(x-1) % o operador dx não é definido na
ultima linha
    f(i,:)=I(i,:)-I(i+1,:);
end

function f=grad0(S)

%
% f=grad0(S)
%
% Esta função aplica o operador GRADMAG0 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=max(abs(DX),abs(DY));
```

```
function f=grad1(S)

%
% f=grad1(S)
%
% Esta função aplica o operador GRADMAG1 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=abs(DX)+abs(DY);

function f=grad2(S)

%
% f=grad2(S)
%
% Esta função aplica o operador GRADMAG1 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=((DX).^2+(DY).^2).^0.5;
```


Apêndice C

Cálculo das Matrizes Retificadoras $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$

Sejam $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ as matrizes de calibração do sistema estéreo hipotético em configuração canônica que retifica as imagens \mathbf{I}_1 e \mathbf{I}_2 geradas pelo sistema real $(\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2)$. Deseja-se determinar os elementos de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ de forma a garantir que pontos homólogos nas imagens retificadas estejam alinhados (mesma coordenada i) mas sem que sejam alterados os centros ópticos da configuração original, para que as imagens retificadas possam ser calculadas a partir de uma transformação linear conhecida de \mathbf{I}_1 e \mathbf{I}_2 .

Em [2], são propostas condições restritivas aos elementos de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ que garantem não só o alinhamento dos pontos homólogos e a manutenção dos centros ópticos mas também a unicidade da solução. A seguir, demonstra-se como estas condições, citadas no capítulo 3, conduzem aos sistemas de equações 3.3.

Definindo

$$\tilde{\mathbf{P}}_{n1} = \left(\begin{array}{c|c} \mathbf{a}_1^T & a_{14} \\ \mathbf{a}_2^T & a_{24} \\ \mathbf{a}_3^T & a_{34} \end{array} \right) \text{ e } \tilde{\mathbf{P}}_{n2} = \left(\begin{array}{c|c} \mathbf{b}_1^T & b_{14} \\ \mathbf{b}_2^T & b_{24} \\ \mathbf{b}_3^T & b_{34} \end{array} \right) \quad (\text{C.1})$$

tem-se:

1. Os centros ópticos \mathbf{c}_{n1} e \mathbf{c}_{n2} das câmeras descritas por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem coincidir, respectivamente, com os centros \mathbf{c}_1 e \mathbf{c}_2 das câmeras $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$. Como, pela

equação do centro óptico, $\tilde{\mathbf{P}}_{n1} \cdot \begin{pmatrix} \mathbf{c}_{n1} \\ 1 \end{pmatrix} = \tilde{\mathbf{P}}_{n2} \cdot \begin{pmatrix} \mathbf{c}_{n2} \\ 1 \end{pmatrix} = 0$, segue:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_3^T \cdot \mathbf{c}_1 + a_{34} = 0 \\ \mathbf{b}_1^T \cdot \mathbf{c}_2 + b_{14} = 0 \\ \mathbf{b}_2^T \cdot \mathbf{c}_2 + b_{24} = 0 \\ \mathbf{b}_3^T \cdot \mathbf{c}_2 + b_{34} = 0 \end{cases} \quad (\text{C.2})$$

2. $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem apresentar o mesmo plano focal F . Como este plano é formado pelos pontos que não são projetados no plano de retina e portanto não possuem imagem, ou seja, $F = \{\mathbf{w} / \mathbf{a}_3^T \cdot \mathbf{w} + a_{34} = \mathbf{b}_3^T \cdot \mathbf{w} + b_{34} = 0\}$, tem-se:

$$\mathbf{a}_3 = \mathbf{b}_3 \text{ e } a_{34} = b_{34} \quad (\text{C.3})$$

3. Pontos homólogos nas imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} devem estar sempre na mesma linha (o que garante ainda que as novas linhas epipolares serão paralelas a um dos eixos das imagens). Já que a coordenada i da imagem de um ponto \mathbf{w} é definida por

$\frac{\mathbf{a}_2^T \cdot \mathbf{w} + a_{24}}{\mathbf{a}_3^T \cdot \mathbf{w} + a_{34}}$ em \mathbf{I}_{n1} e por $\frac{\mathbf{b}_2^T \cdot \mathbf{w} + b_{24}}{\mathbf{b}_3^T \cdot \mathbf{w} + b_{34}}$ em \mathbf{I}_{n2} , deve-se ter:

$$\frac{\mathbf{a}_2^T \cdot \mathbf{w} + a_{24}}{\mathbf{a}_3^T \cdot \mathbf{w} + a_{34}} = \frac{\mathbf{b}_2^T \cdot \mathbf{w} + b_{24}}{\mathbf{b}_3^T \cdot \mathbf{w} + b_{34}}, \forall \mathbf{w} \quad (\text{C.4})$$

Usando a condição imposta no item 2, chega-se a:

$$\mathbf{a}_2 = \mathbf{b}_2 \text{ e } a_{24} = b_{24} \quad (\text{C.5})$$

Neste ponto, observe-se que as matrizes retificadoras diferem apenas pela primeira linha.

4. As novas matrizes de calibração devem apresentar a normalização $\|\mathbf{a}_3\| = \|\mathbf{b}_3\| = 1$. Esta condição permitirá expressar os pontos principais das imagens retificadas e as resoluções horizontal e vertical em função de relações simples entre \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{b}_1 e \mathbf{a}_3 , nos itens 7 e 8.

5. O plano focal F de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ será paralelo à intersecção dos planos focais de $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$. Como \mathbf{a}_3 é o vetor unitário perpendicular a F , deve-se adotar:

$$\mathbf{a}_3^T \cdot (\mathbf{f}_1 \wedge \mathbf{f}_2) = 0 \quad (\text{C.6})$$

onde \mathbf{f}_1 e \mathbf{f}_2 são os vetores unitários perpendiculares aos planos focais de $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$.

6. Os planos projetados em $i = 0$ e em $j = 0$ nas duas imagens retificadas devem ser perpendiculares entre si (ortogonalidade dos sistemas de referência retificados). Estes planos são perpendiculares respectivamente a \mathbf{a}_1 e \mathbf{a}_2 , no caso da imagem \mathbf{I}_{n1} , e a \mathbf{b}_1 e \mathbf{b}_2 , no caso da imagem \mathbf{I}_{n2} . Assim:

$$\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \text{ e } \mathbf{b}_1^T \cdot \mathbf{b}_2 = 0 \quad (\text{C.7})$$

Considerando-se ainda a condição 3 ($\mathbf{a}_2 = \mathbf{b}_2$), obtêm-se:

$$\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \text{ e } \mathbf{b}_1^T \cdot \mathbf{a}_2 = 0 \quad (\text{C.8})$$

7. Os pontos principais i_0 e j_0 das imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} devem ser nulos. Adotando a normalização $\|\mathbf{a}_3\| = \|\mathbf{b}_3\| = 1$ (condição 4), os pontos principais podem ser expressos como:

$$j_0 = \mathbf{a}_1^T \cdot \mathbf{a}_3 \text{ e } i_0 = \mathbf{a}_2^T \cdot \mathbf{a}_3, \text{ em } \mathbf{I}_{n1}, \quad (\text{C.9})$$

$$j_0 = \mathbf{b}_1^T \cdot \mathbf{b}_3 \text{ e } i_0 = \mathbf{b}_2^T \cdot \mathbf{b}_3, \text{ em } \mathbf{I}_{n2} \quad (\text{C.10})$$

Logo:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{a}_3 = 0 \\ \mathbf{a}_2^T \cdot \mathbf{a}_3 = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \end{cases} \quad (\text{C.11})$$

A condição $\mathbf{b}_2^T \cdot \mathbf{b}_3 = 0$ já é garantida em $\mathbf{a}_2^T \cdot \mathbf{a}_3 = 0$, já que $\mathbf{a}_2 = \mathbf{b}_2$ e $\mathbf{a}_3 = \mathbf{b}_3$.

8. As resoluções horizontal (α_u) e vertical (α_v) da câmera $\tilde{\mathbf{P}}_1$ devem ser mantidas na nova configuração representada por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$. Portanto:

$$\begin{cases} \|\mathbf{a}_1 \wedge \mathbf{a}_3\| = \alpha_u \\ \|\mathbf{a}_2 \wedge \mathbf{a}_3\| = \alpha_v \\ \|\mathbf{b}_1 \wedge \mathbf{a}_3\| = \alpha_u \end{cases} \quad (\text{C.12})$$

A condição $\|\mathbf{b}_2 \wedge \mathbf{b}_3\| = \alpha_v$ fica garantida por $\|\mathbf{a}_2 \wedge \mathbf{a}_3\| = \alpha_v$.

Adotando a identidade vetorial $\|\mathbf{x} \wedge \mathbf{y}\| = \|\mathbf{x}\|^2 \cdot \|\mathbf{y}\|^2 - (\mathbf{x}^T \cdot \mathbf{y})^2$, vem:

$$\begin{cases} \|\mathbf{a}_1\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{a}_1^T \cdot \mathbf{a}_3)^2 = \alpha_u^2 \\ \|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{a}_2^T \cdot \mathbf{a}_3)^2 = \alpha_v^2 \\ \|\mathbf{b}_1\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{b}_1^T \cdot \mathbf{a}_3)^2 = \alpha_u^2 \end{cases} \quad (\text{C.13})$$

Pela condição 7, $\mathbf{a}_1^T \cdot \mathbf{a}_3 = \mathbf{a}_2^T \cdot \mathbf{a}_3 = \mathbf{b}_2^T \cdot \mathbf{a}_3 = 0$, vem:

$$\begin{cases} \|\mathbf{a}_1\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_u^2 \\ \|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_v^2 \\ \|\mathbf{b}_1\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_u^2 \end{cases} \quad (\text{C.14})$$

Finalmente, como $\|\mathbf{a}_3\| = 1$, tem-se:

$$\begin{cases} \|\mathbf{a}_1\| = \alpha_u \\ \|\mathbf{a}_2\| = \alpha_v \\ \|\mathbf{b}_1\| = \alpha_u \end{cases} \quad (\text{C.15})$$

As relações assim obtidas podem ser reagrupadas para que se obtenham os sistemas de equações 3.3. Inicialmente, considerem-se as equações 3 e 6 do sistema obtido no item 1, além da normalização $\|\mathbf{a}_3\| = 1$ e da condição 5; tem-se:

$$\begin{cases} \mathbf{a}_3^T \cdot \mathbf{c}_1 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot \mathbf{c}_2 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot (\mathbf{f}_1 \wedge \mathbf{f}_2) = 0 \\ \|\mathbf{a}_3\| = 1 \end{cases} \quad (\text{C.16})$$

que é o primeiro dos sistemas 3.3.

O segundo sistema é obtido a partir das equações 2 e 5 do item 1, da equação $\mathbf{a}_2^T \cdot \mathbf{a}_3 = 0$ (item 7) e de $\|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_v^2$ (item 8):

$$\begin{cases} \mathbf{a}_2^T \cdot \mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{c}_2 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_2\| = \alpha_v \end{cases} \quad (\text{C.17})$$

A equação 1 do item 1 e as relações $\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0$, $\mathbf{a}_1^T \cdot \mathbf{a}_3 = 0$ e $\|\mathbf{a}_1\| = \alpha_u$ conduzem ao terceiro sistema:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_1\| = \alpha_u \end{cases} \quad (\text{C.18})$$

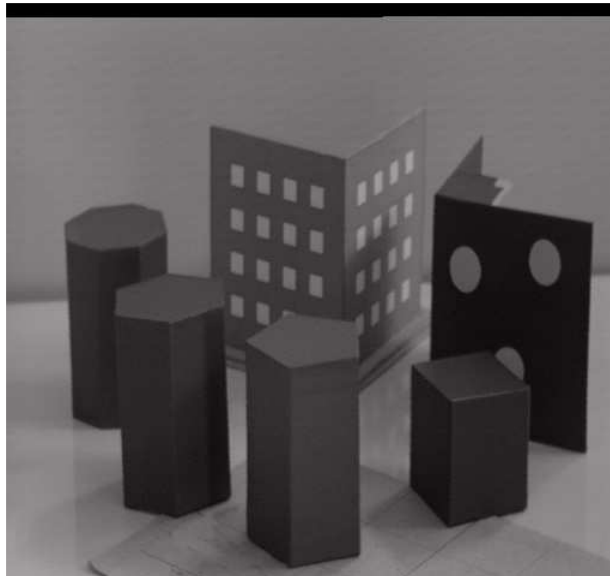
Por fim, o quarto sistema é obtido da equação 4 do item 1 e das relações

$$\mathbf{b}_1^T \cdot \mathbf{a}_2 = 0, \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \text{ e } \|\mathbf{b}_1\| = \alpha_u :$$

$$\begin{cases} \mathbf{b}_1^T \cdot \mathbf{c}_2 + b_{14} = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{b}_1\| = \alpha_u \end{cases} \quad (\text{C.19})$$

UnB – Universidade de Brasília
FT – Faculdade de Tecnologia
ENE – Departamento de Engenharia Elétrica

Visão Computacional Estéreo Aplicada a Formas Poliédricas



Projeto de Graduação de

Cristiano Jacques Miosso Rodrigues

Mirele de Almeida Mencari

Orientador

Dr. Adolfo Bauchspiess

Brasília, DF, Brasil
Março, 1999

Visão Computacional Estéreo Aplicada a Formas Poliédricas

Resumo

Uma das mais importantes aplicações do processamento digital de imagens na área de controle e automação é a guiagem de robôs. Trata-se de determinar, a partir das imagens captadas de um ambiente, as coordenadas de uma trajetória a ser seguida, de objetos a serem manipulados ou ainda de obstáculos presentes, para que se evitem colisões.

Para este fim, a utilização de uma única imagem de uma cena impõe sérias restrições. Na estrutura bidimensional de uma imagem não há informação suficiente para a reconstituição de objetos tridimensionais, ou seja, a noção de profundidade é perdida. Uma solução comumente utilizada é o emprego de sensores ativos, a infravermelho, “laser” ou ultra-som, que permitam recuperar a informação de profundidade e, em conjunto com as imagens mono ou isoladamente, reconstituir os ambientes tridimensionais.

Este trabalho aborda um dos temas atualmente mais pesquisados na área de visão computacional, e que constitui outra técnica viável e promissora para a navegação de robôs: a visão estereo. Por meio do processamento em conjunto das imagens de uma cena geradas por câmeras em posições distintas, denominadas imagens estereo, podem ser calculadas as três coordenadas espaciais dos objetos presentes partindo-se de relações geométricas simples.

Um dos problemas centrais deste intento é a determinação dos pontos homólogos nas imagens estereo. A utilização de mapas de correspondências densos, que permitam determinar os homólogos de cada “*pixel*”, exige um grande esforço computacional, dada a grande complexidade do processamento de imagens.

É proposto então um algoritmo para cálculo da correspondência entre pontos estratégicos nas imagens, levando à obtenção de mapas de profundidade esparsos. Numa fase de pré-processamento, é efetuada uma detecção de bordas das imagens estereo, o que reduz em cerca de 90% a quantidade de pontos utilizados nos cálculos seguintes. Este pré-processamento exige, contudo, que os objetos filmados possam ser bem representados por segmentos de reta. A correspondência é então edetuada a partir da retificação das imagens binárias obtidas.

O sistema foi implementado utilizando o MATLAB 5.0. A aquisição das imagens foi feita utilizando uma câmera monocromática CCD marca PULNIX, série TN-6. Foram construídos um padrão de pontos de referência, para calibração da câmera, e quatro prismas, para teste dos algoritmos desenvolvidos.

Para a validação do sistema implementado foi feita a reconstituição dos cinco sólidos a partir das correspondências manual e automática das imagens estereo. Os resultados obtidos nos dois casos foram equivalentes, mostrando-se satisfatórios diante dos modelos utilizados. Observou-se contudo uma grande sensibilidade a pequenos deslocamentos da câmera em relação às posições de calibração, comprovando a necessidade de uma boa precisão no posicionamento das câmeras.

“O primeiro passo é uma intuição - que surge como uma explosão - em seguida aparecem os problemas. Falham isto e aquilo, vão aparecendo os pequenos erros e dificuldades. Meses de ansiosa observação, estudo e trabalho são necessários antes que o sucesso ou fracasso seja alcançado. “

Thomas Edison

Agradecimentos

Sou grato a todos os que, direta ou indiretamente, auxiliaram-me durante estes últimos cinco anos de estudos, contribuindo de forma significativa para minha formação. Pelo suporte inestimável à realização desta pesquisa, não poderia faltar um registro de especial gratidão:

À minha mãe, Alaide, a quem devo todas as oportunidades de estudo que tive. Sua dedicação, coragem, inteligência e insuperável capacidade de trabalho me inspiram;

Aos meus irmãos Júnior, Alexandre e Simone, ao primo Filipe e à minha tia Marilda, sempre prestativos, que contribuíram com um agradável convívio diário e souberam perdoar todas as minhas falhas;

Ao Professor Dr. Adolfo Bauchspiess, pela dedicação e disponibilidade durante todo o trabalho e pelo conhecimento e experiência que me foram passados. A excelente orientação e as soluções inteligentes e práticas propostas durante as fases mais difíceis da pesquisa foram de grande auxílio;

Ao Professor Dr. Lúcio Martins da Silva, orientador de minha primeira pesquisa de Iniciação Científica, pela excelente orientação, pela confiança em mim depositada e, sobretudo, por ter me mostrado a importância e o prazer da pesquisa científica. Foi para mim uma honra trabalhar ao seu lado;

À pesquisadora Carla e ao técnico Antônio Marrocos, membros do GRACO, sempre dispostos a auxiliar na execução da etapa experimental da pesquisa;

E à colega Mirele de Almeida Mencari, pelo grande empenho e seriedade mostrados em mais de um ano de pesquisa. Sua competência foi imprescindível para a realização deste projeto.

Cristiano Jacques Miosso

Agradecimentos

A realização de um Projeto Final de curso requer um grande esforço e uma considerável dedicação para obtenção das metas propostas. Sua realização se torna mais agradável e fácil com a ajuda de boa parte das pessoas que nós cercam, aos quais sou grata. Mas é indispensável o registro em especial de gratidão:

À família pelo apoio e estímulo durante sua realização e pela compreensão pelos momentos de ausência.

Ao técnico do GRAGO, Antônio Marrocos, pela atenção demonstrada a realizar cada uma de nossas solicitações.

À pesquisadora Carla pelo auxílio, apoio e interesse demonstrado em relação ao nosso trabalho

Em especial ao Professor Doutor Adolfo Bauchspiess pela paciência e dedicação ao nos orientar na realização deste projeto.

E principalmente ao meu colega de trabalho, Cristiano Jacques Miosso Rodrigues, pela amizade, respeito e dedicação conjunta durante o desenvolvimento do projeto.

Mirele de Almeida Mencari

Índice

Capítulo 1 - Introdução	2
Capítulo 2 - Descrição do Sistema Estéreo	5
2.1 Câmera CCD	5
2.2 Pré-Processamento	7
2.3 Retificação.....	8
2.4 Correspondência de Imagens	10
2.5 Reconstituição dos Pontos em 3D	10
Capítulo 3 - Modelamento Matemático de Câmeras Digitais	12
3.1 Modelamento geométrico – a câmera escura de orifício.....	12
3.2 Determinação das coordenadas da imagem de um objeto pontual.....	13
3.3 A matriz de transformação perspectiva.....	16
3.4 Parâmetros intrínsecos e extrínsecos.....	20
Capítulo 4 - A Calibração da Câmera CCD	21
4.1 Medição da matriz de transformação perspectiva	22
4.2 Determinação dos parâmetros intrínsecos e extrínsecos	25
Capítulo 5 - Retificação de Imagens Estéreo	29
5.1 O problema da correspondência entre imagens estéreo	29
5.2 Transformação do plano de retina.....	32
5.3 Cálculo das matrizes de calibração retificadoras de um par de imagens estéreo..	34
5.4 Retificação das imagens do padrão de calibração.....	37

Capítulo 6 - Detecção de Bordas	43
6.1 Thresholding.....	44
6.2 Significado Matemático da Detecção de Bordas.....	46
6.3 Operadores DX e DY	47
Capítulo 7 - Metodologia	56
7.1 Aquisição de Imagens	56
7.1 Calibração da câmera	59
7.2 Retificação das imagens estéreo.....	60
7.3 Correspondência manual das imagens estéreo	60
7.4 Correspondência automática das imagens estéreo	61
7.5 Reconstituição das imagens tridimensionais.....	63
Capítulo 8 - Resultados Experimentais	65
8.1 Resultados da calibração da câmera CCD.....	65
8.2 Reconstituição de objetos a partir da correspondência manual das imagens estéreo	68
8.3 Reconstituição de objetos a partir da correspondência automática das imagens estéreo.....	71
8.4 Estimativa das distâncias dos objetos filmados.....	72
Capítulo 9 - Conclusão	81
Referências Bibliograficas	83
Apêndice A - Sensor	84
A.1 Sensores na Robótica	86
A.2 Sensores Internos	86
A.3 Sensores Externos	88
Apêndice B - Listagens dos Programas	90
B. 1 Programa de Calibração da Câmera CCD	90
B. 2 Programa de Retificação	94
B. 3 Programa de Correspondência	97
B. 4 Programa de Reconstituição	100
B. 5 Programa de Cálculo da Distância do Objeto	102

B. 6 Programas Exemplo do Capítulo 6 103

Apêndice C - Cálculo das Matrizes Retificadoras $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ 106

Índice de Figuras

Figura 2 .1 : Diagrama em blocos do sistema de visão estéreo desenvolvido neste trabalho.	6
Figura 2.2 : Câmera escura de orifício	6
Figura 2.3 : Dada uma imagem A'B' no plano de imagens existe uma infinidade de possibilidades de posições e dimensões do objeto que a formou.	7
Figura 2.4 : Configuração canônica das câmeras para um sistema de visão estéreo.....	9
Figura 2.5 : Par de imagens obtido de um sistema de visão estéreo	9
Figura 2.6 : Configuração divergente das câmeras de um sistema estéreo	10
Figura 3.1 : A câmera escura de orifício	12
Figura 3.2 : Modelo geométrico simplificado da câmera escura de orifício (foco C, plano focal F e plano de retina R).	13
Figura 3.3 : A formação de imagem numa câmera escura de orifício (Projeção Perspectiva).....	14
Figura 3.4 : Visão bidimensional da projeção perspectiva ilustrada em 3.3.	15
Figura 4.1 : O padrão de calibração empregado.....	23
Figura 5.1 : Correspondência entre pontos de imagens estéreo geradas por câmeras em configuração arbitrária.	30
Figura 5.2 : Correspondência entre pontos de imagens estéreo geradas por câmeras em configuração canônica.	31
Figura 5.3 : Transformação dos planos de retina de duas imagens estéreo, durante o processo de retificação.	32
Figura 5.4 : Imagens não - retificadas do padrão de calibração fornecidas pela câmera CCD em duas posições distintas, com um ângulo de 160° entre os planos focais. Observar que os pontos homólogos destacados nas duas imagens não estão alinhados.	38
Figura 5.5 : Imagens retificadas do padrão de calibração. Observar que os pontos homólogos destacados nas duas imagens estão alinhados.....	38

Figura 6.1: Imagem de um anel. Esta imagem será a base de todos os exemplos desenvolvidos neste capítulo.	45
Figura 6.2: Matriz correspondente a imagem F.....	45
Figura 6.3 : Imagens de saída do operador <i>thresholding</i> para diferentes valores de limiar l	45
Figura 6.4 : Gráfico da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.	48
Figura 6.5 : Gráfico da derivada da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.....	48
Figura 6.6 : Gráfico da segunda derivada da função de luminosidade ao longo de uma linha de uma matriz imagem hipotética.	48
Figura 6.7: Matrizes resultantes da aplicação dos operadores DX e DY sobre a matriz f	49
Figura 6.8: Matrizes resultantes da aplicação dos operadores DX e DY sobre a matriz f	49
Figura 6.9 : Diagrama em blocos do operador GRAMADMAG0	51
Figura 6.10 : Diagrama em blocos do operador GRAMADMAG1	51
Figura 6.11 : Diagrama em blocos do operador GRAMADMAG2	51
Figura 6. 12: Imagens A, B e C referentes as matrizes a , b e c com limiar de <i>thresholding</i> igual a 0,3.	52
Figura 6.13 : Matrizes resultantes da aplicação dos operadores GRADMAG0, GRADMAG1 e GRADMAG2 sobre a matriz f	53
Figura 6.14 : Diagrama em blocos de um processo de detecção de bordas.	54
Figura 6.15 : Imagens de saída para cada uma das combinações de máscara (Prewitt e Sobel) e norma (l_∞ , l_1 e l_2).....	55
Figura 6.16 : Diagrama em blocos do processo de detecção de bordas implementado.	55
Figura 7 . 1: Objetos utilizados durante a parte experimental. A partir da central superior no sentido horário: padrão com trinta e dois pontos, padrão com seis pontos, cubo, prisma de base pentagonal, prisma com base hexagonal e prisma com base octogonal.....	56
Tabela 7.1: Posicionamento dos vértices da câmera em relação ao referencial O, para as duas situações consideradas.....	57

- Figura 7.2 : Vista superior da mesa de trabalho. A posições A e B indicam as possíveis posições que a câmera poderia ocupar..... 58
- Figura 7.3: Base de papelão que firmava a câmera na mesa. 58
- Figura 7.4: Janela de operação do programa *centroid.m*. 59
- Figura 7.5: Janela de operação do programa *demoxyz.m*. No exemplo é efetuada a correspondência manual de pontos nas imagens não retificadas do padrão de calibração. 61
- Figura 7.6: Correspondência automática de imagens estéreo retificadas. 62
- Figura 8.1 : Reconstituição das coordenadas espaciais de um cubo, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho. 73
- Figura 8.1 : Reconstituição das coordenadas espaciais de um cubo, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho. 74
- Figura 8.2 : Reconstituição das coordenadas espaciais de um prisma de base pentagonal, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho. 75
- Figura 8.2 : Reconstituição das coordenadas espaciais de um prisma de base pentagonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho. 76
- Figura 8.3: Reconstituição das coordenadas espaciais de um prisma de base hexagonal, a partir das correspondências manual e automática de duas imagens estéreo. (a) Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho. 77
- Figura 8.3: Reconstituição das coordenadas espaciais de um prisma de base hexagonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho. 78
- Figura 8.4: Reconstituição das coordenadas espaciais de um prisma de base octogonal, a partir das correspondências manual e automática de duas imagens estéreo. (a)

Imagens estéreo, (b) Pontos reconstituídos a partir da correspondência manual, em verde, e modelo conhecido do objeto filmado, em vermelho, em verde, e modelo conhecido do objeto filmado, em vermelho.....	79
Figura 8.4: Reconstituição das coordenadas espaciais de um prisma de base octogonal, a partir das correspondências manual e automática de duas imagens estéreo. (c) Bordas Detectadas, (d) Pontos reconstituídos a partir da correspondência automática, em azul, e modelo conhecido do objeto filmado, em vermelho.....	80
Figura A-1 : Sensor de posição de Mactarlane e Donath	87
Figura A-2 : Sensor de proximidade	88

Apêndice D

Sensor

Qualquer ser vivo, seja qual for o seu tamanho e complexidade, precisa obter informações sobre o meio em que vive de modo a garantir a sua sobrevivência. Estas informações são utilizadas na obtenção de alimentos, manutenção ou alteração da taxa de metabolismo do organismo e decisão sobre o momento de reprodução. Estas informações permitem que os seres vivos sejam autônomos e não necessitem de outro “ser” que tomem estas decisões por eles.

Dentre os seres vivos, os animais pluricelulares mais evoluídos (insetos, répteis, aves e mamíferos) apresentam a capacidade de se locomoverem dentro do seu meio ambiente e tomarem decisões sobre o modo de interação com ele. Por apresentarem uma maior autonomia em relação ao meio, estes animais necessitam de uma maior quantidade de informações sobre o mesmo, que são fornecidas pelos órgãos de sentido (tato, paladar, olfato, audição e visão).

Desde o início de sua existência o Homem têm criado ferramentas que o auxiliem na execução de suas tarefas. Com o passar dos séculos, estas ferramentas têm sido aprimoradas de modo a torná-las autônomas, ou seja independentes do homem para executar determinada tarefa. Esta independência se refere a capacidade de tomar decisões inerentes ao processo da tarefa e obter energia , gasta na realização, de fontes que não utilizem energia muscular humana.

Para pequenas tarefas, como moer milho, o homem há alguns séculos conseguiu construir máquinas independentes do esforço humano. Porém, máquinas com a capacidade de tomar decisões capazes de realizar grandes tarefas, como explorar a cratera de um vulcão, só começaram a ser construídas nos meados do século XX. Surgindo, assim, os primeiros robôs, mecanismos autônomos capazes de realizar tarefas complexas sem a intervenção humana.

Para conseguir a desejada autonomia, estes mecanismos, como os seres vivos, necessitam obter informações sobre o meio ambiente em que se encontram. Estas informações são obtidas através de sensores que realizam a mesma função dos órgãos dos sentidos dos seres vivos. Os sensores são capazes de fornecer aos robôs o sentido do “tato”- detetores de pressão e calor – “olfato”, “paladar”- detecção de substâncias químicas existentes no meio – “audição”- reconhecimento de voz e sonar – e “visão”- reconhecimento de padrões e navegação. Sendo o último o assunto deste trabalho.

A visão fornece uma quantidade razoável de informações sobre o meio:

- Luminosidade;
- Presença, distância, forma e tamanho de corpos existentes no meio; e
- Reconhecimento e distinção de um corpo em especial dentre um conjunto de corpos.

Os seres vivos utilizam apenas um tipo de “sensor” – olhos – para obtenção de todas estas informações. Os robôs, ao contrário, podem utilizar um ou mais tipos diferentes de sensores para obtenção destes dados.

A câmera de vídeo é um sensor capaz de fornecer a maioria destas informações quando é utilizada apenas uma e todas quando utilizado um sistema estéreo (com duas ou mais câmeras). Chama-se a atenção para o fato de que a câmera não fornece estas informações prontas, como num termômetro, sendo necessário um processamento. O uso da visão estéreo para obtenção das informações sobre presença, distância, forma e tamanho de corpos existentes no meio é o escopo dentro da visão computacional que é abordado neste texto.

A seguir será feita uma breve descrição sobre sensores.

A.4 Sensores na Robótica

O sucesso na aplicação de um robô numa determinada tarefa depende exclusivamente da sua capacidade de medir parâmetros internos do seu sistema e externos referentes ao ambiente. Esta capacidade é determinada pela quantidade e diversidade de sensores existentes no robô.

Em robótica, os sensores são usados para:

- Medir variáveis de controle;
- Localização de objetos;
- Evitar situações de falha;
- Evitar colisões;
- Monitorização da interação com o ambiente;
- Monitorização das mudanças no ambiente que interfiram no processo; e
- Inspeção dos resultados do processo.

Ainda neste campo, os sensores são classificados em dois tipos: sensores internos e sensores externos. Os primeiros são usados para medir parâmetros do robô que possam afetar o bom funcionamento do seu sistema. O segundo são usados para medir características do sistema e a posição relativa do robô nele.

A.5 Sensores Internos

Parâmetros internos que podem ser medidos são: posição, movimento e esforço de todas as partes do robô (braços, articulações, etc.), temperatura e umidade interna. O conhecimento destes parâmetros permite evitar estragos em qualquer parte do robô que o parariam temporariamente ou permanentemente. As medidas de posição e movimento são relativas a um “frame” e exemplificadas nos próximos parágrafos.

As medidas de posição e movimento, neste caso, são referentes ao movimento de partes do robô, como um braço, e não movimento dele como um todo no ambiente.

Os robôs são modelados em dois espaços: junta e cartesiano. A posição, velocidade e aceleração podem ser medidos em ambos os espaços. O espaço cartesiano descreve o

movimento total do sistema, por exemplo a posição do “punho” de um braço mecânico, enquanto o espaço de juntas descreve se preocupa com o movimento de cada articulação individualmente.

Um exemplo de **sensor de medida de posição** no espaço cartesiano foi desenvolvido por Mactarlane e Donath(1984) [10], figura A.1. O método rastreia o punho do braço de um robô utilizando um scanner a laser. Uma fina cortina de luz é produzida por lentes cilíndricas que focam o laser num espelho plano rotativo que varre o feixe através do volume de interesse. Este feixe de luz é detectado por fotodiodos colocados em posições fixas, que funcionam como referencia, e nas articulações do sistema. A velocidade de varredura é fixa e função da velocidade de rotação do espelho. A diferença de tempo existente entre a detecção do feixe pelo diodo fixo e pelo diodo da articulação é proporcional a distância entre eles no sentido da varredura. O uso de vários lasers em direções diferentes permite obter a posição do punho do robô.

No espaço de juntas a medida de posição pode ser feita utilizando um potenciômetro e de velocidade utilizando um tacômetro. A aceleração pode ser medida através de sistemas de balança de torque e medidores de pressão.

Os **sensores de força** existem em ambas classes de sensores, internos e externos, sendo a sua classificação dependente da utilização. Existem dois métodos comuns utilizados na confecção destes sensores: deflexão de um elemento elástico e variação da resistência de acordo com a variação da força aplicada.

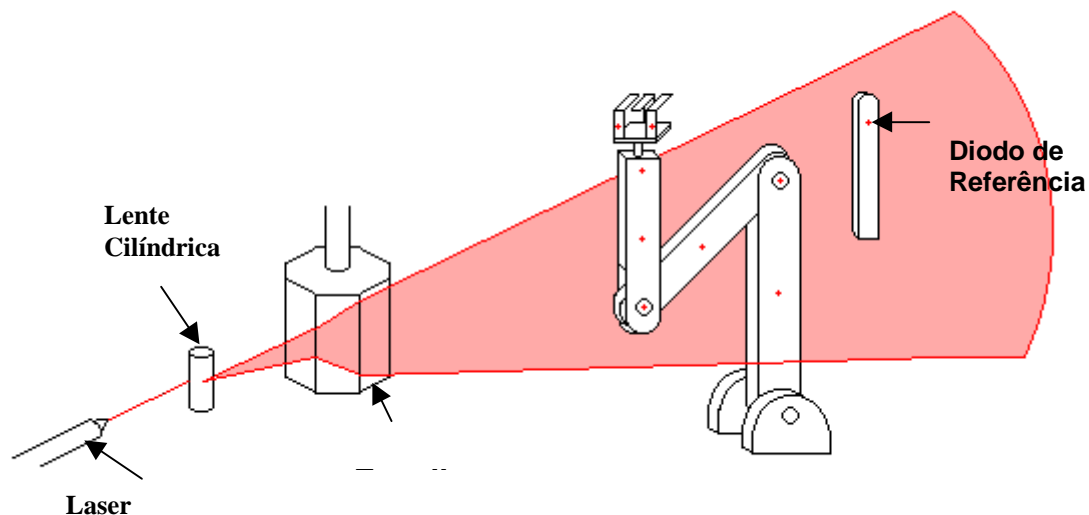


figura A.3 : Sensor de posição de Mactarlane e Donath

A.6 Sensores Externos

Sensores externos medem a interação do robô com o meio. Existe uma infinidade de tipos deste sensor que são usados para as mais variáveis funções como o tato, proximidade, velocidade, som, umidade, composição gasosa, pressão atmosférica, temperatura, etc. Uma observação válida é que a classificação de sensores como internos e externos refere-se ao uso do que a função ou o modelo de sensor.

Os **sensores de tato** são usados para dois propósitos: detectar a presença de toque e detectar as características do objeto tocado. O primeiro é usado quando se deseja saber que parte do robô está tocando algo e o segundo quando se deseja levantar características sobre o objeto que está sendo tocado.

Sensores de proximidade detectam um objeto perto do sensor sem toca-lo fisicamente. Dois princípios são utilizados na sua construção: modificação de um sinal emitido e distúrbio de alguma característica do ambiente.

A modificação do sinal emitido ocorre quando parte do sinal emitido é bloqueado ou refletido na direção de um receptor. **Sensores de proximidade ópticos** utilizam este princípio. Um exemplo é o sensor constituído de um emissor e receptor de luz como mostrado na figura A.2. Este sensor detecta a presença do objeto pela reflexão da do feixe de luz na direção do receptor que ocorre quando o objeto se encontra na região de sensibilidade do sensor.

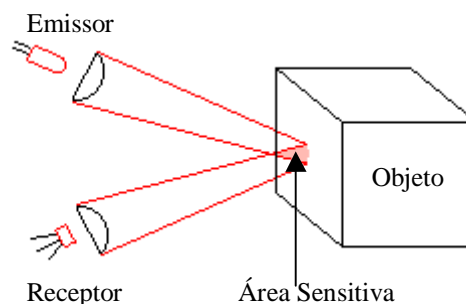


figura A.4 : Sensor de proximidade

Sensores de profundidade medem a distância existente entre o sensor e um objeto na cena e se baseia em dois princípios físicos: tempo de propagação de um pulso e triangulação.

O **sensor de tempo de propagação** emite um pulso e mede o tempo que leva desta emissão até o retorno do mesmo. A metade deste tempo de propagação multiplicado pela velocidade do pulso no meio de propagação fornece a distância existente entre o sensor e o objeto. O exemplo mais comum deste tipo de sensor é o radar.

Sensores de triangulação medem a distância pelo uso de princípios geométricos. A triangulação requer a detecção do objeto a partir de dois pontos de visão e o conhecimento da distância dos pontos em questão. Estes dois pontos pode ser obtidos movimentando um simples sensor ou pelo uso de vários sensores distribuídos no espaço. Um exemplo é o sistema em que um dos sensores emite um feixe de luz que é refletido e detectado pelo segundo sensor, quando isto ocorre, os ângulos de emissão e recepção do feixe de luz são armazenados, e a distância calculada. O sistema de visão estéreo, que é o objeto principal deste trabalho, é um sensor passivo de triangulação.

Outro tipo de sensor de profundidade é o **sonar** que se baseia no fenômeno de eco das ondas sonoras. O princípio é o mesmo utilizado nos sensores de tempo de vôo: o tempo de retorno é proporcional a distância. A diferença é que o sonar utiliza ondas sonoras ao invés de ondas eletromagnéticas.

Apêndice E

Listagens dos Programas

B. 7 Programa de Calibração da Câmera CCD

```
function c=centroid(I,pixels_h,pixels_v,criterio,limiar)

% c=centroid(I,pixels_h,pixels_v,criterio,limiar);
%
% Calcula os centróides de diferentes janelas de uma imagem. Esta é
% inicialmente exibida na tela, sendo que a posição de cada janela é
% definida pelo usuário com o auxílio do "mouse" (botão esquerdo). Os
% dois "pixels" que definem cada centróide podem ser armazenados em
% uma linha da matriz c, bastando que se pressione a tecla espaço
% quando desejado (um sinal sonoro indica que os valores foram
% armazenados). Após o cálculo do último centróide desejado, deve ser
% pressionada a tecla 'ESC'.
%
% I: imagem em níveis de cinza.
% pixels_h: número de pixels presentes numa linha horizontal de cada
%           janela cujo centróide deve ser calculado;
% pixels_v: número de pixels presentes numa linha vertical de cada
%           janela cujo centróide deve ser calculado;
% limiar: valor mínimo ou máximo de intensidade de um ponto em I
%         para que o mesmo seja incluído no cálculo dos centroides.
% criterio: vale '>' ou '<' e estabelece se o valor limiar é mínimo ou
%           máximo, respectivamente.
button=-1;
a_registrar=0;
armazenados=0;
a=round(pixels_h/2);
b=round(pixels_v/2);
som=(0:50000);
som=sin(2*pi*445*som/100000);
imshow(I,256);
while(button~=27)
    [x,y,button]=ginput(1);
    if(button==1)
        x=round(x);
```

```

y=round(y);
A=I(y-b:y+pixels_v-b-1,x-a:x+pixels_h-a-1);
if(critério(1,1)=='>')
    [i,j]=find(A>=limiar);
else
    [i,j]=find(A<=limiar);
end
i=sum(i)/size(i,1);
j=sum(j)/size(j,1);
i=i+y-b-1;
j=j+x-a+1;
imshow(I,256);
X=[x-a;x-a];
Y=[y-b;y+pixels_v-b-1];
line(X,Y);
X=[x+pixels_h-a-1;x+pixels_h-a-1];
Y=[y-b;y+pixels_v-b-1];
line(X,Y);
X=[x-a;x+pixels_h-a-1];
Y=[y-b;y-b];
line(X,Y);
X=[x-a;x+pixels_h-a-1];
Y=[y+pixels_v-b-1;y+pixels_v-b-1];
line(X,Y);
'centroide'
i,j
a_registrar=1;
else
    if((button==32)&(a_registrar==1))
        a_registrar=0;
        armazenados=armazenados+1;
        c(armazenados,1)=i;
        c(armazenados,2)=j;
        sound(som,100000);
    end
end
end
close(1);

```

```

function [T,p]=calib(xyz_cena,ij_imag)

%[T,p]=calib(xyz_cena,ij_imag);
%
% Calcula a matriz de transformação perspectiva T e os parâmetros p
% ( ver relação abaixo ) de uma câmera, devendo para tanto ser
% fornecidos:
% *as coordenadas x,y,z de pelo menos 6 pontos não coplanares de uma
% cena arbitrária ( coordenadas estas referenciadas a um sistema
% também arbitrário).Cada linha da matriz xyz_cena deve ser formada
% pelas coordenadas de um único desses pontos.
% *os índices i e j dos pixels correspondentes àqueles pontos na
% imagem gerada pela câmera. Na matriz ij_imag, cada linha l deve
% conter os índices associados ao ponto cujas coordenadas são os
% elementos da linha l de xyz_cena.
%
% A função calib então retorna:
% *T: matriz de calibração ou de transformação perspectiva (3x4), com
% os elementos normalizados para que se tenha T(3,4)=1.
% *p: matriz-coluna que contém os parâmetros da câmera, na seguinte

```

```

% ordem: f/du, f/dv, hx, hy, hz, ax, ay, az, vx, vy, vz, Cx, Cy, Cz,
% i0,j0.
%

if((~(size(xyz_cena,1)==size(ij_imag,1)))|(~(size(xyz_cena,2)==3))|(~(
size(ij_imag,2)==2)))
    'Entradas invalidas.'
else

    %Cálculo de T
    %Determinacao da matriz U que minimiza os elementos de
    %E=(A*U-B)'*(A*U-B)
    %Determinacao de A e de B
    n=size(xyz_cena,1);
    k=1;
    while(k<=n)
        A(2*k-1,1:3)=xyz_cena(k,1:3);
        A(2*k-1,4)=1;
        A(2*k-1,5:8)=zeros(1,4);
        A(2*k-1,9:11)=xyz_cena(k,1:3)*(ij_imag(k,2))*(-1);
        A(2*k,1:4)=zeros(1,4);
        A(2*k,5:7)=xyz_cena(k,1:3);
        A(2*k,8)=1;
        A(2*k,9:11)=xyz_cena(k,1:3)*(ij_imag(k,1))*(-1);
        B(2*k-1,1)=ij_imag(k,2);
        B(2*k,1)=ij_imag(k,1);
        k=k+1;
    end
    %Cálculo de U
    U=((inv((A')*A))*(A'))*B;
    %Determinacao de T
    T(1,1:4)=(U(1:4,1))';
    T(2,1:4)=(U(5:8,1))';
    T(3,1:3)=(U(9:11,1))';
    T(3,4)=1;

    %Determinacao dos parametros da camera (pelo algoritmo apresentado
    %por Ganapathy)
    %1) Determina-se (a*C)^2 a partir de T3*T3.
    V1=T(3,1:3);
    aC2=1/(V1*V1');

    %2) Determina-se j0 a partir de T1*T3.
    V2=T(1,1:3);
    j0=aC2*(V1*V2');

    %3) Determina-se i0 a partir de T2*T3.
    V2=T(2,1:3);
    i0=aC2*(V1*V2');

    %4) Determinam-se ax,ay,az a partir de T(3,1), T(3,2), T(3,3) (a
    %suposicao de que A*C<0 e' explicada no item 7).
    ax=T(3,1)*sqrt(aC2);
    ay=T(3,2)*sqrt(aC2);
    az=T(3,3)*sqrt(aC2);

    %5) Determina-se o modulo de f/du a partir de T1*T1.
    V1=T(1,1:3);
    fdu=sqrt(V1*(V1')*aC2-j0^2);

    %6) Determina-se o modulo de f/dv a partir de T2*T2.

```

```

        fdv=sqrt(V2*(V2')*aC2-i0^2);

%7) Adotam-se C*a<0 e f/du>0 (o que equivale a arbitrar as
%direcoes de a e de u). Aceita a convencao, supoe-se f/dv>0 e
%calculam-se hx,hy,hz,vx,vy,vz a partir de T(1,1), T(1,2),
%T(1,3), T(2,1), T(2,2), T(2,3).
        hx=(T(1,1)*sqrt(aC2)-j0*ax)/fdv;
        hy=(T(1,2)*sqrt(aC2)-j0*ay)/fdv;
        hz=(T(1,3)*sqrt(aC2)-j0*az)/fdv;
        vx=(T(2,1)*sqrt(aC2)-i0*ax)/fdv;
        vy=(T(2,2)*sqrt(aC2)-i0*ay)/fdv;
        vz=(T(2,3)*sqrt(aC2)-i0*az)/fdv;

        %Em seguida, efetua-se um segundo cálculo de vx,vy,vz
        %partindo-se de v=a*h (produto vetorial).
        vx2=ay*hz-az*hy;
        vy2=az*hx-ax*hz;
        vz2=ax*hy-ay*hx;

        %Determina-se entao o angulo entre os valores experimentais
        %[vx vy vz] e [vx2 vy2 vz2]. Se o mesmo for mais proximo de
        %0 do que de pi, a suposicao de que f/dv>0 e' correta (e,
para
        %v, podem se adotados tanto [vx vy vz] como [vx2 vy2 vz2]).
Caso
        %contrario, tem-se f/dv<0 e deve-se adotar, para v, [vx2 vy2
vz2]
        %ou [-vx -vy -vz].
        V1=[vx vy vz];
        V2=[vx2 vy2 vz2];
        alfa=acos( (V1*(V2')) / (sqrt(sum(V1.*V1))*sqrt(sum(V2.*V2)))
);
        if(alfa>abs(alfa-pi))
            fdv=-fdv;
            V1=(-1)*V1;
        end

%8) Calculam-se Cx,Cy,Cz a partir de a*C, T(1,4) e T(2,4).
        A=[ax ay az;hx hy hz;vx2 vy2 vz2];
        B=[-1;(j0-T(1,4))/fdv;(i0-T(2,4))/fdv]*sqrt(aC2);
        C=inv(A)*B; %C=[Cx;Cy;Cz]

        %Determina-se a matriz-coluna p.
p=[fdv;fdv;hx;hy;hz;ax;ay;az;vx2;vy2;vz2;C(1,1);C(2,1);C(3,1);i0;j0];
end

```

```

function T=matcal(p);

%T=matcal(p);
%
% Calcula a matriz de calibração ou de transformação perspectiva T de
% uma câmera, dados seus parâmetros p, na seguinte ordem: f/du, f/dv,
% hx, hy, hz, ax, ay, az, vx, vy, vz, Cx, Cy, Cz, i0, j0 (p deve ser
% matriz coluna). Os elementos de T são normalizados de forma que se
% tenha T(3,4)=1.
% A definição dos parâmetros p encontra-se em calib.m.

```

```

fdu=p(1,1);
fdv=p(2,1);
hx=p(3,1);
hy=p(4,1);
hz=p(5,1);
ax=p(6,1);
ay=p(7,1);
az=p(8,1);
vx=p(9,1);
vy=p(10,1);
vz=p(11,1);
Cx=p(12,1);
Cy=p(13,1);
Cz=p(14,1);
i0=p(15,1);
j0=p(16,1);

%Alguns produtos internos utilizados
aC=ax*Cx+ay*Cy+az*Cz;
Ch=hx*Cx+hy*Cy+hz*Cz;
Cv=vx*Cx+vy*Cy+vz*Cz;

T(1,1)=(fdu*hx+j0*ax)/(-aC);
T(1,2)=(fdu*hy+j0*ay)/(-aC);
T(1,3)=(fdu*hz+j0*az)/(-aC);
T(1,4)=(fdu*Ch+j0*aC)/aC;

T(2,1)=(fdv*vx+i0*ax)/(-aC);
T(2,2)=(fdv*vy+i0*ay)/(-aC);
T(2,3)=(fdv*vz+i0*az)/(-aC);
T(2,4)=(fdv*Cv+i0*aC)/aC;

T(3,1)=ax/(-aC);
T(3,2)=ay/(-aC);
T(3,3)=az/(-aC);
T(3,4)=1;

```

B. 8 Programa de Retificação

```

function c = extp(a,b)
% calcula o produto externo a x b

c = [0 -a(3) a(2)
      a(3) 0 -a(1)
      -a(2) a(1) 0] * b;

```

```

function [T1,T2,Pn1,Pn2] = rectify(Po1,Po2)
% RECTIFY compute rectification matrices in homogeneous coordinate
%
% [T1,T2,Pn1,Pn2] = rectify(Po1,Po2) computes the rectified
% projection matrices "Pn1" and "Pn2", and the transformation
of the
% retinal plane "T1" and "t2" (in homogeneous coordinate)
which
% perform rectification. The arguments are the two old
projection
% matrices "Po1" and "Po2".
%
% Andrea Fusiello, MVL 1997 (fusiello@dimi.uniud.it)

% focal lenght
% (extp(a,b) is external product of vectors a,b)
au = norm(extp(Po1(1,1:3)', Po1(3,1:3)'));
av = norm(extp(Po1(2,1:3)', Po1(3,1:3)'));

% optical centers
c1 = - inv(Po1(:,1:3))*Po1(:,4);
c2 = - inv(Po2(:,1:3))*Po2(:,4);

% retinal planes
fl = Po1(3,1:3)';
fr = Po2(3,1:3)';

nn = extp(fl,fr);

% solve the four systems
A = [ [c1' 1]' [c2' 1]' [nn' 0]' ]';
[U,S,V] = svd(A);
a3 = 1/norm(V([1 2 3],4)) * V(:,4);

A = [ [c1' 1]' [c2' 1]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
a2 = norm(av)/norm(V([1 2 3],4)) * V(:,4);

A = [ [c1' 1]' [a2(1:3)' 0]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
a1 = norm(au)/norm(V([1 2 3],4)) * V(:,4);

A = [ [c2' 1]' [a2(1:3)' 0]' [a3(1:3)' 0]' ]';
[U,S,V] = svd(A);
b1 = norm(au)/norm(V([1 2 3],4)) * V(:,4);

% adjustment
H = [
    1 0 0
    0 1 0
    0 0 1 ];

% rectifying projection matrices
Pn1 = H * [ a1 a2 a3 ]';
Pn2 = H * [ b1 a2 a3 ]';

% rectifying image transformation
T1 = Pn1(1:3,1:3)* inv(Po1(1:3,1:3));
T2 = Pn2(1:3,1:3)* inv(Po2(1:3,1:3));

```

```

function
[In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,I2,P
2,If2)

% [In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,
% I2,P2,If2);
%
% Realiza a retificação do par de imagens estéreo I1 e I2.
%
% Entradas:
% P1, P2: matrizes de transformação perspectiva das câmaras geradoras
% das imagens I1 e I2.
% If1 e If2: versões binarizadas das imagens I1 e I2. Somente os pixels
% em que If1 e If2 assumem o valor 1 serão considerados
% durante o processo de retificação; assim, será calculado o
% pixel correspondente a I1(i,j) na imagem retificada In1 se
% e somente se If1(i,j)==1.
%
% Saídas:
% In1 e In2: imagens I1 e I2 retificadas.
% Pn1 e Pn2: matrizes de transformação perspectiva as câmaras em
% configuração canônica (hipotéticas) geradoras das imagens
% In1 e In2.
% Ifn1 e Ifn2: imagens If1 e If2 retificadas.
% k_linhas e k_colunas: valores somados às posições de cada pixel das
% imagens retificadas calculadas, com o objetivo
% de evitar índices nulos ou negativo na
% representação matricial estes valores devem ser
% subtraídos dos índices i, j durante o processo
% de reconstituição da cena a partir das
% imagens estéreo).

Po1=P1/sqrt((P1(3,1))^2+(P1(3,2))^2+(P1(3,3))^2);
Po2=P2/sqrt((P2(3,1))^2+(P2(3,2))^2+(P2(3,3))^2);

[T1,T2,Pn1,Pn2] = rectify(Po1,Po2);

[v1 u1]=find(If1~=0);

un1=round((T1(1,1)*u1+T1(1,2)*v1+T1(1,3))./(T1(3,1)*u1+T1(3,2)*v1+T1(3
,3)));
vn1=round((T1(2,1)*u1+T1(2,2)*v1+T1(2,3))./(T1(3,1)*u1+T1(3,2)*v1+T1(3
,3)));

[v2 u2]=find(If2~=0);

un2=round((T2(1,1)*u2+T2(1,2)*v2+T2(1,3))./(T2(3,1)*u2+T2(3,2)*v2+T2(3
,3)));
vn2=round((T2(2,1)*u2+T2(2,2)*v2+T2(2,3))./(T2(3,1)*u2+T2(3,2)*v2+T2(3
,3)));

imin=min([min(vn1) min(vn2)]);
jmin=min([min(un1) min(un2)]);

if(imin<1)
    k_linhas=1-imin;
    vn1=vn1+k_linhas;
    vn2=vn2+k_linhas;

```

```

else
    k_linhas=0;
end

if(jmin<1)
    k_colunas=1-jmin;
    un1=un1+k_colunas;
    un2=un2+k_colunas;
else
    k_colunas=0;
end

In1=zeros(max(vn1),max(un1));
Ifn1=In1;
In2=zeros(max(vn2),max(un2));
Ifn2=In2;

for l=1:size(un1,1)
    In1(vn1(l,1),un1(l,1))=I1(v1(l,1),u1(l,1));
    Ifn1(vn1(l,1),un1(l,1))=1;
end

for l=1:size(un2,1)
    In2(vn2(l,1),un2(l,1))=I2(v2(l,1),u2(l,1));
    Ifn2(vn2(l,1),un2(l,1))=1;
end

```

B. 9 Programa de Correspondência

```

function [Ic,Jc1,Jc2] = corresp(If1,If2,dmin);

%[Ic,Jc1,Jc2,retas1] = corresp(If1,If2,dmin,r);
%
% Calcula os pontos correspondentes do par de imagens binarizadas If1
% e If2, constituídas pelas bordas das imagens estéreo retificadas
% (linhas epipolares horizontais).
%
% dmin é a mínima distância em pixels entre pontos de uma mesma linha
% de If1 e If2 para a qual a correspondência entre os mesmos deverá
% ser feita de forma automática, de acordo com a ordem em que aparecem
% nas duas imagens.
%
%Saídas:
% Ic é o vetor constituído pelos índices das linhas em que se
% encontram os pontos de uma imagem que possuem correspondentes na
% outra imagem.
% Jc1 é o vetor constituído pelos índices das colunas em que se
% encontram os pontos da imagem If1 que possuem correspondentes na
% imagem If2.

```



```

% Jc2 é o vetor constituído pelos índices das colunas em que se
% encontram os pontos da imagem If2 que possuem correspondentes na
% imagem If1.

linhas=min([size(If1,1) size(If2,1)]);
corresp=0; %número de correspondências já efetuadas

%Alocação de memória
Ic=zeros(size(If1,1)*size(If1,2),1);
Jc1=Ic;
Jc2=Ic;
z=Ic; %matriz de zeros;

for k=1:linhas
    A1=If1(k,:);
    A2=If2(k,:);

    %Verificando se a condição imposta por dmin é satisfeita
    s=1;

    I1=find(A1==1);
    J=I1(2:size(I1,2))-I1(1:(size(I1,2)-1));
    if(min(J)<dmin) s=0; end

    I2=find(A2==1);
    J=I2(2:size(I2,2))-I2(1:(size(I2,2)-1));
    if(min(J)<dmin) s=0; end

    %Determinação dos pontos correspondentes
    if((size(I1)==size(I2))&(s==1))
        Ic((corresp+1):(corresp+size(I1,2)),1)=z(1:size(I1,2),1)+k;
        Jc1((corresp+1):(corresp+size(I1,2)),1)=I1';
        Jc2((corresp+1):(corresp+size(I1,2)),1)=I2';
        corresp=corresp+size(I1,2);
    end
end

Ic=Ic(1:corresp,1);
Jc1=Jc1(1:corresp,1);
Jc2=Jc2(1:corresp,1);

```

```

function xyz_cena=demoxyz(I1,P1,I2,P2,k_linhas,k_colunas);

%xyz_cena=demoxyz(I1,P1,I2,P2,k_linhas,k_colunas);
%
% Realiza a reconstituição de pontos de uma cena filmada, a partir
% das imagens correspondentes no par estéreo I1, I2.
%
% P1,P2: matrizes de projeção perspectiva das câmaras geradoras das
% imagens estéreo I1 e I2, respectivamente.
%k_linhas e k_colunas: valores somados às posições de cada pixel das
% imagens retificadas calculadas, com o objetivo
% de evitar índices nulos ou negativos na
% representação matricial (estes valores são
% subtraídos dos índices i, j durante o processo
% de reconstituição da cena a partir das imagens
% estéreo).

```

```
if (nargin==4)
    k_linhas=0;
    k_colunas=0;
end

H=subplot(1,2,1);
imshow(I1);
subplot(1,2,2);
imshow(I2);

linha1=uicontrol('Style','text');
set(linha1,'Position',[10 10 530 80]);
set(linha1,'Units','normalized');

linha1=uicontrol('Style','text');
set(linha1,'Position',[15 62 520 23]);
set(linha1,'BackgroundColor',[1 1 1]);
set(linha1,'Units','normalized');

linha2=uicontrol('Style','text');
set(linha2,'Position',[15 39 520 23]);
set(linha2,'BackgroundColor',[1 1 1]);
set(linha2,'Units','normalized');

linha3=uicontrol('Style','text');
set(linha3,'Position',[15 16 520 23]);
set(linha3,'BackgroundColor',[1 1 1]);
set(linha3,'Units','normalized');

button=-1;
npontos=0; %número de pontos cujas coordenadas já foram calculadas
esquerda=(0==0);
while(button~=27)
    if(esquerda)
        [je,ie,button]=ginput(1);
        set(linha2,'String','');
        set(linha3,'String','');
        set(linha1,'String',sprintf('Coordenadas na imagem da esquerda:
i=%f    j=%f',ie,je));
    else
        [jd,id,button]=ginput(1);
        set(linha2,'String',sprintf('Coordenadas na imagem da direita:
i=%f    j=%f',id,jd));
        npontos=npontos+1;
        xyz_cena(npontos,1:3)=reconsti([ie je],P1,[id
jd],P2,k_linhas,k_colunas);
        set(linha3,'String',sprintf('Coordenadas reconstituídas da cena:
x=%f    y=%f
z=%f',xyz_cena(npontos,1),xyz_cena(npontos,2),xyz_cena(npontos,3)));
    end
    esquerda=~esquerda;
end

close all;
```

B. 10 Programa de Reconstituição

```

function xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)

%xyz_cena=reconsti(ij_imag1,P1,ij_imag2,P2,k_linhas,k_colunas)
%
%Efetua o cálculo das três coordenadas dos pontos de uma cena
%arbitrária, a partir das posições dos "pixels" correspondentes
%em duas imagens distintas da mesma. Cada linha em ij_imag1 e
%em ij_imag2 deve corresponder às coordenadas i e j do pixel
%de um dos pontos a serem reconstituídos. P1 e P2 são as matrizes
%de transformação perspectiva das câmeras que geraram cada uma
%das imagens. A matriz xyz_cena retornada pela função contém
%em cada linha as coordenadas x,y,z de um unico ponto.
%k_linhas e k_colunas: valores somados às posições de cada pixel das
imagens retificadas
%
%                               calculadas, com o objetivo de evitar índices
nulos ou negativos
%
%                               na representação matricial (estes valores
serão subtraídos
%
%                               dos índices i,j durante o processo de
reconstituição da cena a
%
%                               partir das imagens estéreo).

ij_imag1=[ij_imag1(:,1)-k_linhas ij_imag1(:,2)-k_colunas];
ij_imag2=[ij_imag2(:,1)-k_linhas ij_imag2(:,2)-k_colunas];

k=size(ij_imag1);

if((k(1,2)~=2)|(k(1,1)~=size(ij_imag2,1)))
    disp('Entradas invalidas');
else
    k=k(1,1);
    xyz_cena=zeros(k,3);
    for j=1:k
        A(1,1:3)=P1(1,1:3)-ij_imag1(j,2)*P1(3,1:3);
        A(2,1:3)=P1(2,1:3)-ij_imag1(j,1)*P1(3,1:3);
        A(3,1:3)=P2(1,1:3)-ij_imag2(j,2)*P2(3,1:3);
        A(4,1:3)=P2(2,1:3)-ij_imag2(j,1)*P2(3,1:3);

        y(1,1:1)=-P1(1,4)+ij_imag1(j,2)*P1(3,4);
        y(2,1:1)=-P1(2,4)+ij_imag1(j,1)*P1(3,4);
        y(3,1:1)=-P2(1,4)+ij_imag2(j,2)*P2(3,4);
        y(4,1:1)=-P2(2,4)+ij_imag2(j,1)*P2(3,4);

        xyz_cena(j,1:3)=((inv((A')*A))*A'*y)';
    end
end
end

```

```
function xyz_cena=recocena(I1,P1,I2,P2,thresh,dmin,tipo_filtro);

%xyz_cena=recocena(I1,P1,I2,P2,thresh,dmin);
%
% Reconstitui os pontos da cena filmada, a partir das imagens estéreo
% I1, I2.
%
% thresh: "threshold" aplicado a cada imagem durante a detecção de
% bordas, utilizando a máscara de Sobel.
%
% dmin: distância mínima entre dois pontos das bordas de cada imagem,
% na mesma horizontal, de forma que a correspondência dos mesmos
% seja ainda efetuada de forma automática.
%
% tipo_filtro: pode ser omitido ou assumir o valor "vertical" ou
% "horizontal; indica o tipo de borda que será enfatizado
% durante a aplicação do filtro de Sobel.w

disp('Cálculo das bordas das imagens estéreo...');
if(nargin<7)
    If1=edge(I1,'sobel',thresh);
    If2=edge(I2,'sobel',thresh);
else
    If1=edge(I1,'sobel',thresh,tipo_filtro);
    If2=edge(I2,'sobel',thresh,tipo_filtro);
end

disp('Retificação das imagens filtradas...');
[In1,Pn1,Ifn1,In2,Pn2,Ifn2,k_linhas,k_colunas]=retifica(I1,P1,If1,I2,P
2,If2);

disp('Cálculo das discrepâncias...');
[Ic,Jc1,Jc2] = corresp(Ifn1,Ifn2,dmin);

disp('Reconstituição dos pontos da cena filmada...');
xyz_cena=reconsti([Ic Jc1],Pn1,[Ic Jc2],Pn2,k_linhas,k_colunas);
x=xyz_cena(:,1);
y=xyz_cena(:,2);
z=xyz_cena(:,3);

I=min(find(Ic>min(Ic)));
k=1;
medinf=find(x<0);
medinf=x(medinf);
medinf=sum(medinf)/size(medinf,1);
medsup=find(x>0);
medsup=x(medsup);
medsup=sum(medsup)/size(medsup,1);
while(size(I,1)>0)
    X=x(k:(I-1));
    Y=y(k:(I-1));
    Z=z(k:(I-1));
    if((max(X)<medsup*4)&(min(X)>medinf*4))
        plot3(X,Y,Z, '.');
        hold on;
        plot3(X,Y,Z);
        hold on;
    end
end
```

```

    k=I;
    I=min(find(Ic>Ic(I)));
end
X=x(k:size(x,1));
Y=y(k:size(x,1));
Z=z(k:size(x,1));
if((max(X)<medsup*4)&(min(X)>medinf*4))
    plot3(X,Y,Z,'.');
    hold on;
    plot3(X,Y,Z);
    hold on;
end
plot3(medsup*4,medsup*4,0,'w');
plot3(medinf*4,medinf*4,0,'w');
axis('equal');

```

B. 11 Programa de Cálculo da Distância do Objeto

```

function [d1,d2]=distcam(xyz_cena,P1,P2);

%[d1,d2]=distcam(xyz_cena,P1,P2);
%
% Calcula a distância do objeto xyz_cena (reconstituído por recocena)
% aos centros focais das câmeras representadas pelas matrizes de
% transformação perspectiva P1 e P2.

%Cálculo das coordenadas dos centros focais
c1=-inv(P1(:,1:3))*P1(:,4);
c2=-inv(P2(:,1:3))*P2(:,4);

%Cálculo das distâncias
d1=c1'+zeros(size(xyz_cena));
d1=(d1-xyz_cena).^2;
d1=min(sqrt(sum(d1')));

d2=c2'+zeros(size(xyz_cena));
d2=(d2-xyz_cena).^2;
d2=min(sqrt(sum(d2')));

disp('Distância estimada do objeto ao centro focal da câmera 1:');
texto=sprintf('%d cm',d1/10);
disp(texto);

disp(' ');
disp('Distância estimada do objeto ao centro focal da câmera 2:');
texto=sprintf('%d cm',d2/10);
disp(texto);

```

B. 12 Programas Exemplo do Capítulo 6

```
function T=thresh(S,limiar)
%
% T=thresh(S,l)
%
% Esta função realiza o Threshold de uma imagem em níveis de cinza,
% S, e com limiar l

[x,y]=size(S);
T=zeros(x,y);
for i=1:x
    for j=1:y
        if S(i,j)>=limiar
            T(i,j)=1;
        end
    end
end
```

```
function s=filtro(I,M)

%
% s=filtro(I,M)
%
% Esta função realiza filtra a uma imagem S utilizando uma
máscara M
%
% obs: * A matriz M é uma matriz 3x3

[x,y]=size(I);

% O filtro não é definido nas bordas da matriz, logo não é aplica
na primeira
% e na ultima linha e na primeira e ultima coluna logo a matriz de
saída S terá
% duas linhas e duas colunas a menos I.

s=zeros(x-2,y-2);
for i=2:(x-1)
    for j=2:(y-1)

        % O centro da máscara se encontra em i,j da matriz I.Por ser
menor que I, a
        % coordenada de um ponto em s é igual a coordenada em I
reduzida da unidade

        aux=M.*I(i-1:i+1,j-1:j+1);
        s(i-1,j-
1)=aux(1,1)+aux(1,2)+aux(1,3)+aux(2,1)+aux(2,2)+aux(2,3)+aux(3,1)+a
ux(3,2)+aux(3,3);

    end
end
```

```
function f=dx(I)

%
% f=fx(I)
%
% Calcula a derivada parcial de I na direção x
%

[x,y]=size(I);
f=zeros(x,y-1);
for i=2:y % o operador dx não é definido na
primeira coluna
    f(:,i-1)=I(:,i)-I(:,i-1);
end

function f=dy(I)

%
% f=dy(I)
%
% Calcula a derivada parcial de I na direção y%

[x,y]=size(I);
f=zeros(x-1,y);
for i=1:(x-1) % o operador dx não é definido na
ultima linha
    f(i,:)=I(i,:)-I(i+1,:);
end

function f=grad0(S)

%
% f=grad0(S)
%
% Esta função aplica o operador GRADMAG0 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=max(abs(DX),abs(DY));
```

```
function f=grad1(S)

%
% f=grad1(S)
%
% Esta função aplica o operador GRADMAG1 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=abs(DX)+abs(DY);

function f=grad2(S)

%
% f=grad2(S)
%
% Esta função aplica o operador GRADMAG1 a matriz da imagem S
%

[x,y]=size(S);

% cálculo das diferenças

DX=dx(S);
DY=dy(S);

% adequação dos tamanhos das matrizes DX e DY

DX=[zeros(x,1) DX];
DY=[DY;zeros(1,y)];

% Cálculo da norma

f=((DX).^2+(DY).^2).^0.5;
```


Apêndice F

Cálculo das Matrizes Retificadoras $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$

Sejam $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ as matrizes de calibração do sistema estéreo hipotético em configuração canônica que retifica as imagens \mathbf{I}_1 e \mathbf{I}_2 geradas pelo sistema real $(\tilde{\mathbf{P}}_1, \tilde{\mathbf{P}}_2)$. Deseja-se determinar os elementos de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ de forma a garantir que pontos homólogos nas imagens retificadas estejam alinhados (mesma coordenada i) mas sem que sejam alterados os centros ópticos da configuração original, para que as imagens retificadas possam ser calculadas a partir de uma transformação linear conhecida de \mathbf{I}_1 e \mathbf{I}_2 .

Em [2], são propostas condições restritivas aos elementos de $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ que garantem não só o alinhamento dos pontos homólogos e a manutenção dos centros ópticos mas também a unicidade da solução. A seguir, demonstra-se como estas condições, citadas no capítulo 3, conduzem aos sistemas de equações 3.3.

Definindo

$$\tilde{\mathbf{P}}_{n1} = \left(\begin{array}{c|c} \mathbf{a}_1^T & a_{14} \\ \mathbf{a}_2^T & a_{24} \\ \mathbf{a}_3^T & a_{34} \end{array} \right) \text{ e } \tilde{\mathbf{P}}_{n2} = \left(\begin{array}{c|c} \mathbf{b}_1^T & b_{14} \\ \mathbf{b}_2^T & b_{24} \\ \mathbf{b}_3^T & b_{34} \end{array} \right) \quad (\text{C.20})$$

tem-se:

1. Os centros ópticos \mathbf{c}_{n1} e \mathbf{c}_{n2} das câmeras descritas por $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem coincidir, respectivamente, com os centros \mathbf{c}_1 e \mathbf{c}_2 das câmeras $\tilde{\mathbf{P}}_1$ e $\tilde{\mathbf{P}}_2$. Como, pela

equação do centro óptico, $\tilde{\mathbf{P}}_{n1} \cdot \begin{pmatrix} \mathbf{c}_{n1} \\ 1 \end{pmatrix} = \tilde{\mathbf{P}}_{n2} \cdot \begin{pmatrix} \mathbf{c}_{n2} \\ 1 \end{pmatrix} = 0$, segue:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_3^T \cdot \mathbf{c}_1 + a_{34} = 0 \\ \mathbf{b}_1^T \cdot \mathbf{c}_2 + b_{14} = 0 \\ \mathbf{b}_2^T \cdot \mathbf{c}_2 + b_{24} = 0 \\ \mathbf{b}_3^T \cdot \mathbf{c}_2 + b_{34} = 0 \end{cases} \quad (\text{C.21})$$

2. $\tilde{\mathbf{P}}_{n1}$ e $\tilde{\mathbf{P}}_{n2}$ devem apresentar o mesmo plano focal F . Como este plano é formado pelos pontos que não são projetados no plano de retina e portanto não possuem imagem, ou seja, $F = \{\mathbf{w} / \mathbf{a}_3^T \cdot \mathbf{w} + a_{34} = \mathbf{b}_3^T \cdot \mathbf{w} + b_{34} = 0\}$, tem-se:

$$\mathbf{a}_3 = \mathbf{b}_3 \text{ e } a_{34} = b_{34} \quad (\text{C.22})$$

3. Pontos homólogos nas imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} devem estar sempre na mesma linha (o que garante ainda que as novas linhas epipolares serão paralelas a um dos eixos das imagens). Já que a coordenada i da imagem de um ponto \mathbf{w} é definida por

$\frac{\mathbf{a}_2^T \cdot \mathbf{w} + a_{24}}{\mathbf{a}_3^T \cdot \mathbf{w} + a_{34}}$ em \mathbf{I}_{n1} e por $\frac{\mathbf{b}_2^T \cdot \mathbf{w} + b_{24}}{\mathbf{b}_3^T \cdot \mathbf{w} + b_{34}}$ em \mathbf{I}_{n2} , deve-se ter:

$$\frac{\mathbf{a}_2^T \cdot \mathbf{w} + a_{24}}{\mathbf{a}_3^T \cdot \mathbf{w} + a_{34}} = \frac{\mathbf{b}_2^T \cdot \mathbf{w} + b_{24}}{\mathbf{b}_3^T \cdot \mathbf{w} + b_{34}}, \forall \mathbf{w} \quad (\text{C.23})$$

Usando a condição imposta no item 2, chega-se a:

$$\mathbf{a}_2 = \mathbf{b}_2 \text{ e } a_{24} = b_{24} \quad (\text{C.24})$$

Neste ponto, observe-se que as matrizes retificadoras diferem apenas pela primeira linha.

4. As novas matrizes de calibração devem apresentar a normalização $\|\mathbf{a}_3\| = \|\mathbf{b}_3\| = 1$. Esta condição permitirá expressar os pontos principais das imagens retificadas e as resoluções horizontal e vertical em função de relações simples entre \mathbf{a}_1 , \mathbf{a}_2 , \mathbf{b}_1 e \mathbf{a}_3 , nos itens 7 e 8.

5. O plano focal F de \tilde{P}_{n1} e \tilde{P}_{n2} será paralelo à intersecção dos planos focais de \tilde{P}_1 e \tilde{P}_2 . Como \mathbf{a}_3 é o vetor unitário perpendicular a F , deve-se adotar:

$$\mathbf{a}_3^T \cdot (\mathbf{f}_1 \wedge \mathbf{f}_2) = 0 \quad (\text{C.25})$$

onde \mathbf{f}_1 e \mathbf{f}_2 são os vetores unitários perpendiculares aos planos focais de \tilde{P}_1 e \tilde{P}_2 .

6. Os planos projetados em $i=0$ e em $j=0$ nas duas imagens retificadas devem ser perpendiculares entre si (ortogonalidade dos sistemas de referência retificados). Estes planos são perpendiculares respectivamente a \mathbf{a}_1 e \mathbf{a}_2 , no caso da imagem \mathbf{I}_{n1} , e a \mathbf{b}_1 e \mathbf{b}_2 , no caso da imagem \mathbf{I}_{n2} . Assim:

$$\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \text{ e } \mathbf{b}_1^T \cdot \mathbf{b}_2 = 0 \quad (\text{C.26})$$

Considerando-se ainda a condição 3 ($\mathbf{a}_2 = \mathbf{b}_2$), obtêm-se:

$$\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \text{ e } \mathbf{b}_1^T \cdot \mathbf{a}_2 = 0 \quad (\text{C.27})$$

7. Os pontos principais i_0 e j_0 das imagens \mathbf{I}_{n1} e \mathbf{I}_{n2} devem ser nulos. Adotando a normalização $\|\mathbf{a}_3\| = \|\mathbf{b}_3\| = 1$ (condição 4), os pontos principais podem ser expressos como:

$$j_0 = \mathbf{a}_1^T \cdot \mathbf{a}_3 \text{ e } i_0 = \mathbf{a}_2^T \cdot \mathbf{a}_3, \text{ em } \mathbf{I}_{n1}, \quad (\text{C.28})$$

$$j_0 = \mathbf{b}_1^T \cdot \mathbf{b}_3 \text{ e } i_0 = \mathbf{b}_2^T \cdot \mathbf{b}_3, \text{ em } \mathbf{I}_{n2} \quad (\text{C.29})$$

Logo:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{a}_3 = 0 \\ \mathbf{a}_2^T \cdot \mathbf{a}_3 = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \end{cases} \quad (\text{C.30})$$

A condição $\mathbf{b}_2^T \cdot \mathbf{b}_3 = 0$ já é garantida em $\mathbf{a}_2^T \cdot \mathbf{a}_3 = 0$, já que $\mathbf{a}_2 = \mathbf{b}_2$ e $\mathbf{a}_3 = \mathbf{b}_3$.

8. As resoluções horizontal (α_u) e vertical (α_v) da câmera \tilde{P}_1 devem ser mantidas na nova configuração representada por \tilde{P}_{n1} e \tilde{P}_{n2} . Portanto:

$$\begin{cases} \|\mathbf{a}_1 \wedge \mathbf{a}_3\| = \alpha_u \\ \|\mathbf{a}_2 \wedge \mathbf{a}_3\| = \alpha_v \\ \|\mathbf{b}_1 \wedge \mathbf{a}_3\| = \alpha_u \end{cases} \quad (\text{C.31})$$

A condição $\|\mathbf{b}_2 \wedge \mathbf{b}_3\| = \alpha_v$ fica garantida por $\|\mathbf{a}_2 \wedge \mathbf{a}_3\| = \alpha_v$.

Adotando a identidade vetorial $\|\mathbf{x} \wedge \mathbf{y}\| = \|\mathbf{x}\|^2 \cdot \|\mathbf{y}\|^2 - (\mathbf{x}^T \cdot \mathbf{y})^2$, vem:

$$\begin{cases} \|\mathbf{a}_1\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{a}_1^T \cdot \mathbf{a}_3)^2 = \alpha_u^2 \\ \|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{a}_2^T \cdot \mathbf{a}_3)^2 = \alpha_v^2 \\ \|\mathbf{b}_1\|^2 \cdot \|\mathbf{a}_3\|^2 - (\mathbf{b}_1^T \cdot \mathbf{a}_3)^2 = \alpha_u^2 \end{cases} \quad (\text{C.32})$$

Pela condição 7, $\mathbf{a}_1^T \cdot \mathbf{a}_3 = \mathbf{a}_2^T \cdot \mathbf{a}_3 = \mathbf{b}_2^T \cdot \mathbf{a}_3 = 0$, vem:

$$\begin{cases} \|\mathbf{a}_1\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_u^2 \\ \|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_v^2 \\ \|\mathbf{b}_1\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_u^2 \end{cases} \quad (\text{C.33})$$

Finalmente, como $\|\mathbf{a}_3\| = 1$, tem-se:

$$\begin{cases} \|\mathbf{a}_1\| = \alpha_u \\ \|\mathbf{a}_2\| = \alpha_v \\ \|\mathbf{b}_1\| = \alpha_u \end{cases} \quad (\text{C.34})$$

As relações assim obtidas podem ser reagrupadas para que se obtenham os sistemas de equações 3.3. Inicialmente, considerem-se as equações 3 e 6 do sistema obtido no item 1, além da normalização $\|\mathbf{a}_3\| = 1$ e da condição 5; tem-se:

$$\begin{cases} \mathbf{a}_3^T \cdot \mathbf{c}_1 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot \mathbf{c}_2 + a_{34} = 0 \\ \mathbf{a}_3^T \cdot (\mathbf{f}_1 \wedge \mathbf{f}_2) = 0 \\ \|\mathbf{a}_3\| = 1 \end{cases} \quad (\text{C.35})$$

que é o primeiro dos sistemas 3.3.

O segundo sistema é obtido a partir das equações 2 e 5 do item 1, da equação $\mathbf{a}_2^T \cdot \mathbf{a}_3 = 0$ (item 7) e de $\|\mathbf{a}_2\|^2 \cdot \|\mathbf{a}_3\|^2 = \alpha_v^2$ (item 8):

$$\begin{cases} \mathbf{a}_2^T \cdot \mathbf{c}_1 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{c}_2 + a_{24} = 0 \\ \mathbf{a}_2^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_2\| = \alpha_v \end{cases} \quad (\text{C.36})$$

A equação 1 do item 1 e as relações $\mathbf{a}_1^T \cdot \mathbf{a}_2 = 0$, $\mathbf{a}_1^T \cdot \mathbf{a}_3 = 0$ e $\|\mathbf{a}_1\| = \alpha_u$ conduzem ao terceiro sistema:

$$\begin{cases} \mathbf{a}_1^T \cdot \mathbf{c}_1 + a_{14} = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{a}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{a}_1\| = \alpha_u \end{cases} \quad (\text{C.37})$$

Por fim, o quarto sistema é obtido da equação 4 do item 1 e das relações

$$\mathbf{b}_1^T \cdot \mathbf{a}_2 = 0, \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \text{ e } \|\mathbf{b}_1\| = \alpha_u :$$

$$\begin{cases} \mathbf{b}_1^T \cdot \mathbf{c}_2 + b_{14} = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_2 = 0 \\ \mathbf{b}_1^T \cdot \mathbf{a}_3 = 0 \\ \|\mathbf{b}_1\| = \alpha_u \end{cases} \quad (\text{C.38})$$

Apêndice A - Sensor

84

A.1 Sensores na Robótica 86

A.2 Sensores Internos 86

A.3 Sensores Externos 88

Apêndice B - Listagens dos Programas

90

B. 1 Programa de Calibração da Câmera CCD 90

B. 2 Programa de Retificação 94

B. 3 Programa de Correspondência 97

B. 4 Programa de Reconstituição 100

B. 5 Programa de Cálculo da Distância do Objeto 102

B. 6 Programas Exemplo do Capítulo 6 103

Apêndice C - Cálculo das Matrizes Retificadoras \tilde{P}_{n1} e \tilde{P}_{n2} 106

Figura A-1 : Sensor de posição de Mactarlane e Donath 87

Figura A-2 : Sensor de proximidade..... 88