
**SISTEMA DE VISÃO COMPUTACIONAL
PARA RASTREAMENTO DE TRAJETÓRIAS**

CRISTIANO JACQUES MIOSSO

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

SISTEMA DE VISÃO COMPUTACIONAL
PARA RASTREAMENTO DE TRAJETÓRIAS

CRISTIANO JACQUES MIOSSO

ORIENTADOR: ADOLFO BAUCHSPIESS

DISSERTAÇÃO DE MESTRADO EM
ENGENHARIA ELÉTRICA

PUBLICAÇÃO:

BRASÍLIA/DF, ABRIL/2002.

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**SISTEMA DE VISÃO COMPUTACIONAL
PARA RASTREAMENTO DE TRAJETÓRIAS**

CRISTIANO JACQUES MIOSSO

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO DEPARTAMENTO DE ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA UNIVERSIDADE DE BRASÍLIA, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.

APROVADA POR:

Prof. Dr.-Ing. Adolfo Bauchspiess
(Orientador)

Prof. Dr. Francisco Assis de O. Nascimento
(Examinador Interno)

Prof. José Maurício Motta, Ph.D.
(Examinador Externo)

Prof. Docteur d'Etat Liu Hsu
(Examinador Externo)

BRASÍLIA/DF, 24 DE ABRIL DE 2002.

FICHA CATALOGRÁFICA

MIOSSO, CRISTIANO JACQUES

Sistema de Visão Computacional para Rastreamento de Trajetórias

[Distrito Federal] 2002.

x, 140 p., 297mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2002).

Dissertação de Mestrado — Universidade de Brasília. Faculdade de Tecnologia. Departamento de Engenharia Elétrica.

- | | |
|-------------------------------------|--------------------------|
| 1. Visão Computacional | 2. Controle e Automação |
| 3. Processamento Digital de Imagens | 4. Sistemas Inteligentes |
| I. ENE/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

MIOSSO, C. J. (2002). Sistema de Visão Computacional para Rastreamento de Trajetórias. Dissertação de Mestrado, Publicação ENE.DM-001A/90, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 140 p.

CESSÃO DE DIREITOS

NOME DO AUTOR: Cristiano Jacques Miosso

TÍTULO DA DISSERTAÇÃO DE MESTRADO: Sistema de Visão Computacional para Rastreamento de Trajetórias

GRAU / ANO: Mestre / 2002

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação de mestrado pode ser reproduzida sem a autorização por escrito do autor.

Cristiano Jacques Miosso

SQS 108 bloco E apartamento 506

70347/050 — Brasília/DF — Brasil

*Não deixaremos de explorar,
e o fim da nossa exploração
será chegar onde começamos
e conhecer o local pela primeira vez...
quando as labaredas se dobram
sobre a coroa das chamas
e a chama e a rosa são uma só.*

T. S. Eliot

Resumo

Neste trabalho, foi implementado um sistema de visão computacional para rastreamento de trajetórias por manipuladores robóticos. Foram investigadas técnicas de processamento de imagens e análise de cenas que, a partir da informação visual extraída da trajetória-alvo ao longo do processo, permitem determinar as referências de entrada do sistema de posicionamento.

Os problemas envolvidos nas várias etapas do processo foram abordados. Em uma etapa de pré-processamento, cada imagem de entrada é inicialmente compensada quanto à influência de condições não-ideais de iluminação, de forma a se obter informação de reflectância a partir da informação de luminância disponível. Uma contribuição deste trabalho é a proposta de um algoritmo para o cálculo automático de alguns parâmetros da imagem compensada, que permitem descrevê-la em termos de um modelo estatístico adequado e, em seguida, determinar quais *pixels* representam pontos da trajetória. Em testes específicos, o método apresentou bom desempenho durante a binarização de imagens reais e sintéticas, neste caso permitindo o cálculo dos parâmetros com erros inferiores a 3% em relação aos utilizados na geração das imagens.

A etapa seguinte no rastreamento da trajetória diz respeito à definição de uma relação de ordem entre os *pixels* da imagem binarizada. A seqüência de *pixels* então obtida permite determinar os deslocamentos necessários do efetuador para que ele ocupe sucessivamente os pontos especificados, a partir de uma relação linear estabelecida durante a calibração do sistema óptico. Finalmente, as coordenadas dos pontos-alvos são ajustadas em um processo de interpolação, cujo objetivo é garantir uma distância constante entre pontos consecutivos na nova seqüência gerada e, conseqüentemente, uma velocidade uniforme do efetuador ao ser enviado a intervalos regulares de tempo às posições especificadas. Só depois de percorridas estas posições é que uma nova imagem da trajetória é adquirida, dando início a um novo ciclo de processamento.

Para a avaliação de desempenho dos algoritmos implementados, em situações práticas de rastreamento, utilizou-se um *plotter* com dois graus de liberdade guiado por visão computacional. O sistema completo foi avaliado para diversas trajetórias de entrada, sendo então analisados os resultados de todos os processos intermediários do rastreamento e o desempenho global do sistema, em termos dos erros medidos entre as trajetórias efetivas do efetuador e as trajetórias de entrada.

Os resultados observados mostraram que o sistema implementado permite o rastreamento de trajetórias a partir apenas da informação extraída, pelos algoritmos de visão, das imagens adquiridas durante a movimentação do efetuador, como fora objetivado. A precisão alcançada foi bastante aceitável diante da resolução do *plotter* empregado, com erros de posição que atingiram valores máximos da ordem de $3mm$ mas que, em geral, foram bem menores, de acordo com a análise do erro médio quadrático e do desvio padrão associado aos erros, avaliados para algumas trajetórias analíticas de entrada. No estabelecimento desta precisão, os algoritmos de visão não foram os fatores limitantes, já que deslocamentos bastante inferiores — de menos de $0,15mm$ — são detectáveis por aqueles algoritmos para o sistema óptico empregado, como se infere de sua matriz de calibração obtida, e já que o sistema permitiu corrigir os erros sempre que eles ultrapassaram valores da ordem da resolução do *plotter*.

Abstract

In this research, a computer vision system was implemented for automatic path tracking by robotic manipulators. Several image processing and scene analysis techniques were investigated which together allow position references to the positioning system to be computed based solely on visual information extracted from the desired path during the process.

Problems which arise in each processing stage were specifically treated. First, each image is processed so that non-uniform lighting during its formation can be properly compensated, and information about reflectance in the scene obtained from image brightness measurements. A contribution in this research is an algorithm proposed for automatically computing all parameters necessary to describe the compensated image in terms of an appropriate statistical model, and then determining which pixels represent positions in the desired path. In the evaluation experiments carried on, the method performed well when applied to extracting binary images from multi-level real and synthetic ones, in this case allowing all parameters to be computed with errors lower than 3% with respect to those employed when generating the images.

The following stage involves establishing an order relationship between pixels in the resulting binary image. The sequence of pixels thus obtained is then employed in computing the displacements necessary to the end-effector to move in sequence to each position in the desired path; this conversion from image coordinates to workspace coordinates is based on a linear transformation determined by a calibration procedure. Finally, all targets coordinates are adjusted by an interpolation technique, which results in a uniform distance between neighbor positions in the new generated sequence, and consequently a uniform velocity when the end-effector is sent in a regular time basis to the specified positions. Only after the end effector run through all these positions, a new image is acquired, starting a new control cycle.

In order to the algorithms performance to be evaluated in real path-tracking conditions, a plotter with two freedoms and guided by computer vision was employed. The complete system was evaluated for several input paths, the results of each processing stage being analysed separately and the global performance being evaluated based on the differences between the real paths followed by the end-effector and the desired ones, in the system input.

The results obtained showed that the proposed system allows path tracking by the plotter based solely on information extracted by the vision algorithms from images acquired when the end-effector is moving, as it had been planned. Attained precision was considered very reasonable given the employed plotter resolution, with errors in position of about $3mm$ in the worst cases, but which were, in general, much lower, according to the analysis of the rms and the standard deviation associated to errors, as evaluated for some analytical input paths. The vision algorithms were not the critical factors in determining this precision, since differences in position much lower than that — of about $0.15mm$ — can properly detected by them, as it can be concluded from the calibration matrix computed, and since the system caused the errors to be reduced as soon as they assumed values greater than the lowest displacement allowed by the plotter.

ÍNDICE

Capítulo	Página
1 INTRODUÇÃO	1
1.1 OBJETIVO DO TRABALHO	1
1.2 REVISÃO BIBLIOGRÁFICA	3
1.3 ESTRUTURA DA DISSERTAÇÃO	6
2 DESCRIÇÃO GENÉRICA DO SISTEMA DE RASTREAMENTO	9
2.1 EQUIPAMENTO UTILIZADO PARA AVALIAÇÃO DOS ALGORITMOS DE VISÃO COMPUTACIONAL PROPOSTOS	9
2.2 DIAGRAMA DE BLOCOS DO SISTEMA DE VISÃO PROPOSTO	11
3 BINARIZAÇÃO DE IMAGENS DE TRAJETÓRIAS	15
3.1 OBTENÇÃO DA RELAÇÃO ENTRE A LUMINÂNCIA NA IMAGEM E A REFLECTÂNCIA DOS OBJETOS QUE COMPÕEM A CENA OB- SERVADA	16
3.1.1 Relação entre a Potência Radiada pelos Objetos da Cena e a Função de Distribuição Bidirecional de Reflectância (FDBR)	16
3.1.2 Relação entre a Radiação a partir dos Objetos da Cena e a Irradiação no Plano de Imagem	19
3.1.3 Relação entre a Irradiação nos Elementos Fotossensíveis do Plano de Imagem e a Luminância de cada <i>Pixel</i>	21
3.1.4 Relação entre a Luminância de cada <i>Pixel</i> e a FDBR dos Objetos que Compõem a Cena Observada	24
3.2 MODELAMENTO ESTATÍSTICO DE IMAGENS DE TRAJETÓRIAS SOBRE FUNDOS HOMOGÊNEOS	25
3.3 CLASSIFICAÇÃO DE IMAGENS PELO PROCESSO DE LIMARIZAÇÃO	29
3.4 LIMAR ÓTIMO DE BINARIZAÇÃO	31
3.4.1 Probabilidade de classificação incorreta de um <i>pixel</i> , em função do limiar de binarização adotado	31

3.4.2	Escolha do limiar de binarização que minimiza a probabilidade de classificação incorreta de cada <i>pixel</i>	33
3.5	CONSIDERAÇÕES DE ORDEM PRÁTICA SOBRE O PROCESSO DE BINARIZAÇÃO	35
4	DETERMINAÇÃO DOS PARÂMETROS ESTATÍSTICOS DAS DISTRIBUIÇÕES GAUSSIANAS DA TRAJETÓRIA E DO FUNDO PARA CÁLCULO DO LIMIAR ÓTIMO DE BINARIZAÇÃO	43
4.1	DESCRIÇÃO DO PROBLEMA	43
4.2	DESCRIÇÃO DO ALGORITMO PROPOSTO	44
4.2.1	Equacionamento	45
4.2.2	Particionamento da imagem	47
4.2.3	Primeira etapa	47
4.2.4	Segunda etapa	52
4.3	RESULTADOS DE TESTES DE AVALIAÇÃO DO ALGORITMO PROPOSTO	53
5	PRÉ-PROCESSAMENTO DAS IMAGENS DE ENTRADA	61
5.1	INTRODUÇÃO	61
5.2	PRIMEIRA TÉCNICA DE COMPENSAÇÃO DA NÃO-UNIFORMIDADE DA ILUMINAÇÃO	63
5.3	SEGUNDA TÉCNICA DE COMPENSAÇÃO DA NÃO-UNIFORMIDADE DA ILUMINAÇÃO	67
5.4	ESCOLHA DO MÉTODO DE PRÉ-PROCESSAMENTO DAS IMAGENS UTILIZADO	68
6	EXTRAÇÃO DA TRAJETÓRIA A PARTIR DA IMAGEM BINARIZADA	71
6.1	O ALGORITMO DE BUSCA CIRCULAR	73
6.2	CONSIDERAÇÕES SOBRE DETALHES DE IMPLEMENTAÇÃO DO ALGORITMO DE BUSCA CIRCULAR	74
7	MÓDULO DE CONTROLE DA POSIÇÃO DO TCP	77
7.1	DETERMINAÇÃO DA RELAÇÃO ENTRE O DESLOCAMENTO DO ATUADOR E A VARIAÇÃO DAS COORDENADAS DA IMAGEM DE UM PONTO FIXO NA TRAJETÓRIA	77
7.1.1	Expressão matemática para a matriz de calibração em função de parâmetros do sistema óptico	78
7.1.2	Calibração do sistema óptico	83

7.2	CÁLCULO DAS POSIÇÕES DESEJADAS DO TCP CORRESPONDENTES AOS <i>PIXELS</i> EXTRAÍDOS NA BUSCA CIRCULAR	85
7.3	CÁLCULO DAS REFERÊNCIAS DE ENTRADA DO SISTEMA DE POSICIONAMENTO UNIFORMEMENTE ESPAÇADAS	86
7.4	DETERMINAÇÃO DO NÚMERO DE POSIÇÕES DE REFERÊNCIA EXTRAÍDAS DE CADA IMAGEM	87
8	RESULTADOS EXPERIMENTAIS	89
8.1	APRESENTAÇÃO DE ALGUMAS TRAJETÓRIAS DE ENTRADA EMPREGADAS	89
8.2	RESULTADOS DE PRÉ-PROCESSAMENTO DAS IMAGENS	91
8.3	RESULTADOS DE BINARIZAÇÃO DAS IMAGENS PROCESSADAS	94
8.3.1	Valores obtidos para os parâmetros estatísticos das imagens processadas	94
8.3.2	Resultados do processo de limiarização	94
8.4	<i>PIXELS</i> EXTRAÍDOS DAS IMAGENS PELO ALGORITMO DE BUSCA CIRCULAR	96
8.5	REFERÊNCIAS DE ENTRADA DO SISTEMA DE POSICIONAMENTO OBTIDAS	98
8.6	TRAJETÓRIAS DO TCP OBTIDAS	101
9	CONCLUSÕES	117
A	APARATO EXPERIMENTAL: SISTEMA DE CONTROLE DE POSIÇÃO E DE AQUISIÇÃO DE IMAGENS	123
A.1	INTERFACE ENTRE A PORTA PARALELA DO PC E O <i>PLOTTER</i>	123
A.1.1	Circuito de Interface	123
A.1.2	Rotinas de Interface	126
A.2	ROTINAS DE AQUISIÇÃO DAS IMAGENS DE TRAJETÓRIAS	130
A.2.1	Descrição das Rotinas de Aquisição de Imagens Implementadas	130
A.2.2	Determinação da Taxa de Aquisição de Imagens Obtida	132
B	LISTAGENS DOS PROGRAMAS DE INTERFACE COM O APARATO EXPERIMENTAL	135
B.1	ROTINAS DE INTERFACE ENTRE A PORTA PARALELA DO PC E O <i>PLOTTER</i>	135
B.1.1	Rotina de Ajuste da Coordenada x do <i>Plotter</i>	135
B.1.2	Rotina de Ajuste da Coordenada y do <i>Plotter</i>	138
B.1.3	Rotina de Ajuste Simultâneo das Coordenadas x e y do <i>Plotter</i>	141
B.2	ROTINAS DE AQUISIÇÃO DE IMAGENS	143

B.2.1	Rotina de Captura de Imagens Sucessivas em um Mesmo Arquivo Utilizando a API do Windows — imcap.c	143
B.2.2	Rotina de Leitura pelo MatLab das imagens capturadas — imcap.m	146
C	INTERPOLAÇÃO DAS COORDENADAS DA TRAJETÓRIA POR UMA FUNÇÃO POLINOMIAL DA DISTÂNCIA PERCORRIDA	149
D	RESULTADOS INTERMEDIÁRIOS OBTIDOS DURANTE O RAS- TREAMENTO	151
D.1	PRIMEIRA TRAJETÓRIA	151
D.2	TERCEIRA TRAJETÓRIA	158
D.3	QUARTA TRAJETÓRIA	164

ÍNDICE DAS TABELAS

Tabela	Página
4.1 Comparação entre os parâmetros teóricos utilizados na geração da imagem sintética 1 (figura 4.5) e os calculados.	56
4.2 Comparação entre os parâmetros teóricos utilizados na geração da imagem sintética 2 (figura 4.6) e os calculados.	56
8.1 Parâmetros estatísticos referentes à imagem da figura 8.4-(a), utilizados no cálculo do limiar ótimo de binarização.	95
8.2 Coordenadas dos <i>pixels</i> extraídos da imagem da figura 8.6 pela aplicação do algoritmo de busca circular (figura 8.7).	97
8.3 Entradas do módulo de controle da posição do atuador obtidas da figura 8.7, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela 8.2) em variações na posição do atuador (transformação dada pela equação (7.34)).	99
8.4 Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (8.2) e (8.3) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	100
A.1 Associação entre os pinos da porta paralela do PC e os endereços pelos quais eles são acessados.	126
D.1 Parâmetros estatísticos referentes à imagem da figura D.3-(a), utilizados no cálculo do limiar ótimo de binarização.	152
D.2 Coordenadas dos <i>pixels</i> extraídos da imagem da figura D.5 pela aplicação do algoritmo de busca circular (figura D.6).	155
D.3 Entradas do módulo de controle da posição do atuador obtidas da figura D.6, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.2) em variações na posição do atuador (transformação dada pela equação (7.34)).	156

D.4	Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.1) e (D.2) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	157
D.5	Parâmetros estatísticos referentes à imagem da figura D.11-(a), utilizados no cálculo do limiar ótimo de binarização.	158
D.6	Coordenadas dos <i>pixels</i> extraídos da imagem da figura D.13 pela aplicação do algoritmo de busca circular (figura D.14).	161
D.7	Entradas do módulo de controle da posição do atuador obtidas da figura D.14, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.6) em variações na posição do atuador (transformação dada pela equação (7.34)).	162
D.8	Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.3) e (D.4) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	163
D.9	Parâmetros estatísticos referentes à imagem da figura D.19-(a), utilizados no cálculo do limiar ótimo de binarização.	164
D.10	Coordenadas dos <i>pixels</i> extraídos da imagem da figura D.21 pela aplicação do algoritmo de busca circular (figura D.22).	167
D.11	Entradas do módulo de controle da posição do atuador obtidas da figura D.22, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.10) em variações na posição do atuador (transformação dada pela equação (7.34)).	168
D.12	Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.5) e (D.6) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	169

ÍNDICE DAS FIGURAS

Figura	Página	
2.1	<i>Plotter</i> com 2 graus de liberdade utilizado para rastreamento das trajetórias, no sistema de visão computacional implementado.	10
2.2	Disposição da câmera e da lâmpada sobre o TCP do <i>plotter</i> para aquisição das imagens da trajetória a ser rastreada, em um plano paralelo ao espaço de trabalho.	11
2.3	Ligação do <i>plotter</i> e da câmera ao PC com as rotinas de visão computacional e de posicionamento do TCP.	12
2.4	Diagrama de blocos do sistema de acionamento do <i>plotter</i> a partir da porta paralela do PC.	12
2.5	Utilização do módulo de visão computacional para o cálculo das referências de entrada do sistema de acionamento do <i>plotter</i>	13
2.6	Diagrama de blocos do sistema completo de rastreamento de trajetórias proposto.	14
3.1	Sistemas de coordenadas esféricas $R - \theta - \phi$ local ao ponto P de uma superfície, no qual se avalia a potência irradiada por unidade de área durante a formação de uma imagem.	17
3.2	Gráfico de um exemplo de função densidade de probabilidade associada ao nível de cinza de um <i>pixel</i> arbitrário, na imagem de uma trajetória sobre um fundo homogêneo (densidade de mistura). O gráfico foi sintetizado a partir da equação (3.51), considerando que a média e a variância da distribuição de cada <i>pixel</i> da trajetória são $\mu_1 = 0,16$ e $\sigma_1^2 = 0,0002$, que os valores correspondentes da distribuição de cada <i>pixel</i> do fundo são $\mu_2 = 0,23$ e $\sigma_2^2 = 0,0002$, e que a proporção de <i>pixels</i> pertencentes à trajetória para o número total de <i>pixels</i> é de $p_1 = 0,3$	27
3.3	(a) Imagem real de uma trajetória sobre um fundo uniforme, após aplicação de uma técnica de compensação da iluminação não-uniforme e (b) histograma correspondente.	28

3.4	(a) Imagem sintética de uma trajetória sobre um fundo, ambos de reflectâncias constantes, suposta adquirida sob iluminação perfeitamente uniforme. (b) Padrão de iluminação obtido pela colocação de uma fonte que ilumina uma superfície plana de reflectância constante a partir da direita para a esquerda. (c) Imagem obtida da superfície em (a) sob a condição de iluminação representada em (b).	29
3.5	(a) Histograma da imagem na figura 3.4-(a), de uma trajetória sobre um fundo suposta adquirida sob iluminação perfeitamente uniforme. (b) Histograma da imagem na figura 3.4-(c), de uma trajetória sobre um fundo adquirida sob a condição de iluminação representada em 3.4-(b).	30
3.6	(a) e (c) Imagens de duas trajetórias de espessuras distintas, ou ainda de uma mesma trajetória em planos a diferentes distâncias do plano de retina. À exceção da da proporção p_T de <i>pixels</i> correspondentes à trajetória para o número total de <i>pixels</i> , os parâmetros estatísticos das imagens são idênticos: $\mu_1 = 0,5$, $\mu_2 = 0,8$, $\sigma_1^2 = 1,45 \cdot 10^{-3}$, $\sigma_2^2 = 1,45 \cdot 10^{-3}$. (b) e (d) Histogramas das duas imagens, onde se observa a variação entre os dois casos da proporção p_T (0,0166 para a imagem a, 0,6247 para a imagem c).	36
3.7	(a) Histograma da imagem 3.6-(a), com destaque para o limiar de binarização $L = 0,6303$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado o limiar apropriado, todos os <i>pixels</i> foram classificados corretamente.	37
3.8	(a) Histograma da imagem 3.6-(c), com destaque para o limiar de binarização $L = 0,6303$ obtido para a imagem anterior — 3.6-(a). (b) Imagem binarizada utilizando o limiar destacado. Como não foi utilizado o limiar apropriado para a proporção de trajetória para fundo na imagem (p_T), diversos <i>pixels</i> foram classificados incorretamente.	37
3.9	(a) Histograma da imagem 3.6-(c), com destaque para o limiar de binarização $L = 0,6525$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado desta vez o limiar apropriado, todos os <i>pixels</i> foram classificados corretamente.	38
3.10	(a) e (c) Imagens de dois segmentos de comprimentos distintos de uma mesma trajetória. À exceção da da proporção p_T de <i>pixels</i> correspondentes à trajetória para o número total de <i>pixels</i> , os parâmetros estatísticos das imagens são idênticos: $\mu_1 = 0,5$, $\mu_2 = 0,8$, $\sigma_1^2 = 1,5 \cdot 10^{-3}$, $\sigma_2^2 = 1,5 \cdot 10^{-3}$. (b) e (d) Histogramas das duas imagens, onde se observa a variação de um segmento para o outro da proporção p_T (0,0107 para a imagem a, 0,5334 para a imagem c).	39

3.11	(a) Histograma da imagem 3.10-(a), com destaque para o limiar de binarização $L = 0,6274$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado o limiar apropriado, todos os <i>pixels</i> foram classificados corretamente.	40
3.12	(a) Histograma da imagem 3.10-(c), com destaque para o limiar de binarização $L = 0,6274$ obtido para a imagem anterior — 3.10-(a). (b) Imagem binarizada utilizando o limiar destacado. Como não foi utilizado o limiar apropriado para a proporção de trajetória para fundo na imagem (p_T), diversos <i>pixels</i> foram classificados incorretamente.	40
3.13	(a) Histograma da imagem 3.10-(c), com destaque para o limiar de binarização $L = 0,6506$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado desta vez o limiar apropriado, todos os <i>pixels</i> foram classificados corretamente.	41
4.1	Divisão da imagem de entrada (a) em regiões retangulares contíguas sem sobreposição (b), para cálculo das médias e variâncias (c) que serão utilizadas na determinação dos parâmetros estatísticos das distribuições de probabilidade do objeto e do fundo.	47
4.2	Histograma de uma imagem de um objeto sobre um fundo, como destaque para a estimativa inicial do limiar fornecida pelo método descrito na seção 4.2.3 (linha grossa contínua), ainda distinta do limiar ótimo (linha grossa seccionada).	52
4.3	Segunda etapa do algoritmo proposto para cálculo automático dos parâmetros estatísticos das distribuições associadas à trajetória e ao fundo. Na seqüência de histogramas, observa-se que a cada iteração é obtida uma estimativa mais precisa do limiar ótimo de binarização.	53
4.4	Diagrama de blocos do algoritmo proposto para cálculo automático dos parâmetros estatísticos das distribuições associadas à trajetória e ao fundo, para determinação do limiar ótimo de binarização L	54
4.5	(a) Exemplo de uma imagem sintética com 2 regiões de reflectância afetadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar de binarização calculado. (c) Imagem binarizada a partir do limiar calculado.	57
4.6	(a) Exemplo de uma imagem sintética com 2 regiões de reflectância afetadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar de binarização calculado. (c) Imagem binarizada a partir do limiar calculado.	57

4.7	(a) Exemplo de uma imagem real de uma trajetória sobre um fundo. (b) Histograma correspondente, com destaque para o limiar calculado. (c) Imagem binarizada a partir do limiar calculado.	58
4.8	(a) Exemplo de uma imagem sintética com 2 regiões de reflectância altamente degradadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar teórico; observa-se a sobreposição das duas distribuições. (c) Imagem binarizada a partir do limiar teórico ótimo.	58
4.9	(a) Resultado da aplicação do filtro de Wiener à imagem 4.8. (b) Histograma correspondente, com destaque para o limiar calculado. (c) Imagem binarizada a partir do limiar calculado.	59
5.1	Diagrama de blocos que descreve o algoritmo <i>retinex</i>	65
5.2	Diagrama de blocos que descreve a primeira técnica avaliada de compensação da não-uniformidade da iluminação durante a formação das imagens de trajetórias.	67
6.1	Descrição do algoritmo de busca circular, com a direção de varredura das circunferências em sentido anti-horário.	73
7.1	Sistema óptico utilizado durante a formação de imagens de pontos em um plano \mathcal{T} paralelo ao plano de retina \mathcal{I}	79
7.2	Visão unidimensional dos eixos j' e \hat{y} do sistema óptico da figura 7.1, com destaque para a transformação perspectiva durante a formação da imagem A' de um ponto A	80
7.3	Relação genérica entre a orientação e a posição de dois sistemas de coordenadas no plano da trajetória, um arbitrário $(x'y')$ e o outro com os eixos paralelos aos do plano de retina $(\hat{x}\hat{y})$	81
8.1	Imagens digitalizadas de quatro trajetórias submetidas ao sistema de rastreamento: (a) circunferência com $60,5mm$ de raio; (b) curva fechada com ondulações; (c) espiral; (d) retângulo com $152mm$ e $100mm$ de lados.	90
8.2	(a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.1-(b)). (b) Histograma correspondente.	92
8.3	(a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura 8.2-(a). (b) Imagem obtida após a compensação.	93
8.4	(a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura 8.2-(a). (b) Histograma correspondente.	93
8.5	Histograma associado à imagem da figura 8.4-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela 8.1).	95

8.6	Imagem da figura 8.4-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura 8.5.	95
8.7	Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura 8.6 pela aplicação do algoritmo de busca circular.	96
8.8	Entradas do módulo de controle da posição do TCP obtidas da figura 8.7, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela 8.2) em variações na posição do TCP (transformação dada pela equação (7.34)).	98
8.9	Entradas do módulo de controle da posição do TCP uniformemente espaçadas, obtidas pela aplicação dos polinômios (8.2) e (8.3) a distâncias d ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	99
8.10	Referências de entrada do sistema de posicionamento do TCP obtidas para uma das trajetórias rastreadas (figura 8.1-(a)), a partir de 84 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.	102
8.11	Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(b)), a partir de 72 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.	103
8.12	Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(c)), a partir de 60 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.	103
8.13	Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(d)), a partir de 114 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.	104
8.14	(a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.	105
8.15	(a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.	106
8.16	(a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.	106

8.17	(a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.	107
8.18	Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.14-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.14-(b)).	108
8.19	Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.15-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.15-(b)).	109
8.20	Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.16-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.16-(b)).	110
8.21	Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.17-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.17-(b)).	111
8.22	Erros avaliados entre as posições de referência calculadas pelo sistema de visão e a trajetória da figura 8.14-(a), durante o rastreamento. O valor médio destes erros é de $0,0046mm$, e o desvio-padrão, de $0,6521mm$. O erro raiz média quadrática é de $0,6517mm$	113
8.23	Erros avaliados entre as posições de referência calculadas pelo sistema de visão e a trajetória da figura 8.17-(a), durante o rastreamento. O valor médio destes erros é de $0,0846mm$, e o desvio-padrão, de $0,8685mm$. O erro raiz média quadrática é de $0,8722mm$	114
A.1	Diagrama de blocos que descreve o circuito de interface entre a porta paralela do PC e o <i>plotter</i> . O atributo L indica que o pino correspondente atua com lógica invertida, ou seja, a escrita dos valores 0 e 1 leva a saída a $5V$ e a $0V$, respectivamente.	124
A.2	Configuração do conversor DAC0800 para a faixa de operação de $-12V$ a $12V$ ($24V_{pp}$). MSB e LSB referem-se aos <i>bits</i> mais e menos significativos da palavra de entrada, respectivamente.	126
A.3	Configuração utilizada dos conversores associados aos canais x (a) e y (b) do <i>plotter</i> , para operação nas faixas de $-5,63V$ a $5,63V$ e $-3,83V$ a $3,82V$, respectivamente.	127
A.4	Circuito completo de interface entre a porta paralela e o <i>plotter</i>	128
D.1	(a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.14-(a)). (b) Histograma correspondente.	151

D.2	(a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.1-(a). (b) Imagem obtida após a compensação.	152
D.3	(a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.1-(a). (b) Histograma correspondente. .	153
D.4	Histograma associado à imagem da figura D.3-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.1).	153
D.5	Imagem da figura D.3-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.4.	154
D.6	Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.5 pela aplicação do algoritmo de busca circular.	154
D.7	Entradas do módulo de controle da posição do atuador obtidas da figura D.6, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.2) em variações na posição do atuador (transformação dada pela equação (7.34)).	155
D.8	Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.1) e (D.2) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	156
D.9	(a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.16-(a)). (b) Histograma correspondente.	158
D.10	(a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.9-(a). (b) Imagem obtida após a compensação.	159
D.11	(a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.9-(a). (b) Histograma correspondente. .	159
D.12	Histograma associado à imagem da figura D.11-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.5).	160
D.13	Imagem da figura D.11-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.12.	160
D.14	Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.13 pela aplicação do algoritmo de busca circular.	161
D.15	Entradas do módulo de controle da posição do atuador obtidas da figura D.14, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.6) em variações na posição do atuador (transformação dada pela equação (7.34)).	162

D.16 Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.3) e (D.4) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	163
D.17 (a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.17-(a)). (b) Histograma correspondente.	164
D.18 (a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.17-(a). (b) Imagem obtida após a compensação.	165
D.19 (a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.17-(a). (b) Histograma correspondente.	165
D.20 Histograma associado à imagem da figura D.19-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.9).	166
D.21 Imagem da figura D.19-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.20.	166
D.22 Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.21 pela aplicação do algoritmo de busca circular.	167
D.23 Entradas do módulo de controle da posição do atuador obtidas da figura D.22, pela conversão das distâncias do centro da imagem aos <i>pixels</i> extraídos (tabela D.10) em variações na posição do atuador (transformação dada pela equação (7.34)).	168
D.24 Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.5) e (D.6) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).	169

Capítulo 1

INTRODUÇÃO

Sistemas de visão computacional vêm sendo amplamente utilizados na guiagem de manipuladores robóticos e robôs móveis. Trata-se, por exemplo, de determinar, a partir das imagens captadas de um ambiente, as coordenadas de uma trajetória a ser seguida, de objetos a serem manipulados ou de obstáculos presentes, para que se evitem colisões. O objetivo pode ser ainda corrigir ou minimizar o erro na posição que um robô ou manipulador assume a cada instante, pela utilização da informação visual na retroação do sinal de saída de um sistema de posicionamento.

Avanços na área têm ocorrido sobretudo em aplicações industriais e em ambientes de laboratório, em que as tarefas a serem realizadas pelos robôs são rigidamente definidas e o processo de visão pode ser consideravelmente simplificado pelo controle das condições de iluminação e pela utilização exclusiva de formas bem estruturadas e conhecidas. Na indústria, uma abordagem comum é o emprego de imagens binárias, pela maior facilidade de manipulação e de armazenamento; a análise é, no entanto, restrita a objetos e formas cujos graus de liberdade se restringem a um plano paralelo ao plano de retina da câmera [13], sendo mais de dois níveis de luminância necessários à reconstituição de formas tridimensionais, independentemente da técnica utilizada.

O rastreamento de trajetórias é uma aplicação comum da visão computacional para controle de robôs móveis e manipuladores. Objetiva-se que, partindo da informação visual, o robô ou efetuador siga uma seqüência de posições a ser especificada durante o processo, seja por um alvo móvel a ser rastreado ponto a ponto, seja por uma curva de referência apresentada ao sistema de rastreamento. Esta pesquisa aborda alguns aspectos do segundo problema.

1.1 OBJETIVO DO TRABALHO

Neste trabalho, é avaliado um sistema de visão computacional para o rastreamento de trajetórias por manipuladores robóticos. A avaliação dos algoritmos implementados é feita em situações práticas de rastreamento: uma câmera, fixa ao efetuador de um dispositivo

para traçados gráficos, adquire imagens da trajetória a ser rastreada e as envia a um computador para processamento. A execução das rotinas de visão permite então extrair das imagens as referências de posição do efetuador, enviadas em seguida a um sistema de posicionamento situado entre o computador e o manipulador. Além disso, por se deslocar a câmera junto com o efetuador, ela permite também medir, a cada imagem adquirida, a diferença entre sua posição efetiva e a posição especificada pela curva que representa o trajeto desejado. Assim, o sistema de visão é utilizado não só para gerar as referências de posição mas também para medir o erro de posicionamento, de forma a coordenar o movimento seguinte no sentido de reduzi-lo.

A estratégia que utiliza a informação visual para retroação do sinal de saída em sistemas de posicionamento é denominada servo-visão. É a abordagem mais natural em tarefas de rastreamento que utilizam a câmera fixa ao efetuador [3][4], já que a distância entre os pontos da trajetória na imagem e um *pixel* de referência conduz de forma imediata ao erro de posição, em coordenadas locais da câmera. No sistema implementado, cabe ressaltar, a necessidade de uma taxa de atualização das referências de entrada do sistema de posicionamento superior à taxa máxima de aquisição de imagens impossibilitou, no entanto, a aquisição de uma nova imagem a cada mudança de posição do efetuador. A metodologia então empregada foi a realização do controle em malha aberta durante o envio ao atuador das posições de referência extraídas de uma imagem, e o fechamento da malha durante a aquisição da imagem seguinte.

A utilização de um *plotter* com dois graus de liberdade para realização das tarefas de rastreamento, por sua vez, restringiu o estudo às trajetórias planas, que foram então situadas paralelamente ao plano de retina da câmera. Esta condição favorece a análise das imagens adquiridas com base em técnicas de binarização, ou seja, a classificação dos *pixels* como pertencentes ou não à curva da trajetória se mostra suficiente para o cálculo das posições de referência.

Neste sentido, os algoritmos de binarização de imagens têm também importância fundamental nesta pesquisa, bem como a determinação, numa etapa dos algoritmos de visão denominada análise de cenas, dos *pixels* associados às posições a serem percorridas, em seqüência, pelo efetuador. A conversão em coordenadas no espaço de trabalho dos deslocamentos necessários medidos em coordenadas de imagem exige, ainda, a calibração do sistema óptico empregado.

O sistema implementado foi avaliado em função do desempenho em cada etapa de processamento envolvida, e em termos dos erros das trajetórias reais do efetuador em relação às trajetórias de entrada consideradas.

1.2 REVISÃO BIBLIOGRÁFICA

Diversas pesquisas têm sido realizadas com relação à utilização de técnicas de visão para rastreamento de trajetórias e guiagem de robôs. Alguns trabalhos que serviram de base para a presente pesquisa são a seguir revisados. A ordem de apresentação adotada segue, de forma aproximada, aquela em que os problemas são abordados pelo sistema proposto.

Na determinação dos *pixels* que representam os pontos da trajetória a ser rastreada, surge o problema da classificação da imagem tendo em vista a diferença de reflectância entre a trajetória e o fundo, suposto homogêneo. Entretanto, a informação disponível diz respeito à luminância em cada *pixel*, diretamente proporcional à energia nele irradiada durante a formação da imagem e que, portanto, depende não só da reflectância dos objetos representados mas também das condições de iluminação e de outros fatores, como a direção dos raios de luz que partem da cena e sensibilizam o plano de retina da câmera. Neste contexto, métodos que permitam extrair da luminância a informação de reflectância são de grande interesse.

Um destes métodos foi apresentado por Nayar e Bolle em 1993 [22]. O algoritmo é bastante geral no sentido de que se aplica a imagens não só de objetos planos, mas também de estruturas tridimensionais; a única hipótese quanto à geometria da cena é que a direção da reta normal aos objetos não varie bruscamente de um ponto a outro, ou seja, eles devem ser constituídos de superfícies que formam curvas suaves. Para avaliação dos algoritmos, foram adquiridas imagens de diferentes objetos, variando-se a posição da câmera e da fonte de luz de forma que tanto a potência irradiada nas superfícies analisadas como a direção dos raios incidentes no plano de retina variassem de uma imagem para outra. Constatou-se, assim, que o método permitiu, como fora proposto, avaliar a reflectância a menos de uma constante multiplicativa, ou seja, determinar a razão entre a reflectância de cada região da imagem e seu fundo; é eliminada, portanto, a influência das condições de iluminação e das estruturas tridimensionais das superfícies.

No caso da análise de trajetórias planas, porém, uma abordagem tão genérica se mostra um esforço computacional desnecessário, já que, satisfeitas algumas condições pelo sistema óptico, vários fatores que, em princípio, influenciam o valor da luminância podem, de fato, ser desprezados, conforme será abordado no capítulo 3. Ainda assim, a influência das condições de iluminação na relação entre reflectância e luminância é um aspecto crítico, a não ser em condições bastante específicas que não puderam ser satisfeitas no sistema implementado. Este aspecto foi comprovado por Land ainda em 1959 [15], mesmo para o caso de estruturas planas.

Na literatura especializada, duas classes de algoritmos que abordam o problema são relatados. No primeiro caso, trata-se de realizar o processo de limiarização diretamente a partir dos níveis de cinza associados a cada *pixel*, sendo determinados limiares para várias

regiões da imagem por métodos específicos que permitam levar em conta condições não-ideais de iluminação. O outro grupo, ao contrário, utiliza uma etapa de pré-processamento que tem por fim compensar estas condições, e, em seguida, realiza um processo de binarização que supõe iluminação perfeitamente uniforme.

Parker, em 1991, propôs um algoritmo de binarização pertencente à primeira classe [26]. Designado método do gradiente local de intensidade (LIG – *local intensity gradient*), ele permitiu a classificação de imagens adquiridas sob iluminação bastante heterogênea da cena. O desempenho, para diversas classes de imagens, foi superior ao das várias técnicas do mesmo tipo avaliadas, comparando-se o número total de imagens classificadas corretamente em cada caso.

A segunda classe de algoritmos mencionada teve início em 1964 com Land [14], que propôs um modelo denominado *retinex* aplicável a sinais de domínio contínuo unidimensional, mas motivado pelo problema de se compensar a influência da iluminação não-uniforme na geração de imagens de superfícies planas de reflectâncias constantes. Em seu trabalho, Land desenvolveu uma implementação em *hardware* para o *retinex*, mas a aplicação da técnica à análise de imagens digitais só foi possível quando foram propostas adaptações da teoria a sinais de domínio bidimensional e em tempo discreto.

Horn, em 1974 [12], desenvolveu uma destas adaptações. Ela permite, assim, realizar a compensação da não-uniformidade da iluminação durante a geração de imagens reais e obter uma estimativa da reflectância de cada região em relação à região de máxima reflectância representada. Também esta abordagem permite, portanto, a determinação da reflectância a menos de uma constante multiplicativa; o novo valor relativo estabelecido é denominado *lightness*.

O método apresentado por Horn foi uma das técnicas de pré-processamento investigadas nesta pesquisa. A conclusão a respeito da técnica mais apropriada ao problema em estudo foi, porém, em favor de outro algoritmo [8], portanto incorporado ao sistema de rastreamento completo. O método, mais largamente utilizado e que resulta em menor esforço computacional, tem no entanto a desvantagem de exigir um certo controle das condições de iluminação, que, embora não precise ser uniforme, não deve variar com o tempo.

Uma vez obtida uma medida da reflectância da trajetória a partir das imagens de entrada, surge o problema da classificação dos *pixels* com base nesta informação. O processo de limiarização é a técnica mais empregada com este objetivo; entretanto, o limiar de binarização apropriado é função de vários parâmetros estatísticos das duas distribuições que, a princípio, não são conhecidos.

Alguns métodos foram propostos com o objetivo de permitir a realização automática da binarização, inclusive por meio do cálculo do limiar ótimo, mas em geral se aplicam a classes específicas de imagens. Um exemplo é o algoritmo desenvolvido em 2000 por Ohyama e colaboradores [23] para detecção de uma cavidade cardíaca em ecocardiogramas. Apesar

de se tratar de um método de classificação de imagens baseado no histograma de níveis de cinza, uma de suas etapas envolve uma seqüência de operações em imagens binárias com o intuito de isolar a região correspondente à cavidade procurada de regiões falsas, e neste processo considerações são feitas acerca de sua estrutura. Assim, o algoritmo, posto que apresentou bom desempenho comparado à classificação não-automatizada das imagens, é bastante específico à análise de ecocardiogramas e não permite a extração de trajetórias.

Uma abordagem para determinação automática dos parâmetros estatísticos necessários ao cálculo do limiar ótimo que se mostra apropriada ao problema em estudo foi desenvolvida no contexto desta pesquisa. O método, publicado por Miosso e colaboradores em 2001 [19], aplica-se a imagens de superfícies formadas por 2 regiões de reflectâncias constantes, como é o caso das imagens de trajetórias aqui analisadas. Nas avaliações realizadas, obteve-se um bom desempenho durante a binarização de imagens reais e sintéticas, neste caso sendo estimados os parâmetros com erros inferiores a 3% em relação aos empregados na geração das imagens.

No contexto mais geral da utilização da visão computacional para guiagem de manipuladores ou robôs móveis, diversos trabalhos têm sido publicados nos últimos anos, com diferentes abordagens e para várias finalidades. Uma aplicação bastante visada é a guiagem de robôs móveis submarinos, para os quais sistemas de GPS (*Global Positioning System*) não são ainda disponíveis e a utilização de informação visual para posicionamento se apresenta como uma alternativa importante. Em [16], Marks relatou, em linhas gerais, as experiências até então realizadas neste caso, bem como os principais problemas encontrados. Fujisawa, por outro lado, apresentou, em 2001 [7], um sistema de servo-visão para rastreamento de trajetórias por um robô móvel terrestre, e mostrou como a informação visual pode ser utilizada para controlar simultaneamente a posição e o alinhamento do robô enquanto ele executa o rastreamento de um alvo móvel. Bauchspiess, em 1997 [4], utilizou a técnica de servo-visão para controlar um robô de soldagem, e mostrou como ela pode ser combinada a um sistema de inferência *fuzzy* para automatizar este processo.

Já Moraes, em 1998 [20], realizou um estudo comparativo dos desempenhos das estratégias de servo-visão e *look and move*; foram efetuados testes com os algoritmos aplicados à dinâmica simulada de diferentes manipuladores e a um manipulador real, o Zebra-Zero. O trabalho permitiu concluir que, em relação à técnica *look and move*, em que o sistema de visão é utilizado apenas no cálculo das referências de posição, e não para avaliar o erro na trajetória do efetuador, a estratégia de servo-visão leva a um desempenho bastante superior, no que diz respeito à robustez a imprecisões na calibração da câmera ou no modelamento cinemático dos manipuladores.

Na implementação de sistemas de visão computacional para rastreamento de trajetórias ou navegação de robôs, a minimização do tempo de processamento é freqüentemente um problema central, devendo os algoritmos empregados permitir a reconstituição de obstáculos ou o cálculo de referências de posição em tempo real. Apesar de não ter sido

uma preocupação principal nesta pesquisa, o problema é abordado em diversos trabalhos apresentados na área, seja pela otimização das implementações dos algoritmos, seja pela utilização de arquiteturas de processamento altamente paralelizadas e específicas. Horn, em 1974 e 1986 [13][12] propôs, por exemplo, diversos sistemas analógicos e arquiteturas paralelas para a determinação de estimativas de reflectâncias a partir de imagens adquiridas sob condições reais de iluminação. Um dos sistemas analógicos desenvolvidos é bastante adequado ao cálculo do inverso do operador laplaciano, problema que, em uma das técnicas de compensação aqui investigadas, foi aproximado por um filtro digital FIR, resultando em grande esforço computacional.

Mais recentemente, em 1993, Quenot e seus colaboradores [28] propuseram uma arquitetura para implementação em tempo real de algoritmos mais genéricos de processamento de imagens, que faz uso de 1024 processadores *data-flow* para processamentos em baixo nível e 12 processadores de uso geral para processamentos em nível médio. Uma das aplicações analisadas é a detecção de bordas, o que sugere sua utilização também para extração de trajetórias, já que técnicas deste tipo constituem uma possível abordagem do problema.

1.3 ESTRUTURA DA DISSERTAÇÃO

A seguir, no **capítulo 2**, é apresentada uma descrição do sistema de rastreamento proposto. Inicialmente, trata-se do equipamento utilizado para avaliar, em situações práticas de rastreamento, os algoritmos de visão implementados. É apresentada, então, a estratégia de rastreamento, bem como um diagrama de blocos do sistema completo que permite descrever sucintamente os módulos que o compõem, e que serão detalhados nos capítulos seguintes.

O **capítulo 3** trata do processo de binarização das imagens adquiridas da trajetória de entrada durante o rastreamento. Para tanto, é abordado, na primeira parte, o processo de formação das imagens, o que permite estabelecer uma relação entre a luminância de cada *pixel* e a reflectância do ponto que ele representa na cena, bem como avaliar de que forma as condições de iluminação influenciam esta relação. Em seguida, uma análise da medida de probabilidade associada à luminância de cada *pixel* permite obter um modelo estatístico da imagem de uma trajetória sobre um fundo homogêneo quanto à reflectância; mostra-se, então, que o processo de limiarização permite classificar os *pixels* da imagem, de forma a determinar quais representam os pontos da trajetória a ser rastreada. A última parte do capítulo tece comentários de ordem prática sobre o processo, inclusive no que diz respeito à necessidade de se determinar um limiar de binarização apropriado.

O **capítulo 4** representa uma contribuição desta pesquisa. É feita a proposta de um algoritmo em duas etapas para a determinação automática dos parâmetros estatísticos de uma imagem de uma trajetória, segundo o modelo apresentado no capítulo 3. Conhecidos

estes parâmetros, é possível determinar o valor ótimo do limiar de binarização por meio de uma equação em forma fechada. Testes preliminares do algoritmo proposto são apresentados; o método é aplicado a imagens reais e sintéticas de trajetórias, o que permite avaliar seu desempenho na determinação dos parâmetros estatísticos e do limiar de binarização.

A influência de condições não-ideais de iluminação durante a formação das imagens é em seguida abordada, no **capítulo 5**. Tendo em vista os experimentos propostos por Land, avalia-se de que forma estas condições podem levar à degradação dos histogramas das imagens, impossibilitando, assim, a aplicação direta dos algoritmos propostos nos capítulos 3 e 4. São então investigadas duas técnicas de compensação destas condições, e sua comparação é feita de forma a se escolher o método de pré-processamento mais adequado ao sistema em estudo.

O **capítulo 6** aborda o problema de, a partir das imagens binarizadas, extrair os *pixels* que representam posições a serem percorridas, em seqüência, pelo TCP (*Tool Center point*). As coordenadas destes *pixels* serão então convertidas em referências de entrada do sistema de posicionamento, pelo módulo de controle descrito no **capítulo 7**. Mostra-se que, da calibração do sistema óptico, obtém-se uma relação linear entre as distâncias dos *pixels* extraídos a um *pixel* de referência e os deslocamentos necessários do TCP.

Finalmente, o **capítulo 8** apresenta resultados de testes de desempenho do sistema completo. São apresentadas quatro trajetórias de entrada, bem como as referências de entrada do sistema de posicionamento obtidas em cada caso pelos algoritmos de visão e as trajetórias efetivas seguidas pelo TCP. Em um dos casos, são ainda ilustrados todos os resultados intermediários obtidos pelo sistema; para as demais trajetórias, os resultados intermediários são também apresentados, no apêndice D.

As conclusões desta análise e sugestões para trabalhos futuros são apresentadas no **capítulo 9**.

Capítulo 2

DESCRIÇÃO GENÉRICA DO SISTEMA DE RASTREAMENTO

Conforme já ressaltado no capítulo 1, o objetivo desta pesquisa foi implementar um sistema de visão computacional destinado à guiagem de manipuladores robóticos durante a realização de tarefas que envolvem o rastreamento de trajetórias previamente estabelecidas. Para que os algoritmos propostos pudessem ser executados e avaliados em situações práticas de rastreamento, foi empregado um *plotter* com dois graus de liberdade, que, controlado por um microcomputador PC, permitiu o rastreamento de trajetórias tendo como entradas exclusivamente as imagens delas adquiridas.

A seção 2.1 descreve sucintamente os equipamentos empregados — o *plotter*, o microcomputador PC e a câmera — e a sua ligação para a constituição de um sistema simples de rastreamento de trajetórias. Em seguida, na seção 2.2, é apresentado um diagrama de blocos do sistema completo, juntamente com uma descrição introdutória de seu funcionamento. Uma discussão mais detalhada de cada módulo do sistema, bem como do equipamento principal de rastreamento e dos circuitos de interface, é realizada nos próximos capítulos, segundo a estrutura da dissertação proposta na seção ??.

2.1 EQUIPAMENTO UTILIZADO PARA AVALIAÇÃO DOS ALGORITMOS DE VISÃO COMPUTACIONAL PROPOSTOS

O dispositivo empregado no sistema proposto de rastreamento de trajetórias é um *plotter* com dois graus de liberdade, ilustrado na figura 2.1. Seu *Tool Center Point* (TCP) ou efetuador pode se deslocar em dois eixos perpendiculares x e y ; a posição exata em relação a cada eixo é definida pela diferença de potencial elétrico aplicada no canal correspondente, conforme é detalhado no apêndice A.

O objetivo dos algoritmos implementados foi fazer com que o TCP percorresse no plano xy trajetórias bem definidas, a partir da informação extraída de imagens adquiridas

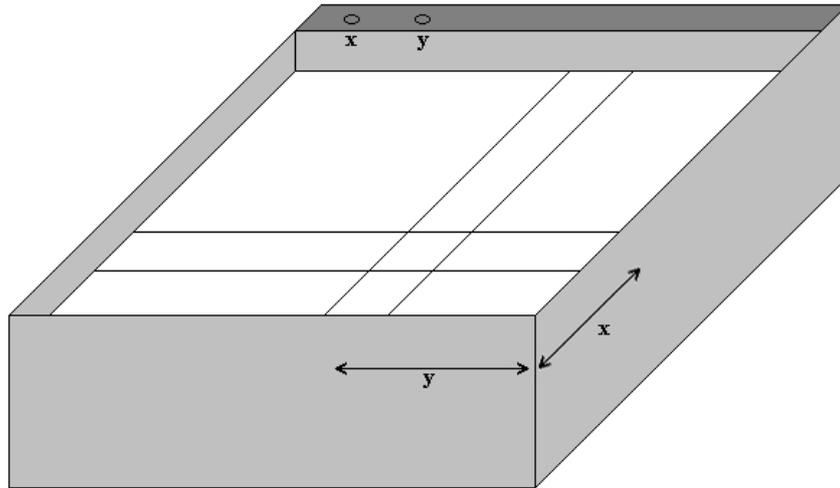


Figura 2.1: *Plotter* com 2 graus de liberdade utilizado para rastreamento das trajetórias, no sistema de visão computacional implementado.

destas trajetórias a intervalos regulares de tempo. Para tanto, foi fixada no TCP uma câmera de vídeo, voltada para um plano acima do *plotter* paralelo a xy , ao qual foram fixadas as trajetórias a serem rastreadas (ver figura 2.2). Junto à câmera foi ainda fixada uma lâmpada halógena, por razões que serão apresentadas no capítulo 5.

A adoção de uma câmera presa ao TCP, em vez de uma fixa na base do *plotter*, deveu-se ao fato de permitir aquela estrutura uma maior resolução espacial. Na configuração *eye-in-hand*, como é denominada [10], cada imagem adquirida deve representar apenas um segmento da trajetória em torno da posição atual do manipulador, já que a informação referente aos segmentos restantes é adquirida à medida que ele é deslocado e novas imagens da trajetória são tomadas a partir de outras posições. Na configuração com a câmera fixa no espaço de trabalho, por outro lado, toda a trajetória é projetada no plano de retina da câmera, o que significa que uma mesma diferença em número de *pixels* entre dois pontos da imagem corresponde, nesta situação, a uma maior distância entre os pontos a eles associados na trajetória. Assim, um mesmo erro na avaliação da posição de um ponto da trajetória a partir de uma imagem na primeira situação se traduz em um erro menor de posicionamento do TCP, o que significa uma maior precisão durante o rastreamento.

Todo o rastreamento é então realizado a partir das imagens adquiridas na situação descrita pela figura 2.2. A câmera, modelo WebCam 3 da Creative, é ligada a um Pentium III 300MHz, por meio de uma porta USB. Neste computador, são executados os algoritmos de visão computacional, que têm por objetivo determinar as posições a que deve ser enviado o TCP, informação extraída exclusivamente das imagens adquiridas. Calculadas estas posições, resta ainda ao PC controlar o movimento do TCP, o que é feito via porta paralela. Um circuito de interface entre esta porta e o *plotter* permite estabelecer os sinais de tensão elétrica apropriados nos canais x e y , de acordo com os

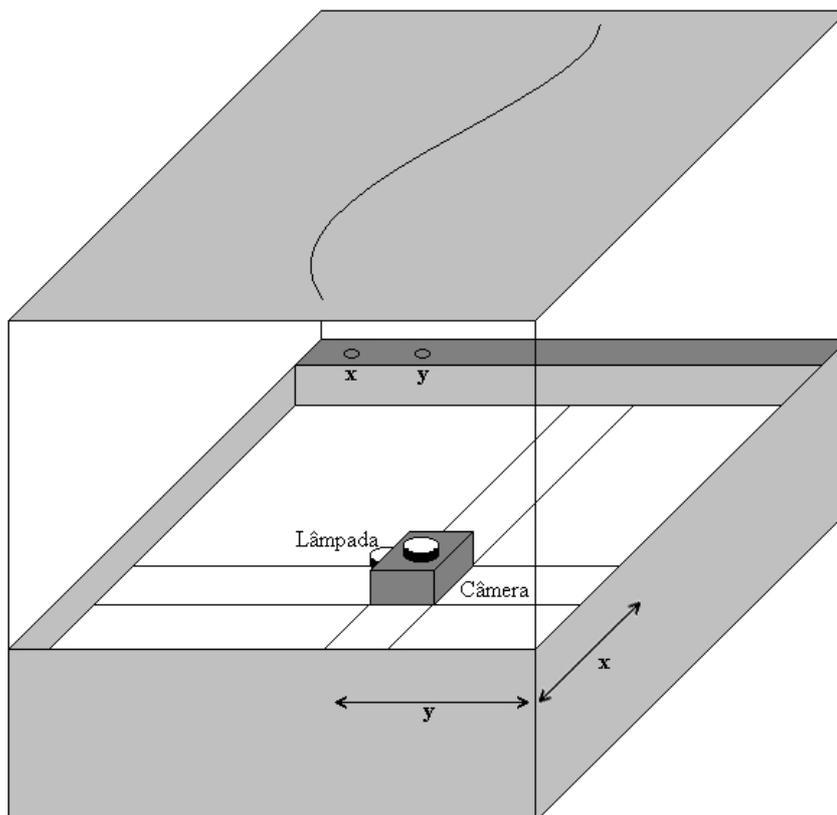


Figura 2.2: Disposição da câmera e da lâmpada sobre o TCP do *plotter* para aquisição das imagens da trajetória a ser rastreada, em um plano paralelo ao espaço de trabalho.

comandos enviados pelo PC.

A figura 2.3 ilustra a ligação entre o PC, o *plotter* e a câmera de vídeo. O circuito de interface utilizado e as rotinas de aquisição de imagens programadas são apresentados no apêndice A.

2.2 DIAGRAMA DE BLOCOS DO SISTEMA DE VISÃO PROPOSTO

Nesta seção, é apresentado, de maneira introdutória, o sistema de rastreamento de trajetórias implementado. O objetivo não é apresentar as técnicas investigadas na solução do problema, mas apenas mostrar a função que exercem no sistema os módulos que o compõem. Cada etapa de processamento envolvida será detalhada posteriormente, ao longo da dissertação.

Considere-se inicialmente o sistema de acionamento do *plotter*, ilustrado na figura 2.4, cuja função é enviar ao atuador uma seqüência de posições especificadas pelas coordenadas $X[n]$ e $Y[n]$, em tempo discreto. O sistema faz uso de rotinas em MatLab que, a intervalos de tempo T , enviam ao circuito de interface PC/*plotter* as seqüências de *bits* apropriadas, que tanto comandam a alteração das tensões de saída $\hat{v}_x(t)$ e $\hat{v}_y(t)$ como informam seus novos valores. No *plotter*, os sinais de tensão assim gerados se traduzem nas posições $\hat{x}_s(t)$

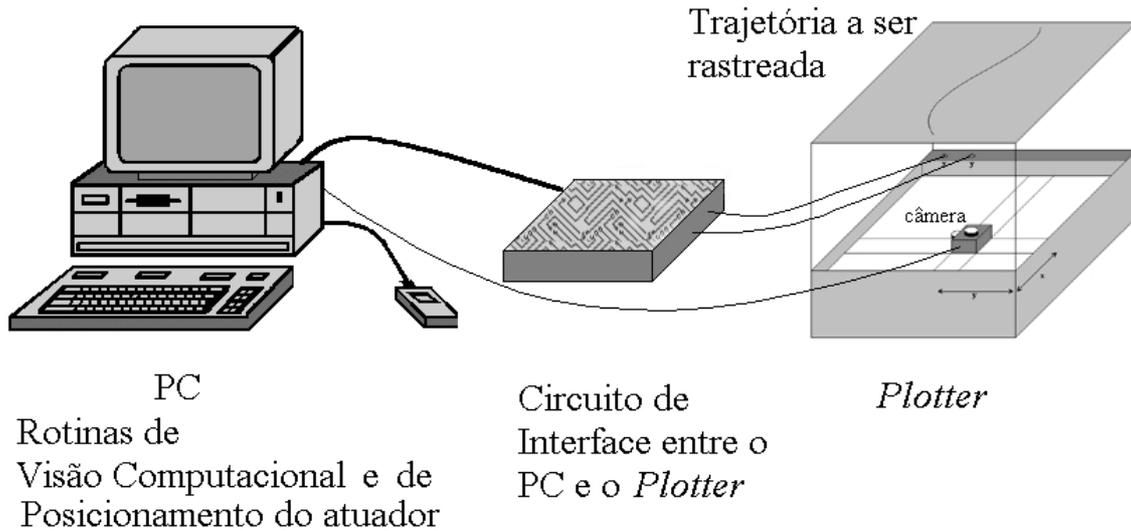


Figura 2.3: Ligação do *plotter* e da câmera ao PC com as rotinas de visão computacional e de posicionamento do TCP.

e $\hat{y}_s(t)$ do TCP, ao longo do tempo.

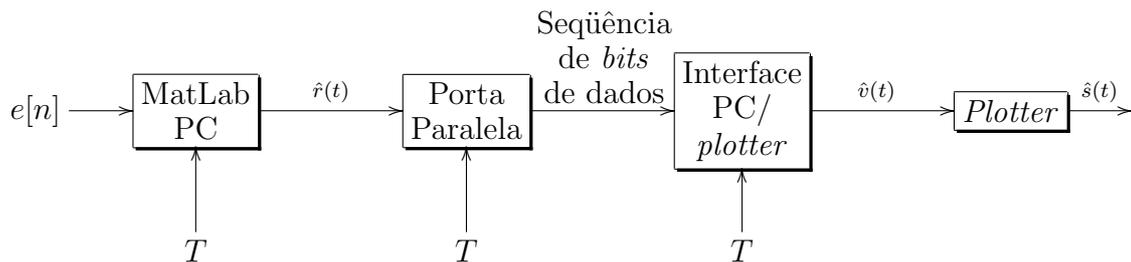


Figura 2.4: Diagrama de blocos do sistema de acionamento do *plotter* a partir da porta paralela do PC.

A determinação da seqüência desejada de posições $(X[n], Y[n])$ é feita exclusivamente a partir das imagens adquiridas da trajetória a ser rastreada. A idéia, neste caso, é extrair de cada imagem as coordenadas de um número fixo de *pixels* associados a pontos da trajetória, a serem percorridos em seqüência. As novas posições a que se deve enviar o TCP são então calculadas de forma que, durante a formação de imagens subseqüentes, os pontos correspondentes àqueles *pixels* extraídos passem a se projetar numa posição fixa de referência, como o *pixel* central. Por estar a câmera presa ao TCP, a única forma de pontos em seqüência na trajetória formarem imagens em um mesmo *pixel*, em amostragens subseqüentes do sinal por ela gerado, é percorrendo o TCP a mesma trajetória que é fornecida como entrada, conforme desejado.

O sistema implementado, no entanto, não necessita efetivamente adquirir uma imagem da trajetória a cada vez que o TCP é enviado a uma nova posição. De fato, adoção de uma taxa de atualização das posições igual à taxa de aquisição de imagens, ainda que a máxima permitida pela câmera, faria com que o TCP desacelerasse ao atingir um novo

ponto da trajetória, e acelerasse ao ser enviado à nova posição (as desvantagens de uma velocidade não-constante do TCP são discutidas no capítulo 6). Assim, adotou-se uma taxa de atualização das posições superior à de aquisição de imagens, ou seja, para cada imagem adquirida o TCP é enviado, em seqüência, a mais de uma posição de referência obtida daquela imagem. A escolha apropriada das duas taxas de amostragem é discutida no capítulo 7.

O diagrama da figura 2.5 sintetiza as idéias apresentadas. A intervalos regulares de tempo, de medida T_I , a câmera adquire uma imagem da trajetória de entrada a partir da posição atual do TCP. A cada imagem adquirida, os algoritmos de visão computacional são então aplicados com o intuito de se determinar as coordenadas i e j de um número determinado de *pixels* representantes de pontos na trajetória. O TCP deverá então ser mobilizado de forma a reduzir, em seqüência, as diferenças Δi e Δj de cada ponto em relação ao *pixel* de referência (centro) da imagem; assim, o controlador deve converter os valores Δi e Δj em deslocamentos necessários do TCP, em seguida determinando a seqüência de pontos a que ele deverá ser enviado. Uma etapa adicional de processamento é ainda efetuada pelo módulo de controle, para gerar então uma seqüência de pontos uniformemente distanciados ao longo de uma linha contínua que une os pontos da seqüência original. Desta forma, ao ser a nova seqüência enviada ao sistema de acionamento do *plotter* (figura 2.4), o TCP será mobilizado em direção a cada ponto a intervalos regulares de tempo, de medida T , garantindo uma velocidade uniforme ao longo do trajeto.

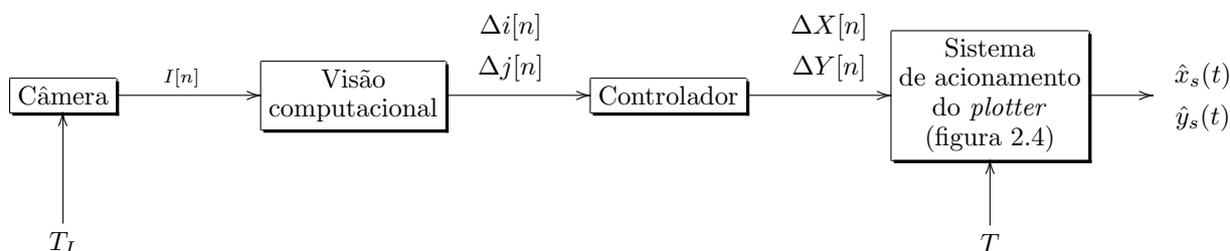


Figura 2.5: Utilização do módulo de visão computacional para o cálculo das referências de entrada do sistema de acionamento do *plotter*.

Cabe ressaltar que, neste sistema, não se dispõe de um sensor específico de posição, além da própria câmera. Esta não permite, no entanto, determinar de forma absoluta a posição do TCP, mas apenas a diferença entre esta posição e as posições desejadas para diferentes instantes (a qual é medida em termos das distâncias, em número de *pixels*, entre os pontos extraídos da imagem pelo módulo de visão computacional e o ponto de referência). Assim, na terminologia comumente empregada em sistemas de controle [5], a câmera pode ser vista como um comparador entre o sinal de saída do sistema — coordenadas $\hat{x}_s(t)$ e $\hat{y}_s(t)$ — e as referências de entrada — coordenadas $\hat{x}(t)$ e $\hat{y}(t)$.

Esta abordagem é ilustrada na figura 2.6. A diferença entre as posições $(\hat{x}(t), \hat{y}(t))$ desejadas do TCP e as posições $(\hat{x}_s(t), \hat{y}_s(t))$ efetivamente ocupadas determina a imagem

que se obtém da trajetória como um todo. Neste caso, se as posições de saída ($\hat{x}_s(t), \hat{y}_s(t)$) não são determinadas de forma absoluta pela câmera, tampouco o são as referências de entrada ($\hat{x}(t), \hat{y}(t)$), já que a trajetória não é observada de uma posição fixa no espaço de trabalho. Portanto, o rastreamento deve começar a partir de uma posição conhecida do TCP, o que permite determinar de forma absoluta as referências $X[n]$ e $Y[n]$ de entrada do módulo de posicionamento a partir dos deslocamentos calculados das imagens pelos módulos de visão computacional e de controle.

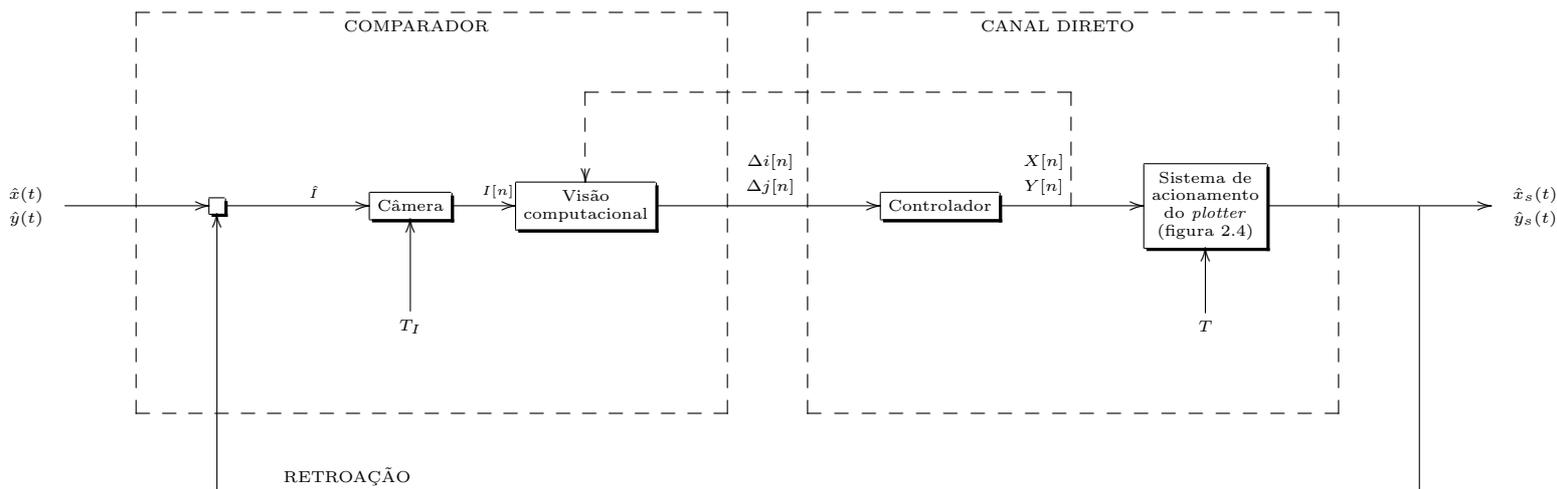


Figura 2.6: Diagrama de blocos do sistema completo de rastreamento de trajetórias proposto.

Outra observação importante é o fato de que, apesar de permitir a câmera fechar a malha de controle do sistema, como evidenciado na figura 2.6, o controle se dá de fato, na maior parte do tempo, em malha aberta. Isto ocorre em virtude da natureza multifrequencial do sistema, já ressaltada: a aquisição de imagens se dá a uma taxa inferior à de atualização das posições do TCP. Assim, se por um lado cada imagem permite avaliar o erro entre a posição atual do TCP e a posição desejada, pela diferença entre as coordenadas da trajetória e do *pixel* de referência, por outro lado uma vez determinadas as entradas do módulo de posicionamento, o TCP é enviado às novas posições sem a utilização de um sensor de saída que permita medir o erro durante o percurso. O controle segue, pois, em malha aberta até que uma nova imagem seja adquirida.

Os capítulos seguintes abordam em detalhes os módulos que compõem o sistema da figura 2.6, com destaque para os algoritmos de visão computacional propostos para a solução do problema, de que tratam os capítulos 3, 4, 5 e 6. O módulo de controle é então tratado no capítulo 7. O sistema de acionamento do *plotter* pela porta paralela do PC, bem como as rotinas de aquisição de imagens, é apresentado detalhadamente no apêndice A.

Capítulo 3

BINARIZAÇÃO DE IMAGENS DE TRAJETÓRIAS

No sistema proposto de seguimento de trajetórias baseado em visão computacional, a primeira etapa da extração das trajetórias a serem seguidas corresponde, conforme será ressaltado no capítulo ??, à binarização das imagens adquiridas. Nesta fase dos algoritmos de visão, denominada *análise de imagens*, o objetivo é classificar os *pixels* que as compõem como representantes da trajetória propriamente dita ou do fundo da cena observada.

Como aborda a seção 3.3, a técnica de binarização utilizada é baseada na diferenciação da trajetória e do fundo a partir da informação de reflectância¹, suposta constante em cada uma daquelas regiões. Assim, este capítulo inicia, na seção 3.1, com um estudo da relação matemática entre a reflectância associada aos objetos que formam a cena analisada e a única informação disponível ao sistema de visão, representada pela luminância² em cada *pixel* da imagem. Em seguida, na seção 3.2, é apresentado um modelamento estatístico das imagens de trajetórias sobre fundos homogêneos, necessário à descrição do processo de binarização empregado. Na seção 3.3, é conceituado este processo e, em particular, o de limiarização, com destaque à forma pela qual aquela informação de luminância é utilizada na classificação dos *pixels* da imagem. Nesta técnica, é essencial a determinação prévia de um limiar de binarização, e a seção 3.4 trata do estabelecimento de uma expressão matemática para o valor teórico ótimo deste limiar, em função dos parâmetros estatísticos da imagem de cujo cálculo automático trata o capítulo 4. Finalmente, a seção 3.5 apresenta algumas considerações de ordem prática sobre o processo de binarização e o cálculo daqueles parâmetros.

¹Reflectância – é uma característica intrínseca dos materiais, a ser definida na página 19.

²Luminância – no caso das imagens em níveis de cinza, é o próprio valor associado ao *pixel*, diretamente proporcional à energia nele irradiada durante a formação das imagens e, conforme será descrito adiante, resultante da interação das condições de iluminação com a reflectância do objeto que ele representa (em geral, depende ainda de outros fatores, como a direção do raio de luz que sensibiliza o *pixel* em relação à normal ao objeto).

3.1 OBTENÇÃO DA RELAÇÃO ENTRE A LUMINÂNCIA NA IMAGEM E A REFLECTÂNCIA DOS OBJETOS QUE COMPÕEM A CENA OBSERVADA

A potência irradiada em cada elemento fotossensível do plano de imagem e, conseqüentemente, a luminância associada a este *pixel* estão diretamente relacionadas à potência radiada pelo objeto observado em direção àquele elemento, através da lente da câmera. Esta radiação, por sua vez, depende, entre outros fatores, da função de distribuição bidirecional de reflectância (FDBR) do objeto. Assim, para que se obtenha a relação desejada entre a reflectância do objeto e a luminância no *pixel* ou conjunto de *pixels* em que se forma sua imagem, é inicialmente considerada, na seção 3.1.1, a dependência entre a potência radiada pelo objeto e sua FDBR. Em seguida, a seção 3.1.2 aborda a relação entre a irradiação³ no plano de imagem e a radiação⁴ a partir do objeto observado. Uma expressão para a luminância de cada *pixel* é então apresentada na seção 3.1.3, em função da potência nele irradiada. Por fim, é estabelecida, na seção 3.1.4, a relação desejada entre luminância na imagem e reflectância do objeto, a partir da síntese de todas as expressões matemáticas anteriores.

3.1.1 Relação entre a Potência Radiada pelos Objetos da Cena e a Função de Distribuição Bidirecional de Reflectância (FDBR)

Segundo a radiometria, são quatro os fatores determinantes da potência luminosa radiada por uma fonte secundária de luz:

1. a potência radiada por todas as fontes de luz que a iluminam;
2. a direção a partir da qual ela é iluminada por cada fonte de luz;
3. a parcela da potência incidente no objeto que é por ele refletida, que depende de sua microestrutura;
4. a direção a partir da qual a potência radiada é medida ou considerada, denominada direção de observação.

Na situação mais genérica possível, cada ponto do objeto pode ser iluminado a partir de todo um hemisfério de direções. Para que seja computado o efeito desta fonte genérica estendida de luz, considere-se inicialmente o sistema de coordenadas esféricas $R-\theta-\phi$ local a cada ponto P da superfície em estudo definido pelas seguintes grandezas (figura 3.1):

³Irradiação – potência que incide numa superfície, por unidade de área.

⁴Radiação – potência emitida por uma superfície, por unidade de área e de ângulo sólido.

a distância R de um ponto considerado a P , o ângulo polar θ entre a reta normal \hat{n} ao objeto em P e a reta r que o une ao ponto considerado, o ângulo de azimute ϕ , entre a projeção de r no plano tangente à superfície em P e uma direção de referência neste plano. Assim, o ponto P pode ser iluminado a partir de cada direção especificada pelas coordenadas θ_i e ϕ_i , referentes a uma fonte situada a uma distância qualquer de P e com os possíveis valores de θ_i (ângulo polar) e ϕ_i (azimute) variando, respectivamente, de 0 a $\pi/2$ e de $-\pi$ a π .

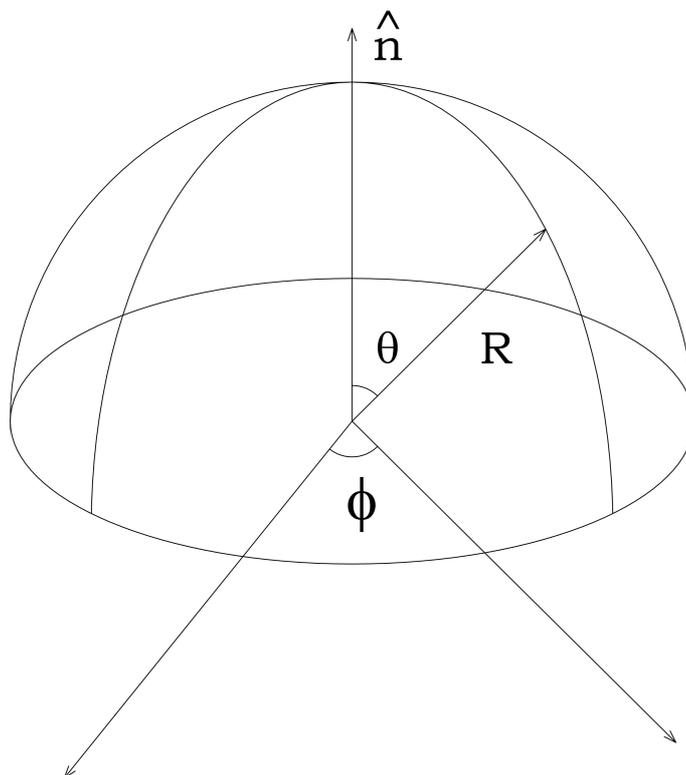


Figura 3.1: Sistemas de coordenadas esféricas $R - \theta - \phi$ local ao ponto P de uma superfície, no qual se avalia a potência irradiada por unidade de área durante a formação de uma imagem.

Se $E(\theta_i, \phi_i, \lambda)$ é a potência por unidade de área, ângulo sólido e comprimento de onda λ radiada pela fonte de luz que ilumina P a partir da direção (θ_i, ϕ_i) , então a potência total $E_0(\lambda)$ irradiada em P , por unidade de área e comprimento de onda, é obtida por integração da função $E(\theta_i, \phi_i, \lambda) \cdot \frac{\cos(\theta_i)}{R^2}$ ao longo de uma semi-calota esférica de raio R centrada em P (o fator $\frac{\cos(\theta_i)}{R^2}$ é associado ao ângulo sólido $\Psi = \frac{A \cos(\theta_i)}{R^2}$ definido pelo cone de direções que parte de cada ponto (R, θ_i, ϕ_i) em direção a uma pequena área A em torno de P [?]; como se considera aqui a potência irradiada em P por unidade de área, o fator se reduz a $\frac{\Psi}{A} = \frac{\cos(\theta_i)}{R^2}$). Assim:

$$E_0(\lambda) = \int_0^{\pi/2} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda) \cdot \frac{\cos(\theta_i)}{R^2} \cdot |J_{x,y,z}(R, \theta_i, \phi_i)| d\phi_i d\theta_i, \quad (3.1)$$

onde $J_{x,y,z}(R, \theta_i, \phi_i)$ é a matriz jacobiana associada à transformação de coordenadas do sistema local R, θ_i, ϕ_i definido para as coordenadas cartesianas $x(R, \theta_i, \phi_i)$, $y(R, \theta_i, \phi_i)$, $z(R, \theta_i, \phi_i)$, dadas por:

$$\begin{cases} x(R, \theta_i, \phi_i) = R \cos \phi_i \text{sen} \theta_i \\ y(R, \theta_i, \phi_i) = R \text{sen} \phi_i \text{sen} \theta_i \\ z(R, \theta_i, \phi_i) = R \cos \theta_i \end{cases} \quad (3.2)$$

Da definição da matriz jacobiana [1], tem-se:

$$J_{x,y,z}(R, \theta_i, \phi_i) = \begin{bmatrix} \frac{\partial x(R, \theta_i, \phi_i)}{\partial R} & \frac{\partial y(R, \theta_i, \phi_i)}{\partial R} & \frac{\partial z(R, \theta_i, \phi_i)}{\partial R} \\ \frac{\partial x(R, \theta_i, \phi_i)}{\partial \theta_i} & \frac{\partial y(R, \theta_i, \phi_i)}{\partial \theta_i} & \frac{\partial z(R, \theta_i, \phi_i)}{\partial \theta_i} \\ \frac{\partial x(R, \theta_i, \phi_i)}{\partial \phi_i} & \frac{\partial y(R, \theta_i, \phi_i)}{\partial \phi_i} & \frac{\partial z(R, \theta_i, \phi_i)}{\partial \phi_i} \end{bmatrix}, \quad (3.3)$$

e o determinante $|J_{x,y,z}(R, \theta_i, \phi_i)|$ se reduz a:

$$|J_{x,y,z}(R, \theta_i, \phi_i)| = \begin{vmatrix} \cos \phi_i \text{sen} \theta_i & \text{sen} \phi_i \text{sen} \theta_i & \cos \theta_i \\ R \cos \phi_i \cos \theta_i & R \text{sen} \phi_i \cos \theta_i & -R \text{sen} \theta_i \\ -R \text{sen} \phi_i \text{sen} \theta_i & R \cos \phi_i \text{sen} \theta_i & 0 \end{vmatrix}, \quad (3.4)$$

$$|J_{x,y,z}(R, \theta_i, \phi_i)| = R^2 \text{sen}^3 \theta_i \text{sen}^2 \phi_i + R^2 \text{sen} \theta_i \cos^2 \phi_i \cos^2 \theta_i + R^2 \text{sen}^2 \phi_i \text{sen} \theta_i \cos^2 \theta_i + R^2 \text{sen}^3 \theta_i \cos^2 \theta_i, \quad (3.5)$$

$$|J_{x,y,z}(R, \theta_i, \phi_i)| = R^2 \text{sen} \theta_i. \quad (3.6)$$

Substituindo a equação (3.6) em (3.1), obtém-se:

$$E_0(\lambda) = \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda) \cdot \frac{\cos \theta_i}{R^2} \cdot R^2 \cdot \text{sen} \theta_i d\phi_i d\theta_i, \quad (3.7)$$

$$E_0(\lambda) = \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda) \text{sen} \theta_i \cos \theta_i d\phi_i d\theta_i. \quad (3.8)$$

No cálculo da potência radiada pelo ponto P numa direção arbitrária (θ_e, ϕ_e) , cada parcela que contribui para a potência irradiada $E_0(\lambda)$ deve ser multiplicada pela razão $f(\theta_i, \phi_i, \theta_e, \phi_e, \lambda)$ entre a potência radiada por P na direção (θ_e, ϕ_e) e a potência nele irradiada a partir de cada direção (θ_i, ϕ_i) . Esta razão é a denominada *reflectância* do objeto; por depender tanto da direção de incidência, (θ_i, ϕ_i) , quanto da de reflexão, (θ_e, ϕ_e) , a função f é denominada *função de distribuição bidirecional de reflectância* (FDBR).

Em termos desta função, a potência radiada por P na direção (θ_e, ϕ_e) e no comprimento de onda λ é portanto dada por:

$$L(\theta_e, \phi_e, \lambda) = \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda) f(\theta_i, \phi_i, \theta_e, \phi_e, \lambda) \sin \theta_i \cos \theta_i d\phi_i d\theta_i. \quad (3.9)$$

3.1.2 Relação entre a Radiação a partir dos Objetos da Cena e a Irradiação no Plano de Imagem

Considere-se agora o problema de se relacionar a potência por unidade de área, ângulo sólido e comprimento de onda $L(\theta_e, \phi_e, \lambda)$ radiada por P à potência irradiada no elemento fotossensível de coordenadas (i, j) onde se forma sua imagem. Como mostrado na figura ??, supõe-se que os raios de luz que partem de P atingem a lente da câmera com um ângulo α , considerado o mesmo para todos os raios se o diâmetro d da lente é muito pequeno comparado à sua distância $-z$ a P , medida ao longo do eixo óptico.

Neste caso, os raios que atingem a lente provenientes de P subentendem um ângulo sólido dado por:

$$\Psi = \frac{A_d \cdot \cos \alpha}{(R_{L,P})^2}, \quad (3.10)$$

onde A_d é a área iluminada da lente e $R_{L,P}$ é sua distância a P .

Como:

$$A_d = \pi \cdot \left(\frac{d}{2}\right)^2 \quad e \quad (3.11)$$

$$R_{L,P} = \frac{-z}{\cos \alpha}, \quad (3.12)$$

segue:

$$\Psi = \pi \cdot \frac{d^2}{4} \cdot \frac{\cos^3 \alpha}{z^2}. \quad (3.13)$$

Calculado o ângulo sólido Ψ associado ao cone de direções a partir das quais P ilumina a lente, pode ser determinada a potência a ela fornecida por uma pequena área A_P em torno de P ao longo do objeto considerado. Se θ é o ângulo entre a normal \hat{n} ao objeto em P e os raios de luz que dele partem em direção à lente (figura ??), esta potência é dada por:

$$P_L(\lambda) = L(\theta_e, \phi_e, \lambda) \cdot \Psi \cdot A_P \cdot \cos \theta, \quad (3.14)$$

ou seja, a potência por unidade de área, ângulo sólido e comprimento de onda, L , é integrada ao longo da área centrada em P considerada e do ângulo sólido Ψ , resultando na potência fornecida à lente por comprimento de onda, P_L .

Substituindo a equação (3.13) em (3.14), obtém-se:

$$P_L(\lambda) = L(\theta_e, \phi_e, \lambda) \cdot \frac{\pi}{4} \cdot d^2 \cdot \frac{\cos^3 \alpha}{z^2} \cdot A_P \cdot \cos \theta. \quad (3.15)$$

Se toda a potência P_L é fornecida à superfície de área A_I onde se forma a imagem de A_P no plano de retina, ou seja, se a lente é um sistema óptico sem perdas, então a potência irradiada em A_I por unidade de área e de comprimento de onda é dada por:

$$P_I(i, j, \lambda) = \frac{P_L(\lambda)}{A_I}. \quad (3.16)$$

Substituindo (3.15) em (3.16), obtém-se:

$$P_I(i, j, \lambda) = L(\theta_e, \phi_e, \lambda) \cdot \frac{\pi}{4} \cdot d^2 \cdot \frac{\cos^3 \alpha}{z^2} \cdot \frac{A_P}{A_I} \cdot \cos \theta, \quad (3.17)$$

e a relação procurada entre $P_I(i, j, \lambda)$ e $L(\theta_e, \phi_e, \lambda)$ pode ser finalmente obtida se for calculada a relação $\frac{A_P}{A_I}$ entre a área A_P no objeto e área A_I no plano de retina em que se forma sua imagem.

Para tanto, observe-se, na figura ??, os raios que partem da superfície de área A_P em direção ao plano de retina, passando pelo centro da lente. Estes raios definem um ângulo sólido Ψ_P dado por:

$$\Psi_P = \frac{A_P \cdot \cos \theta}{(R_{L,P})^2}. \quad (3.18)$$

Os mesmos raios, após atravessarem a lente, definem, em direção à imagem, outro ângulo sólido Ψ_I , dado por:

$$\Psi_I = \frac{A_I \cdot \cos \alpha}{(R_{L,I})^2}, \quad (3.19)$$

com $R_{L,I}$ a distância entre a superfície de área A_I definida e os raios que a atingem provenientes do centro da lente.

Considerando-se que todos os raios de luz que atravessam o centro da lente, suposta esférica ideal, não têm suas direções alteradas [?], conclui-se que os ângulos sólidos Ψ_P e Ψ_I são idênticos. Das equações (3.18) e (3.19), obtém-se portanto:

$$\frac{A_P \cos \theta}{(R_{L,P})^2} = \frac{A_I \cos \alpha}{(R_{L,I})^2}. \quad (3.20)$$

Como $R_{L,P} = \frac{-z}{\cos \alpha}$ e $R_{L,I} = \frac{f}{\cos \alpha}$, onde f é a distância ao longo do eixo óptico do plano de imagem à lente, a equação (3.20) se reduz a:

$$\frac{A_P \cos \theta \cos^2 \alpha}{z^2} = \frac{A_I \cos \alpha \cos^2 \alpha}{f^2}. \quad (3.21)$$

Portanto:

$$\frac{A_P}{A_I} = \left(\frac{z}{f} \right)^2 \cdot \frac{\cos \alpha}{\cos \theta}. \quad (3.22)$$

Substituindo-se a equação (3.22) em (3.17), obtém-se a relação procurada entre $P_I(i, j, \lambda)$ e $L(\theta_e, \phi_e, \lambda)$:

$$P_I(i, j, \lambda) = L(\theta_e, \phi_e, \lambda) \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha. \quad (3.23)$$

3.1.3 Relação entre a Irradiação nos Elementos Fotossensíveis do Plano de Imagem e a Luminância de cada *Pixel*

Durante o processo de formação de imagens por uma câmera digital, a luminância atribuída a cada *pixel* é uma medida quantizada do fluxo de elétrons gerado neste elemento fotossensível pela incidência de fótons provenientes dos objetos que compõem a cena. Assim, para que se obtenha a relação entre esta luminância e a irradiação $P_I(i, j, \lambda)$, é necessário analisar a formação em virtude do efeito fotoelétrico de pares elétron-lacuna nos elementos que compõem o plano de retina.

A irradiação $P_I(i, j, \lambda)$ está associada à incidência no plano de retina de fótons cujo comprimento de onda é λ . De fato, se $b(\lambda)$ é o número de fótons de comprimento de onda λ que nele incidem por unidade de área e de tempo, então:

$$P_I(i, j, \lambda) = k_1 \cdot b(\lambda), \quad (3.24)$$

com k_1 uma constante de proporcionalidade.

Cada um dos $b(\lambda) \cdot A_I \cdot \Delta t$ fótons que incidem no elemento fotossensível de área A_I no intervalo de tempo Δt pode gerar um par elétron-lacuna. Entretanto, não há uma descrição determinística do fenômeno. Considera-se apenas a probabilidade $q(\lambda)$ de que o par de portadores seja gerado, e de que o elétron contribua para o fluxo final medido. Esta probabilidade é denominada *eficiência quântica*, e depende tanto da microestrutura do elemento fotossensível quanto do método utilizado para coletar os elétrons gerados.

Neste sentido, o fluxo de elétrons é modelado matematicamente como uma variável estocástica. Ela é associada ao experimento composto de $b(\lambda) \cdot A_I \cdot \Delta t$ repetições independentes em que se avalia a geração ou não de um par de portadores de carga elétrica, com probabilidade $q(\lambda)$ de sucesso em cada repetição. Uma variável estocástica com estas características tem distribuição binomial, com valores possíveis variando de 0 a $b(\lambda) \cdot A_I \cdot \Delta t$. Como o valor esperado de uma variável estocástica de distribuição binomial é o produto do número de repetições pela probabilidade de sucesso em cada repetição [25], tem-se:

$$\overline{e(\lambda)} = q(\lambda) \cdot b(\lambda) \cdot A_I \cdot \Delta t, \quad (3.25)$$

onde $\overline{e(\lambda)}$ é a expectância da variável estocástica $e(\lambda)$, correspondente ao número de elétrons gerados no elemento fotossensível de área A_I no intervalo Δt , por unidade de comprimento de onda dos fótons incidentes.

Considerando todos os possíveis comprimentos de onda, tem-se que o valor esperado $\bar{\epsilon}$ do fluxo total de elétrons ϵ é:

$$\bar{\epsilon} = \int_{-\infty}^{\infty} \overline{e(\lambda)} d\lambda; \quad (3.26)$$

$$\bar{\epsilon} = A_I \cdot \Delta t \cdot \int_{-\infty}^{\infty} q(\lambda) \cdot b(\lambda) d\lambda. \quad (3.27)$$

Como a luminância $I(i, j)$ é uma medida quantizada do fluxo ϵ , seu valor esperado é dado por:

$$\overline{I(i, j)} = k_2 \cdot \bar{\epsilon}, \quad (3.28)$$

onde k_2 é uma constante de proporcionalidade.

Substituindo a equação (3.27) em (3.28), obtém-se:

$$\overline{I(i, j)} = k_2 \cdot A_I \cdot \Delta t \cdot \int_{-\infty}^{\infty} q(\lambda) \cdot b(\lambda) d\lambda. \quad (3.29)$$

Da relação entre $P_I(i, j, \lambda)$ e $b(\lambda)$ (equação (3.24)):

$$\overline{I(i, j)} = \frac{k_2}{k_1} \cdot A_I \cdot \Delta t \cdot \int_{-\infty}^{\infty} q(\lambda) \cdot P_I(i, j, \lambda) d\lambda. \quad (3.30)$$

Durante a formação de imagens coloridas, são utilizados três tipos de sensores, com diferentes eficiências quânticas $q_1(\lambda)$, $q_2(\lambda)$ e $q_3(\lambda)$, ou ainda um único tipo de sensor combinado com três filtros diferentes, normalmente um vermelho, um verde e um azul. Desta forma, são associados a cada *pixel* (i, j) três valores de luminância $I_1(i, j)$, $I_2(i, j)$ e $I_3(i, j)$, correspondentes à irradiação $P_I(i, j, \lambda)$ ponderada por funções $q(\lambda)$ que assumem valores máximos em três faixas diferentes de frequência. A informação de cor está relacionada portanto às proporções de energia que irradiam no plano de imagem em diferentes regiões do espectro de frequências.

No caso das imagens em níveis de cinza, como as utilizadas pelo sistema de visão aqui pesquisado, só se dispõe da informação da energia irradiada em uma única faixa de frequências. Com efeito, considerando que a eficiência quântica $q(\lambda)$ do único tipo de sensor utilizado se anula fora do intervalo $[\lambda_1, \lambda_2]$, obtém-se, da equação (3.30):

$$\overline{I(i, j)} = \frac{k_2}{k_1} \cdot A_I \cdot \Delta t \cdot \int_{\lambda_1}^{\lambda_2} q(\lambda) \cdot P_I(i, j, \lambda) d\lambda, \quad (3.31)$$

e a luminância $I(i, j)$ depende apenas da irradiação $P_I(i, j, \lambda)$ na região $[\lambda_1, \lambda_2]$ do espectro.

Como a função $q(\lambda) \cdot P_I(i, j, \lambda)$ é contínua em $[\lambda_1, \lambda_2]$, o teorema da média [2] garante que sua integral neste intervalo é igual ao seu valor em um ponto interno a $[\lambda_1, \lambda_2]$ multiplicado pelo comprimento $\lambda_2 - \lambda_1$. Assim:

$$\overline{I(i, j)} = \frac{k_2}{k_1} \cdot A_I \cdot \Delta t \cdot (\lambda_2 - \lambda_1) \cdot q(\lambda_3) \cdot P_I(i, j, \lambda_3), \quad (3.32)$$

com $\lambda_1 \leq \lambda_3 \leq \lambda_2$. O valor esperado da luminância $I(i, j)$ fica assim relacionado à irradiação em um único ponto desconhecido (λ_3) do intervalo de frequências $[\lambda_1, \lambda_2]$ em que o elemento fotossensível utilizado permite capturar os fótons incidentes.

Cabe agora observar que a natureza estatística da descrição realizada não prejudica a obtenção da expressão matemática para a luminância no *pixel* (i, j) . Ela está relacionada à análise em termos probabilísticos da formação de pares de portadores de carga por cada fóton que incide no elemento fotossensível; contudo, se p é a probabilidade de que um fóton gere um elétron que contribui para o fluxo total medido, e se n é o número total de fótons incidentes, então a frequência relativa de elétrons capturados (número total de elétrons dividido por n) tem valor esperado p com variância $\frac{p(1-p)}{n}$. Neste caso, a lei dos grandes números ou desigualdade de Bernoulli [25] garante que a probabilidade de que esta frequência relativa se afaste de p mais do que um número ϵ qualquer ($p[|f - p| \geq \epsilon]$) não ultrapassa $\frac{p(1-p)}{n \cdot \epsilon^2}$, ou seja:

$$p[|f - p| \geq \epsilon] \leq \frac{p(1-p)}{n \cdot \epsilon^2}. \quad (3.33)$$

Observe-se que o segundo membro da inequação tende a 0 quando $n \rightarrow \infty$. Assim, se o número total de fótons incidentes for suficientemente grande, a probabilidade de que f se afaste consideravelmente de p será arbitrariamente pequena. Em outras palavras, ficará garantido que o número total de elétrons gerados e capturados será suficientemente próximo do valor esperado $n \cdot p$ (ou, na equação (3.25), de $q(\lambda) \cdot b(\lambda) \cdot A_I \cdot \Delta t$). Conseqüentemente, também se poderá garantir que a luminância $I(i, j)$ estará arbitrariamente próxima de seu valor esperado, dado pela equação (3.32).

A exigência de que o número de fótons incidentes $b(\lambda) \cdot A_I \cdot \Delta t$ seja suficientemente grande se traduz num produto elevado da área de coleta dos fótons (A_I) pelo tempo de coleta (Δt). Em outras palavras, a câmera deve satisfazer um compromisso entre resolução espacial e resolução temporal. Se isto ocorre, o valor esperado da luminância será uma previsão bastante confiável de seu valor medido $I(i, j)$ e a equação (3.32) pode ser rescrita como:

$$I(i, j) = \frac{k_2}{k_1} \cdot A_I \cdot \Delta t \cdot (\lambda_2 - \lambda_1) \cdot q(\lambda_3) \cdot P_I(i, j, \lambda_3). \quad (3.34)$$

Finalmente, observe-se que os termos k_1 , k_2 , A_I , Δt e $\lambda_2 - \lambda_1$ são constantes, enquanto que $q(\lambda_3)$ depende apenas do comprimento de onda λ_3 . Definindo

$$k(\lambda_3) = \frac{k_2}{k_1} \cdot A_I \cdot \Delta t \cdot (\lambda_2 - \lambda_1) \cdot q(\lambda_3), \quad (3.35)$$

pode-se reescrever (3.34) como:

$$I(i, j) = k(\lambda_3) \cdot P_I(i, j, \lambda_3). \quad (3.36)$$

3.1.4 Relação entre a Luminância de cada *Pixel* e a FDBR dos Objetos que Compõem a Cena Observada

A partir das relações obtidas nas seções 3.1.1, 3.1.2 e 3.1.3, chega-se à relação procurada entre a luminância $I(i, j)$ de cada *pixel* e a FDBR $f(\theta_i, \phi_i, \theta_e, \phi_e, \lambda)$ dos objetos que compõem a cena. Substituindo a equação (3.9) em (3.23) e a igualdade resultante em (3.36), obtém-se:

$$I(i, j) = k(\lambda_3) \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha \cdot \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda_3) f(\theta_i, \phi_i, \theta_e, \phi_e, \lambda_3) \sin \theta_i \cos \theta_i d\phi_i d\theta_i. \quad (3.37)$$

Uma simplificação considerável pode ser ainda realizada na equação (3.37), se os objetos analisados são planos e formados por regiões de reflectâncias constantes, como é o caso das imagens em estudo de trajetórias sobre fundos homogêneos. Nesta situação, se puder ser considerado que a reflectância em cada região pouco varia não só de um ponto a outro mas também com as direções de reflexão (θ_e, ϕ_e) e de incidência (θ_i, ϕ_i) , obtém-se da equação (3.37):

$$I(i, j) = k(\lambda_3) \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha \cdot \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda_3) R(i, j) \sin \theta_i \cos \theta_i d\phi_i d\theta_i, \quad (3.38)$$

onde a dependência da reflectância R com a posição (i, j) do *pixel* reflete o fato de que em cada *pixel* forma-se a imagem de um ponto de uma região específica (trajetória ou fundo), e a reflectância varia de uma região para outra.

Por independer $R(i, j)$ de θ_i ou ϕ_i , tem-se ainda:

$$I(i, j) = k(\lambda_3) \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha \cdot \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda_3) \sin \theta_i \cos \theta_i d\phi_i d\theta_i \cdot R(i, j). \quad (3.39)$$

Observe-se que o termo $\left(\frac{d}{f}\right)^2$ apenas depende das características geométricas do sistema óptico, enquanto que o termo $k(\lambda_3)$ é constante para todos os *pixels* da imagem já que, pela equação (3.35), depende apenas do tipo de elemento fotossensível utilizado e do comprimento de onda λ_3 . Já o termo $\cos^4 \alpha$ depende do ângulo α entre o eixo óptico e o raio de luz que parte do centro da lente para o *pixel*, e portanto varia com suas coordenadas (i, j) . No entanto, esta variação pode ser considerada desprezável devido à quarta potência a que se eleva o cosseno daquele ângulo, desde que as dimensões do plano de imagem sejam pequenas comparadas à distância f ; isto se aplica sobretudo às lentes telefoto [13]. Assim, o termo

$$C = k(\lambda_3) \cdot \frac{\pi}{4} \cdot \left(\frac{d}{f}\right)^2 \cdot \cos^4 \alpha \quad (3.40)$$

pode ser considerado independente do *pixel* (i, j) , e a equação (3.39) se reduz a:

$$I(i, j) = C \cdot \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda_3) \text{sen}\theta_i \cos\theta_i d\phi_i d\theta_i \cdot R(i, j). \quad (3.41)$$

Por fim, note-se que a integral dupla no segundo membro da equação depende apenas das condições de iluminação do ponto P representado, o qual varia com a posição (i, j) do *pixel*. Definindo:

$$Il(i, j) = C \cdot \int_0^{\frac{\pi}{2}} \int_{-\pi}^{\pi} E(\theta_i, \phi_i, \lambda_3) \text{sen}\theta_i \cos\theta_i d\phi_i d\theta_i, \quad (3.42)$$

obtém-se da equação (3.41):

$$I(i, j) = Il(i, j) \cdot R(i, j). \quad (3.43)$$

A equação (3.43) é a relação entre a luminância $I(i, j)$ de cada *pixel* e a reflectância $R(i, j)$ do ponto P de que ele é imagem, em função das condições de iluminação deste ponto expressas no termo $Il(i, j)$.

3.2 MODELAMENTO ESTATÍSTICO DE IMAGENS DE TRAJETÓRIAS SOBRE FUNDOS HOMOGÊNEOS

A descrição apresentada do processo de formação de imagens por uma câmera digital permite concluir que o nível de cinza em cada *pixel* de uma imagem de uma trajetória sobre um fundo plano é diretamente proporcional ao produto da reflectância do ponto correspondente na cena pela energia nele irradiada pela fonte de iluminação. Se a trajetória e o fundo apresentam reflectâncias constantes e são iluminados de maneira uniforme, conclui-se que a imagem apresenta, idealmente, apenas dois níveis de cinza, um para a trajetória (t) e outro para o fundo (f), sendo a luminância no *pixel* (i, j) dada por:

$$I(i, j) = \begin{cases} t, & \text{se } (i, j) \text{ representa um ponto da trajetória} \\ f, & \text{se } (i, j) \text{ representa um ponto do fundo} \end{cases}. \quad (3.44)$$

Nesta situação, o processo de binarização é trivial: *pixels* com o menor nível de cinza encontrado são atribuídos à trajetória, suposta mais escura do que o fundo, enquanto que aqueles com o maior nível de cinza são associados ao fundo.

Na prática, no entanto, dois fatores concorrem para que as imagens formadas apresentem mais do que os dois níveis de cinza ideais e, possivelmente, com que *pixels* associados à trajetória apresentem níveis de cinza inferiores aos do fundo em alguns casos mas superiores em outros. O primeiro destes fatores diz respeito ao fato de que erros experimentais estão inerentemente associados a qualquer processo de medição, e o nível de cinza em cada *pixel* em uma imagem pode ser considerado uma medida quantizada da reflectância de um objeto, desde que iluminado uniformemente. Neste sentido, diversas fontes de ruído

atuam na formação da imagem, que pode portanto ser modelada como a soma de diferentes variáveis estocásticas. Tendo em vista o teorema do limite central [25], pode-se afirmar portanto que o nível de cinza em cada *pixel* constitui uma variável estocástica de distribuição gaussiana, ou seja, com função densidade de probabilidade dada por:

$$f_p(x) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_c}{\sigma_c} \right)^2 \right], \quad (3.45)$$

onde μ_c , o valor esperado da distribuição, é t se p é um *pixel* pertencente à trajetória, f se pertencente ao fundo, e σ_t e σ_f são os desvios-padrões associados às distribuições de um *pixel* da trajetória e de um *pixel* do fundo, respectivamente.

A equação (3.45) é também a expressão para a função densidade de probabilidade de um *pixel* arbitrário, condicionada ao fato de pertencer ele à trajetória (evento T) ou ao fundo (evento F). Assim:

$$f(x|T) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \cdot \exp \left[-\frac{1}{2} \left(\frac{x - t}{\sigma_t} \right)^2 \right] e \quad (3.46)$$

$$f(x|F) = \frac{1}{\sqrt{2\pi\sigma_f^2}} \cdot \exp \left[-\frac{1}{2} \left(\frac{x - f}{\sigma_f} \right)^2 \right]. \quad (3.47)$$

Qualquer *pixel* nas imagens aqui analisadas pertencem ao fundo ou à trajetória, e estes eventos são mutuamente excludentes. Desta forma, T e F constituem uma partição do espaço amostral associado ao experimento em que se escolhe ao acaso um *pixel* na imagem, em seguida classificado como pertencente ao fundo ou à trajetória. Pelo teorema da probabilidade total [25], segue portanto:

$$f(x) = f(x|T) \cdot P(T) + f(x|F) \cdot P(F), \quad (3.48)$$

onde $P(E)$ é a medida de probabilidade do evento E qualquer.

Se p_T é a proporção na imagem de *pixels* pertencentes à trajetória, tem-se ainda:

$$P(T) = p_T, \quad (3.49)$$

$$P(F) = 1 - p_T \quad (3.50)$$

e, das equações (3.46), (3.47) e (3.48), segue:

$$f(x) = p_T \cdot \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp \left[-\frac{1}{2} \left(\frac{x - t}{\sigma_t} \right)^2 \right] + (1 - p_T) \cdot \frac{1}{\sqrt{2\pi\sigma_f^2}} \exp \left[-\frac{1}{2} \left(\frac{x - f}{\sigma_f} \right)^2 \right]. \quad (3.51)$$

A equação (3.51) mostra que a função densidade de probabilidade associada ao nível de cinza em um *pixel* arbitrário é uma média ponderada das distribuições da trajetória e

do fundo, com pesos dados pelas proporções destas duas áreas na imagem. Por esta razão, esta função é designada *densidade de mistura*.

A figura 3.2 apresenta o gráfico de um exemplo de função densidade de mistura, sintetizado a partir da equação (3.51) para valores específicos de μ_1 , μ_2 , σ_1 , σ_2 e p_1 .

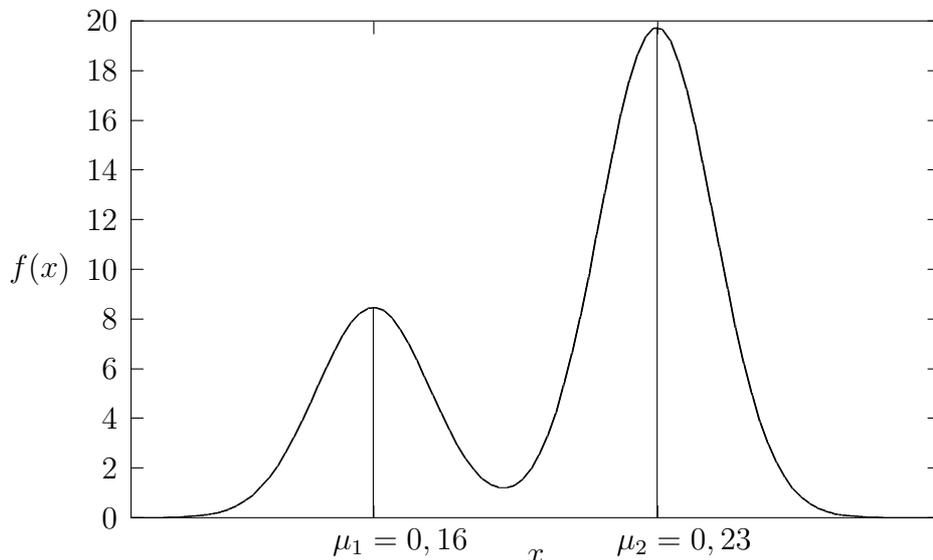


Figura 3.2: Gráfico de um exemplo de função densidade de probabilidade associada ao nível de cinza de um *pixel* arbitrário, na imagem de uma trajetória sobre um fundo homogêneo (densidade de mistura). O gráfico foi sintetizado a partir da equação (3.51), considerando que a média e a variância da distribuição de cada *pixel* da trajetória são $\mu_1 = 0,16$ e $\sigma_1^2 = 0,0002$, que os valores correspondentes da distribuição de cada *pixel* do fundo são $\mu_2 = 0,23$ e $\sigma_2^2 = 0,0002$, e que a proporção de *pixels* pertencentes à trajetória para o número total de *pixels* é de $p_1 = 0,3$.

É importante observar que uma imagem de $M \times N$ *pixels* pode ser analisada estatisticamente como o resultado de $M \cdot N$ repetições de um experimento em que se avalia o nível de cinza de um *pixel* arbitrário, de trajetória ou de fundo. A lei dos grandes números [25] garante, portanto, que a proporção de *pixels* cujo nível de cinza se situa numa faixa qualquer de valores é uma variável estocástica cujo valor esperado é igual à probabilidade de que um *pixel* isolado tenha o nível de cinza na mesma faixa, segundo a equação (3.51), e cuja variância é tanto menor quanto maior o número de *pixels* ($M \cdot N$). Assim, se o número de *pixels* for suficientemente grande, aquela proporção será arbitrariamente próxima da integral de f na faixa de valores considerada. Como esta análise independe desta faixa, conclui-se que o histograma da imagem, que apresenta a quantidade de *pixels* cujo nível de cinza se situa em cada faixa em torno de um valor inteiro, terá uma forma semelhante à do gráfico da densidade de mistura correspondente.

Este fato pode ser observado na figura 3.3. O item (a) mostra a imagem de uma trajetória real sobre um fundo uniforme. Uma técnica de processamento foi contudo empregada para compensar a iluminação não-uniforme utilizada durante a aquisição, conforme será explicado adiante. O item (b), por sua vez, apresenta o histograma correspondente.

O fato de que este se assemelha ao gráfico de uma densidade de mistura (figura 3.2) é uma consequência direta da lei dos grandes números, conforme já ressaltado.

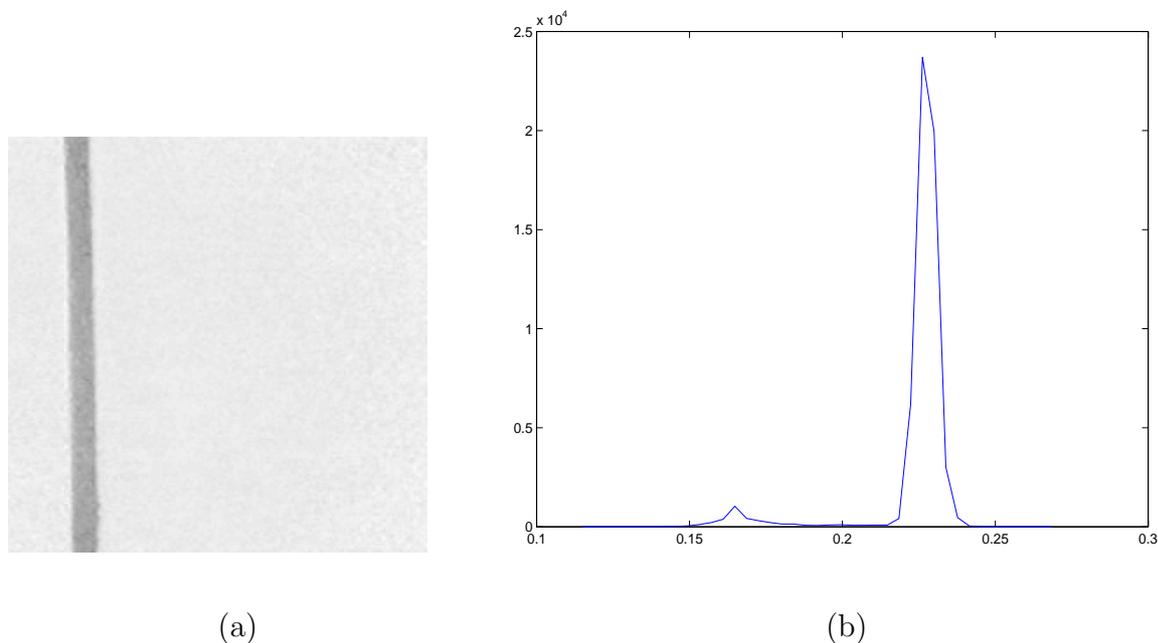


Figura 3.3: (a) Imagem real de uma trajetória sobre um fundo uniforme, após aplicação de uma técnica de compensação da iluminação não-uniforme e (b) histograma correspondente.

O outro fator mencionado que faz com que as imagens de trajetórias sobre fundos homogêneos não sejam compostas, na prática, de apenas dois níveis de cinza diz respeito às condições de iluminação da cena observada. Conforme já apresentado, o nível de cinza em cada *pixel* da imagem é diretamente proporcional ao produto da reflectância do ponto correspondente no objeto pela energia nele irradiada pela fonte de iluminação. Conseqüentemente, se a iluminação é não-uniforme, ou seja, se a energia irradiada varia com a posição considerada no objeto, dois *pixels* poderão apresentar luminâncias distintas ainda que correspondam a pontos em uma região de reflectância constante. Observa-se, inclusive, que um *pixel* associado a um ponto na trajetória pode apresentar nível de cinza inferior a alguns dos *pixels* do fundo mas superior aos dos demais, desde que a variação na iluminação seja suficientemente grande. Neste caso, o processo de binarização não pode ser realizado simplesmente atribuindo-se os *pixels* com menor nível de cinza à trajetória e os demais ao fundo.

As figuras 3.4 e 3.5 ilustram a influência da iluminação não-uniforme nas luminâncias ao longo de uma imagem. Na figura 3.4-(a), é apresentada uma imagem de uma trajetória sobre um fundo homogêneo, suposta adquirida sob iluminação uniforme e sintetizada considerando as funções densidade de probabilidade (3.46) e (3.47) associadas aos níveis de cinza da trajetória e do fundo, respectivamente. A figura 3.4-(b) corresponde a um padrão de iluminação: o aumento da luminância da esquerda para a direita sugere o gradiente de iluminação obtido pela colocação de uma fonte que ilumina a cena da direita para a

esquerda. A imagem representada é aquela que, nesta condição, obter-se-ia idealmente a partir de uma superfície plana de reflectância constante. Finalmente, a figura 3.4-(c) ilustra a imagem obtida a partir da superfície representada em 3.4-(a) sob as condições de iluminação em 3.4-(b). Como o efeito da iluminação é multiplicativo, a imagem 3.4-(c) foi sintetizada a partir do produto *pixel-a-pixel* das imagens 3.4-(a) e 3.4-(b).

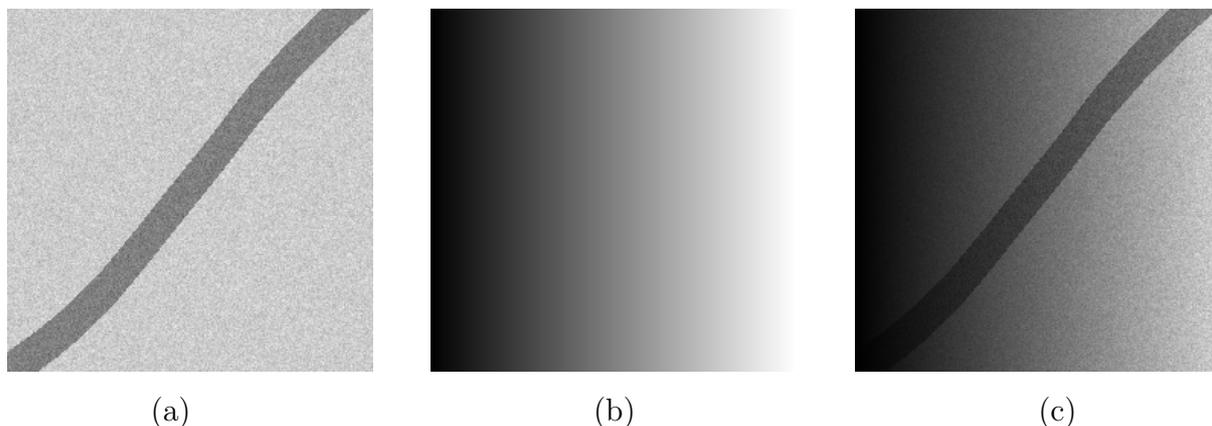


Figura 3.4: (a) Imagem sintética de uma trajetória sobre um fundo, ambos de reflectâncias constantes, suposta adquirida sob iluminação perfeitamente uniforme. (b) Padrão de iluminação obtido pela colocação de uma fonte que ilumina uma superfície plana de reflectância constante a partir da direita para a esquerda. (c) Imagem obtida da superfície em (a) sob a condição de iluminação representada em (b).

Na figura 3.5, são exibidos os histogramas das imagens da superfície em 3.4-(a), sem e com o efeito da iluminação não-uniforme. Em 3.5-(a) é mostrado o histograma da imagem 3.4-(a); como se supõe neste caso que a iluminação é perfeitamente uniforme, uma vez mais se obtém um gráfico cuja forma é semelhante ao da função densidade de probabilidade de mistura (figura 3.2). A figura 3.5-(b), correspondente ao histograma da imagem 3.4-(c), mostra de que forma a iluminação não-uniforme (figura 3.4-(b)) pode corromper o histograma em 3.5-(a). Note-se que, nesta situação, é impossível binarizar a imagem simplesmente associando os *pixels* com os menores níveis de cinza à trajetória e aqueles com os maiores níveis de cinza ao fundo, uma vez que a iluminação não-uniforme tem o efeito de misturar as distribuições no histograma correspondentes a estas duas áreas.

3.3 CLASSIFICAÇÃO DE IMAGENS PELO PROCESSO DE LIMIAÇÃO

O histograma típico de uma imagem de uma trajetória sobre um fundo uniformemente iluminado, como o exemplificado na figura 3.3, é constituído de duas curvas semelhantes à gaussiana, que constituem realizações das duas distribuições de probabilidade (3.46) e (3.47). Considerando que cada uma destas curvas representa a frequência acumulada de *pixels* associados à trajetória ou ao fundo, isoladamente, e supondo que não

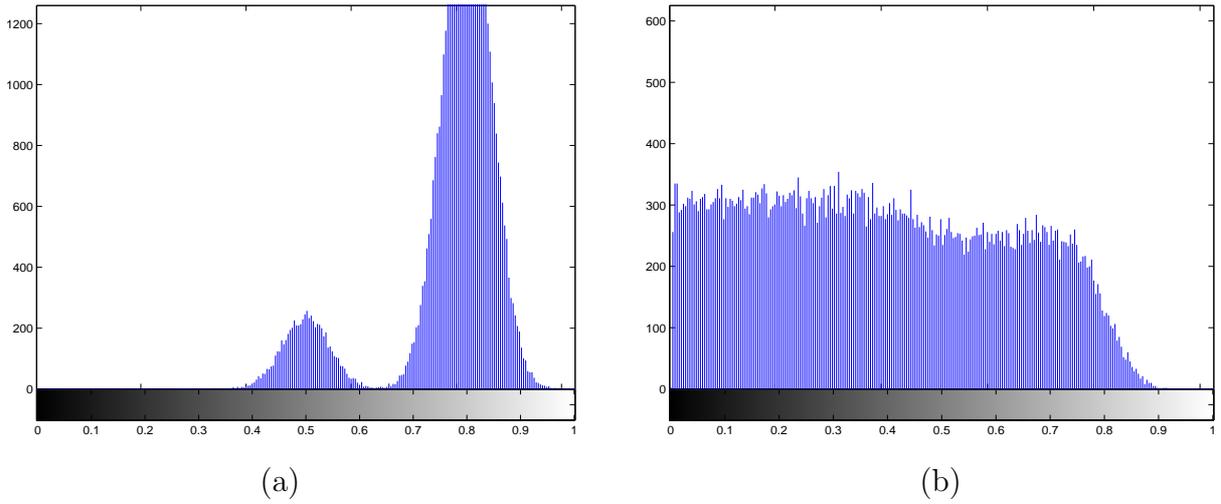


Figura 3.5: (a) Histograma da imagem na figura 3.4-(a), de uma trajetória sobre um fundo suposta adquirida sob iluminação perfeitamente uniforme. (b) Histograma da imagem na figura 3.4-(c), de uma trajetória sobre um fundo adquirida sob a condição de iluminação representada em 3.4-(b).

haja sobreposição entre as duas curvas, ou seja, que todos os pontos de ordenada acima de zero em cada curva tenham ordenada nula na outra, conclui-se que existe um valor L superior às luminâncias de todos os *pixels* de uma classe — trajetória ou fundo — mas inferior às da outra classe. Neste caso, a imagem binarizada I_b pode ser obtida a partir da relação:

$$I_b(i, j) = \begin{cases} 1 & \text{se } I(i, j) < L \\ 0 & \text{se } I(i, j) \geq L \end{cases} . \quad (3.52)$$

Este processo de classificação dos *pixels* da imagem baseada na comparação de suas luminâncias a um valor de referência fixo é denominado limiarização, e a constante L é o limiar de binarização. A seção 3.4 e o capítulo 4 tratam da determinação do valor adequado de L a partir exclusivamente da imagem de entrada a ser binarizada.

Conforme ressaltado na seção anterior, uma iluminação não-uniforme da cena pode, contudo, fazer com que *pixels* em cada classe tenham luminâncias inferiores aos da outra em alguns casos mas superiores em outros, descaracterizando o histograma da imagem como constituído de duas curvas gaussianas e impedindo a aplicação direta do processo de limiarização. O mesmo problema surge, ainda que em menor grau, se as expectativas das duas distribuições são muito próximas, relativamente aos valores de seus desvios-padrões, o que leva à sobreposição das duas curvas que compõem o histograma. Assim, anteriormente à limiarização, foram empregados, neste trabalho, técnicas de compensação da iluminação não-uniforme e um processo de pré-filtragem da imagem de entrada, descritos no capítulo 5. Estes procedimentos permitiram, nos casos analisados, separar as duas distribuições nos histogramas das imagens, e assim realizar a binarização com uso do método

aqui descrito.

3.4 LIMIAR ÓTIMO DE BINARIZAÇÃO

Para que possa ser efetuado o processo de binarização das imagens pelo método descrito na seção 3.3, faz-se necessário determinar, num primeiro momento, o limiar de binarização L apropriado. Nesta seção, obtém-se uma expressão matemática em forma fechada para este limiar, fundamentada no modelo estatístico das imagens deduzido na seção 3.2. Observa-se, no entanto, que ela é dada em função de parâmetros estatísticos que, em geral, não são conhecidos de antemão, e de um algoritmo absolutamente automático de determinação de L se espera que permita o cálculo destas variáveis a partir exclusivamente da imagem de entrada. No capítulo 4, é proposto um algoritmo para determinação automática de todos os parâmetros estatísticos que, conforme deduzido nesta seção, são necessários ao cálculo do limiar ótimo.

Por ora, aborda-se apenas a relação entre o limiar de binarização e os parâmetros estatísticos mencionados, a princípio desconhecidos.

3.4.1 Probabilidade de classificação incorreta de um *pixel*, em função do limiar de binarização adotado

O limiar de binarização deve ser escolhido de forma que a menor quantidade possível de *pixels* na imagem seja classificada incorretamente, durante o processo de limiarização. Dada a natureza estocástica das luminâncias associadas a cada *pixel*, evidenciada nas distribuições de probabilidade (3.46) e (3.47), esta escolha deve ser baseada num modelo probabilístico que descreva o tipo de classificação realizado. Assim, é inicialmente calculada, nesta seção, a probabilidade $p_e(L)$ de que cada *pixel* na imagem seja classificado incorretamente, em função do limiar de binarização L adotado. Em seguida, determina-se, na seção seguinte, o valor de L que minimiza esta probabilidade, definido como o limiar ótimo de binarização.

Dois tipos de erros podem ocorrer durante a limiarização. Considerando μ_1 o menor valor entre μ_1 e μ_2 , na equação (3.51), a distribuição de média μ_1 pode apresentar valores significativamente acima de zero para luminâncias maiores que L . Neste caso, *pixels* correspondentes a esta distribuição, com níveis de cinza cujo valor esperado μ_1 é menor do que L , podem ainda assim ter luminância superior a este limiar, e portanto ser erroneamente associados à distribuição de média μ_2 . Da mesma forma, a distribuição de média μ_2 pode apresentar valores significativamente acima de zero para níveis de cinza abaixo de L , resultando possivelmente na classificação incorreta de alguns de seus *pixels* correspondentes, caso apresentem níveis de cinza abaixo deste limiar.

Assim, durante o cálculo de $p_e(L)$ devem ser considerados:

- o evento $\{(d_1, x)/x > L\}$, que ocorre quando o *pixel* em questão está associado à distribuição d_1 de expectância μ_1 mas tem luminância superior a L ;
- o evento $\{(d_2, x)/x \leq L\}$, que ocorre quando o *pixel* em questão está associado à distribuição d_2 de expectância μ_2 mas tem luminância menor ou igual a L .

Uma classificação incorreta se dá quando da realização de um destes eventos, ou seja, quando o evento $\{(d_1, x)/x > L\} \cup \{(d_2, x)/x \leq L\}$ ocorre. Assim:

$$p_e(L) = P(\{(d_1, x)/x > L\} \cup \{(d_2, x)/x \leq L\}), \quad (3.53)$$

onde $P(E)$ é a medida de probabilidade do evento E qualquer.

Como os eventos $\{(d_1, x)/x > L\}$ e $\{(d_2, x)/x \leq L\}$ são mutuamente excludentes, segue:

$$p_e(L) = P(\{(d_1, x)/x > L\}) + P(\{(d_2, x)/x \leq L\}); \quad (3.54)$$

mas:

$$\{(d_1, x)/x > L\} = (L, \infty) \cap D_1 \text{ e} \quad (3.55)$$

$$\{(d_2, x)/x \leq L\} = (-\infty, L] \cap D_2, \quad (3.56)$$

onde $D_1 = \{(d_1, x)/x \in R\}$ e $D_2 = \{(d_2, x)/x \in R\}$.

Por definição da probabilidade condicional [25]:

$$P((L, \infty)|D_1) = \frac{P((L, \infty) \cap D_1)}{P(D_1)} \text{ e} \quad (3.57)$$

$$P((-\infty, L]|D_2) = \frac{P((-\infty, L] \cap D_2)}{P(D_2)}. \quad (3.58)$$

Logo:

$$P((L, \infty) \cap D_1) = P((L, \infty)|D_1) \cdot P(D_1) \text{ e} \quad (3.59)$$

$$P((-\infty, L] \cap D_2) = P((-\infty, L]|D_2) \cdot P(D_2). \quad (3.60)$$

Mas $P((L, \infty)|D_1)$ pode ser obtida integrando-se a função densidade de probabilidade condicionada — equações (3.46) e (3.47) — no intervalo (L, ∞) . Assim:

$$P((L, \infty)|D_1) = \int_L^\infty f(x|D_1) dx. \quad (3.61)$$

Analogamente, para o evento D_2 :

$$P((-\infty, L]|D_2) = \int_{-\infty}^L f(x|D_2) dx. \quad (3.62)$$

Substituindo as equações (3.61) e (3.62) em (3.59) e (3.60), segue:

$$P((L, \infty) \cap D_1) = p_1 \cdot \int_L^\infty f(x|D_1) dx \quad \text{e} \quad (3.63)$$

$$P((-\infty, L] \cap D_2) = (1 - p_1) \cdot \int_{-\infty}^L f(x|D_2) dx, \quad (3.64)$$

e a equação (3.53) fica:

$$p_e(L) = p_1 \cdot \int_L^\infty f(x|D_1) dx + (1 - p_1) \cdot \int_{-\infty}^L f(x|D_2) dx. \quad (3.65)$$

Finalmente, substituindo em (3.65) as expressões para as funções densidade de probabilidade condicionada — equações (3.46) e (3.47) — obtém-se a expressão que relaciona a probabilidade de que um *pixel* seja classificado incorretamente ao limiar de binarização L adotado:

$$p_e(L) = \frac{p_1}{\sqrt{2\pi\sigma_1^2}} \cdot \int_L^\infty \exp\left[-\frac{1}{2}\left(\frac{x - \mu_1}{\sigma_1}\right)^2\right] dx + \frac{1 - p_1}{\sqrt{2\pi\sigma_2^2}} \cdot \int_{-\infty}^L \exp\left[-\frac{1}{2}\left(\frac{x - \mu_2}{\sigma_2}\right)^2\right] dx. \quad (3.66)$$

3.4.2 Escolha do limiar de binarização que minimiza a probabilidade de classificação incorreta de cada *pixel*

Dada a equação (3.66), que relaciona a probabilidade de que um *pixel* na imagem seja classificado incorretamente ao valor de L adotado, o problema de determinação do limiar ótimo de binarização se resume ao cálculo do valor de L que minimiza aquela probabilidade. Em outras palavras, deseja-se obter o ponto de mínimo global da função p_e .

Os pontos de mínimo locais de uma função contínua de uma variável são aqueles em que a derivada de primeira ordem é nula e a derivada de segunda ordem é positiva [2]. Assim, o limiar ótimo L deve satisfazer as condições

$$\frac{d}{dL} p_e(L) = 0 \quad \text{e} \quad (3.67)$$

$$\frac{d^2}{dL^2} p_e(L) > 0. \quad (3.68)$$

Da equação (3.66), obtém-se, para a derivada de primeira ordem:

$$\frac{d}{dL} p_e(L) = -\frac{p_1}{\sqrt{2\pi\sigma_1^2}} \cdot \exp\left[-\frac{1}{2}\left(\frac{L - \mu_1}{\sigma_1}\right)^2\right] + \frac{1 - p_1}{\sqrt{2\pi\sigma_2^2}} \cdot \exp\left[-\frac{1}{2}\left(\frac{L - \mu_2}{\sigma_2}\right)^2\right], \quad (3.69)$$

e a equação (3.67) fica:

$$\frac{p_1}{\sigma_1} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_1}{\sigma_1} \right)^2 \right] = \frac{1 - p_1}{\sigma_2} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_2}{\sigma_2} \right)^2 \right]. \quad (3.70)$$

Aplicando o logaritmo natural aos dois membros da equação, obtém-se:

$$\ln \frac{p_1}{\sigma_1} - \frac{1}{2} \left(\frac{L - \mu_1}{\sigma_1} \right)^2 = \ln \frac{1 - p_1}{\sigma_2} - \frac{1}{2} \left(\frac{L - \mu_2}{\sigma_2} \right)^2, \quad (3.71)$$

que é uma equação polinomial na incógnita L .

Reescrevendo (??):

$$(\sigma_1^2 - \sigma_2^2)L^2 + (2\mu_1\sigma_2^2 - 2\mu_2\sigma_1^2)L + \left[\sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln \left(\frac{\sigma_2 p_1}{\sigma_1(1 - p_1)} \right) \right] = 0. \quad (3.72)$$

Se $\sigma_1^2 \neq \sigma_2^2$, ou seja, se as distribuições de probabilidade associadas às luminâncias da trajetória e do fundo têm variâncias distintas, trata-se de uma equação de segunda ordem; caso contrário, a equação é de primeira ordem. A solução geral é dada por:

$$L = \begin{cases} \frac{\mu_2^2 - \mu_1^2 + 2\sigma^2 \ln \left(\frac{p_1}{1 - p_1} \right)}{2(\mu_2 - \mu_1)} & \text{se } \sigma_1^2 = \sigma_2^2 = \sigma^2 \\ \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} & \text{caso contrário} \end{cases}, \quad (3.73)$$

onde:

$$\begin{aligned} A &= \sigma_1^2 - \sigma_2^2, \quad B = 2\mu_1\sigma_2^2 - 2\mu_2\sigma_1^2 \\ \text{e } C &= \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln \left(\frac{\sigma_2 p_1}{\sigma_1(1 - p_1)} \right). \end{aligned} \quad (3.74)$$

Deve-se ainda verificar quais raízes em 3.73 satisfazem também a inequação 3.68. Derivando a equação (3.69) com respeito a L , obtém-se:

$$\frac{d^2}{dL^2} p_e(L) = \frac{p_1(L - \mu_1)}{\sqrt{2\pi\sigma_1^2} \cdot \sigma_1} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_1}{\sigma_1} \right)^2 \right] - \frac{(1 - p_1)(L - \mu_2)}{\sqrt{2\pi\sigma_2^2} \cdot \sigma_2} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_2}{\sigma_2} \right)^2 \right], \quad (3.75)$$

e a inequação 3.68 pode ser rescrita como:

$$\frac{p_1(L - \mu_1)}{\sqrt{2\pi\sigma_1^2} \cdot \sigma_1} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_1}{\sigma_1} \right)^2 \right] > \frac{(1 - p_1)(L - \mu_2)}{\sqrt{2\pi\sigma_2^2} \cdot \sigma_2} \cdot \exp \left[-\frac{1}{2} \left(\frac{L - \mu_2}{\sigma_2} \right)^2 \right]. \quad (3.76)$$

Assim, o limiar ótimo de binarização é obtido a partir da equação (3.73), em forma fechada, devendo em seguida ser verificado qual das raízes satisfaz o critério (3.76). Observe-se que os cálculos envolvidos exigem, contudo, que sejam conhecidos os cinco parâmetros estatísticos p_1 , μ_1 , σ_1^2 , μ_2 e σ_2^2 . No capítulo 4 é proposto um algoritmo para a determinação automática destes parâmetros a partir exclusivamente da imagem de entrada a ser binarizada.

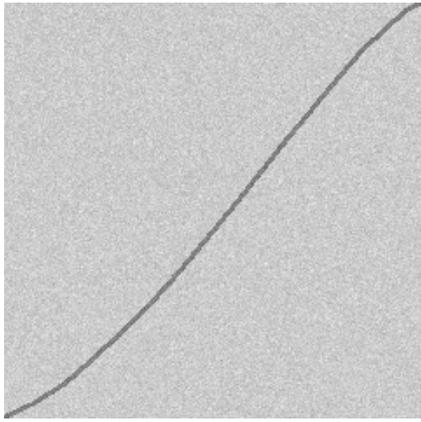
3.5 CONSIDERAÇÕES DE ORDEM PRÁTICA SOBRE O PROCESSO DE BINARIZAÇÃO

O fato de ser o limiar de binarização função de cinco variáveis estatísticas a princípio desconhecidas e o esforço computacional adicional representado pela determinação automática destes parâmetros, destacado no capítulo 4, sugerem a determinação, por tentativa e erro, de um valor empírico deste limiar, previamente ao início do seguimento da trajetória. Entretanto, o que se observou é que, mesmo que parâmetros como as expectâncias e as variâncias das distribuições f_T e f_F sejam mantidos constantes durante todo o processo, o que já exige um bom controle da operação no que diz respeito às condições de iluminação e à uniformidade da trajetória e do fundo, a variação da proporção p_1 de uma imagem para outra e a conseqüente mudança do valor de L segundo (3.73) podem levar a classificações incorretas, caso o limiar utilizado não seja adaptado.

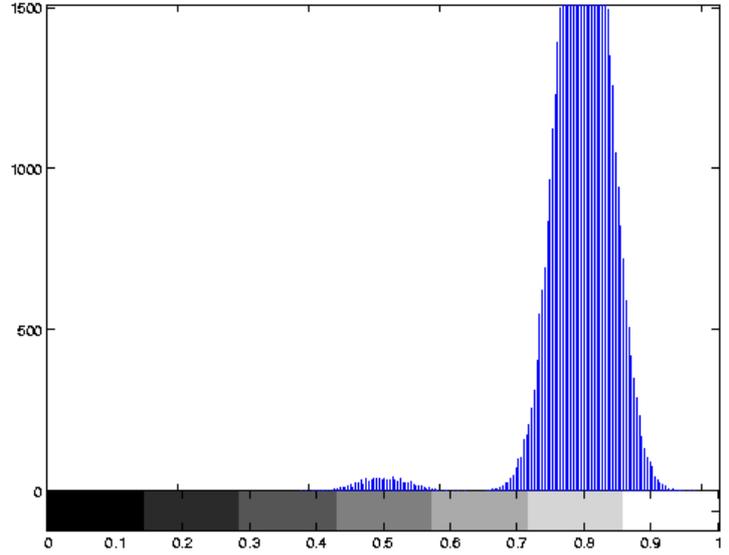
Dois exemplos práticos comprovam este fato. O primeiro caso é apresentado na figura 3.6; dois segmentos de trajetória — figuras 3.6-(a) e 3.6-(c) — apresentam espessuras distintas, o que resulta em diferentes valores da proporção p_1 , conforme se observa nos histogramas correspondentes — figuras 3.6-(b) e 3.6-(d). Neste caso, um limiar $L = 0,6303$, calculado a partir da equação (3.73) para a imagem 3.6-(a), é adequado à classificação desta trajetória, mas não da representada em 3.6-(c). De fato, as figuras 3.7 e 3.8 destacam o limiar utilizado durante a binarização daquelas imagens, bem como os resultados da binarização. No primeiro caso, verifica-se a classificação correta de todos os *pixels*, enquanto que, no segundo, diversos *pixels* são classificados incorretamente; um limiar diferente ($L = 0,6525$) deve então ser adotado neste caso, garantindo a classificação correta conforme mostra a figura 3.9. Note-se que, no entanto, as imagens mostradas na figura 3.6 são idênticas no que diz respeito às médias e às variâncias das distribuições f_T e f_F , apenas variando a proporção de trajetória para *pixel* em cada caso.

É importante observar que as imagens podem apresentar linhas mais grossas ou mais finas ainda que as trajetórias correspondentes tenham espessuras iguais, ou mesmo que se trate de uma só trajetória. Isto ocorre se as imagens forem adquiridas em planos de retina a diferentes distâncias da cena observada; neste caso, o número de *pixels* entre as bordas da linha, na imagem, será maior quando a aquisição se der a uma menor distância da trajetória. Esta situação é inevitável no caso de manipuladores que devem seguir trajetórias não-planas, tendo o TCP mais de dois graus de liberdade.

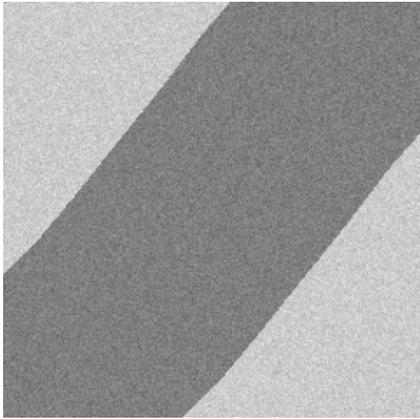
O mesmo problema pode surgir ainda que a espessura da trajetória seja constante e que ela esteja situada num plano a uma distância fixa ao plano de retina. A figura 3.10 mostra dois trechos de diferentes comprimentos de uma mesma trajetória. Os histogramas correspondentes — figuras 3.10-(b) e 3.10-(d) — mostram uma variação considerável na proporção p_1 entre as duas imagens. Mais uma vez, um limiar que se mostra adequado à binarização da primeira imagem ($L = 0,6274$), destacado na figura 3.11 juntamente



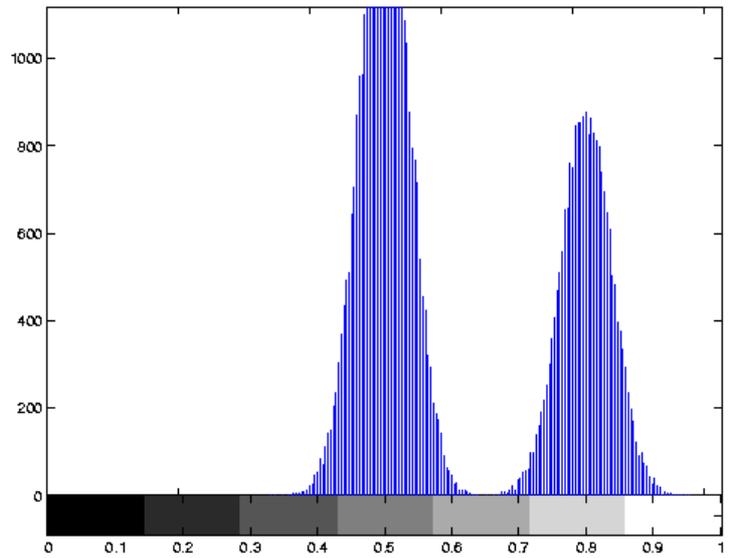
(a)



(b)

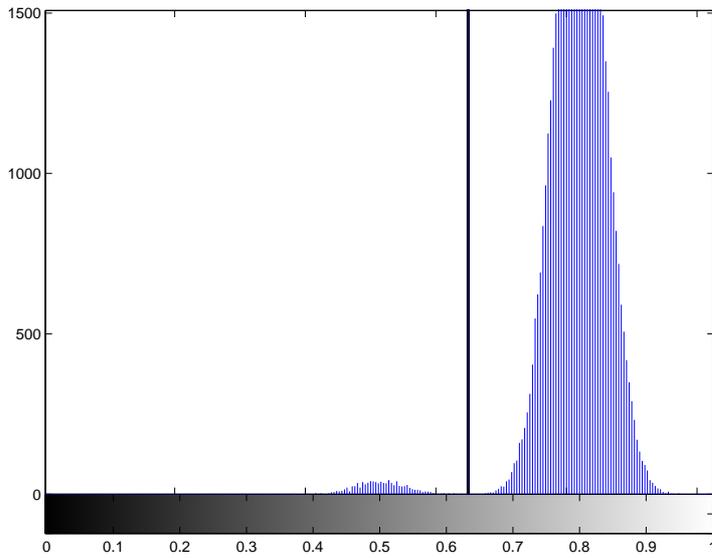


(c)

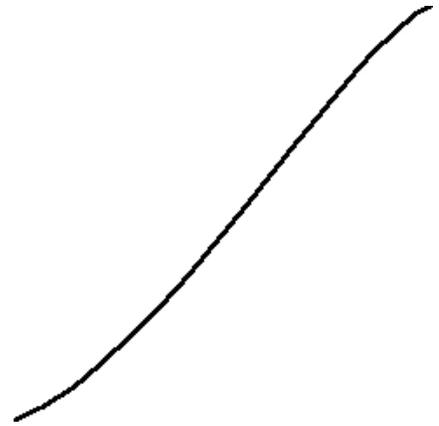


(d)

Figura 3.6: (a) e (c) Imagens de duas trajetórias de espessuras distintas, ou ainda de uma mesma trajetória em planos a diferentes distâncias do plano de retina. À exceção da da proporção p_T de *pixels* correspondentes à trajetória para o número total de *pixels*, os parâmetros estatísticos das imagens são idênticos: $\mu_1 = 0,5$, $\mu_2 = 0,8$, $\sigma_1^2 = 1,45 \cdot 10^{-3}$, $\sigma_2^2 = 1,45 \cdot 10^{-3}$. (b) e (d) Histogramas das duas imagens, onde se observa a variação entre os dois casos da proporção p_T (0,0166 para a imagem a, 0,6247 para a imagem c).

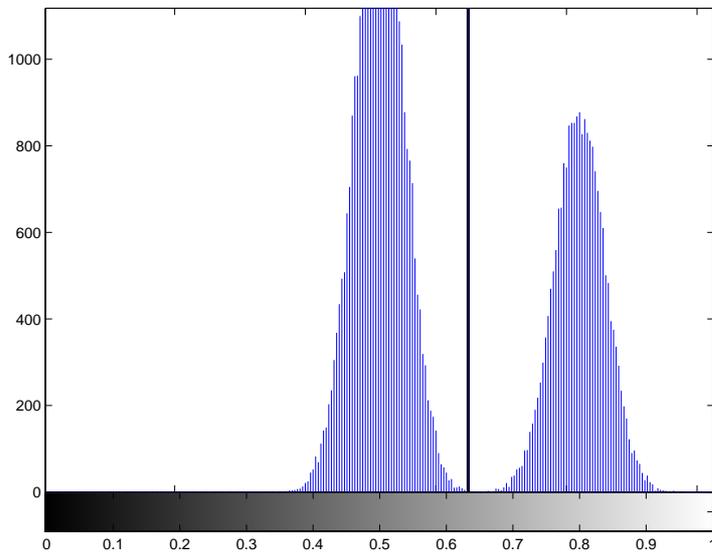


(a)

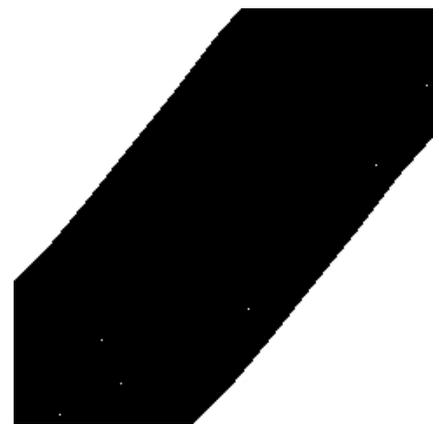


(b)

Figura 3.7: (a) Histograma da imagem 3.6-(a), com destaque para o limiar de binarização $L = 0,6303$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado o limiar apropriado, todos os *pixels* foram classificados corretamente.



(a)



(b)

Figura 3.8: (a) Histograma da imagem 3.6-(c), com destaque para o limiar de binarização $L = 0,6303$ obtido para a imagem anterior — 3.6-(a). (b) Imagem binarizada utilizando o limiar destacado. Como não foi utilizado o limiar apropriado para a proporção de trajetória para fundo na imagem (p_T), diversos *pixels* foram classificados incorretamente.

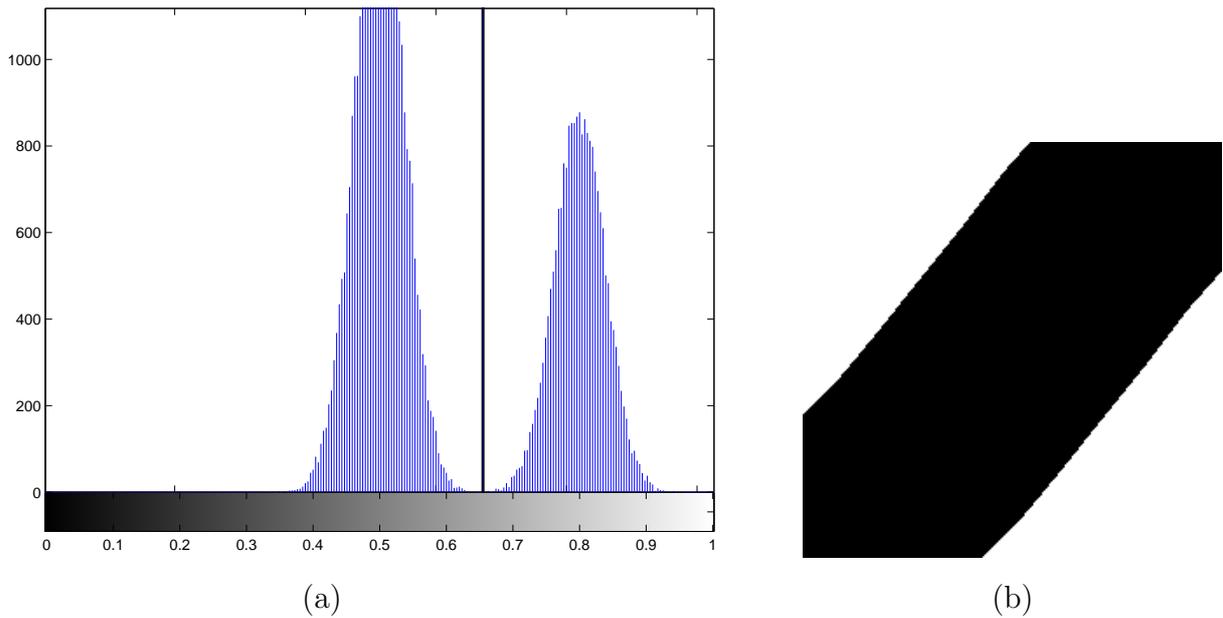
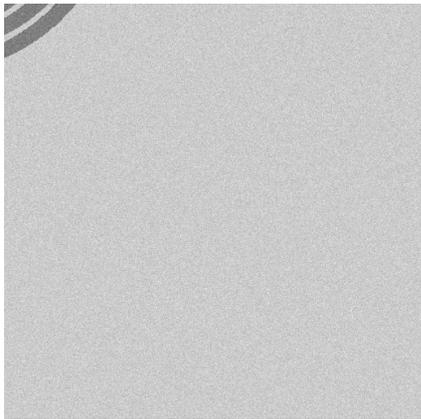


Figura 3.9: (a) Histograma da imagem 3.6-(c), com destaque para o limiar de binarização $L = 0,6525$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado desta vez o limiar apropriado, todos os *pixels* foram classificados corretamente.

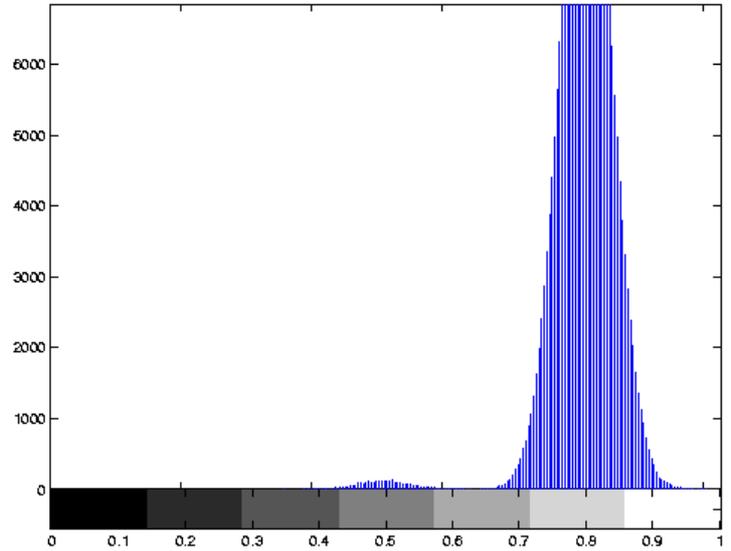
com a imagem binarizada correspondente, resulta em diversas classificações incorretas no segundo caso, conforme se observa na figura 3.12. O limiar $L = 0,6506$, calculado segundo (3.73) levando em conta as características da imagem 3.10-(c), leva à classificação perfeita — figura 3.13 — do segundo trecho da trajetória.

Estes resultados mostram que, a menos que as distribuições da trajetória e do fundo sejam bastante afastadas, ou seja, tenham expectâncias muito distintas relativamente aos desvios-padrões, o que garantiria uma ampla faixa de valores de L adequados à binarização, deve-se atualizar este limiar a cada trecho percorrido da trajetória, conferindo maior robustez à limiarização das imagens adquiridas.

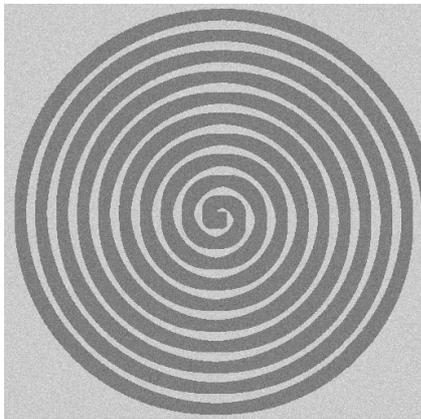
O que se observou, no entanto, é que apenas a proporção de trajetória (p_T) varia consideravelmente entre imagens consecutivas, permanecendo praticamente constantes os demais parâmetros estatísticos. Considerando-se que há uma expressão matemática em forma fechada que relaciona aquela proporção às demais variáveis que descrevem a imagem, conforme será mostrado no capítulo ??, optou-se assim por determinar p_1 a cada nova imagem adquirida e as demais variáveis estatísticas, de cálculo mais complexo, a uma taxa inferior, garantindo a atualização do limiar de binarização a cada ciclo sem um aumento apreciável do esforço computacional.



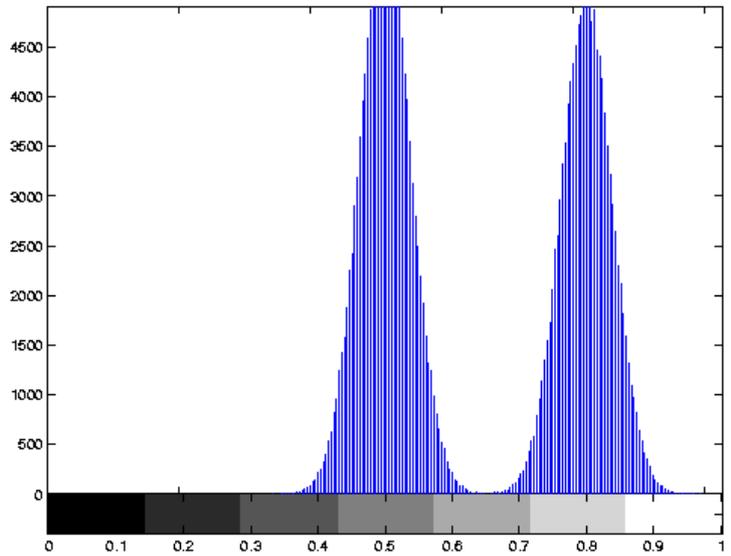
(a)



(b)



(c)



(d)

Figura 3.10: (a) e (c) Imagens de dois segmentos de comprimentos distintos de uma mesma trajetória. À exceção da da proporção p_T de *pixels* correspondentes à trajetória para o número total de *pixels*, os parâmetros estatísticos das imagens são idênticos: $\mu_1 = 0,5$, $\mu_2 = 0,8$, $\sigma_1^2 = 1,5 \cdot 10^{-3}$, $\sigma_2^2 = 1,5 \cdot 10^{-3}$. (b) e (d) Histogramas das duas imagens, onde se observa a variação de um segmento para o outro da proporção p_T (0,0107 para a imagem a, 0,5334 para a imagem c).

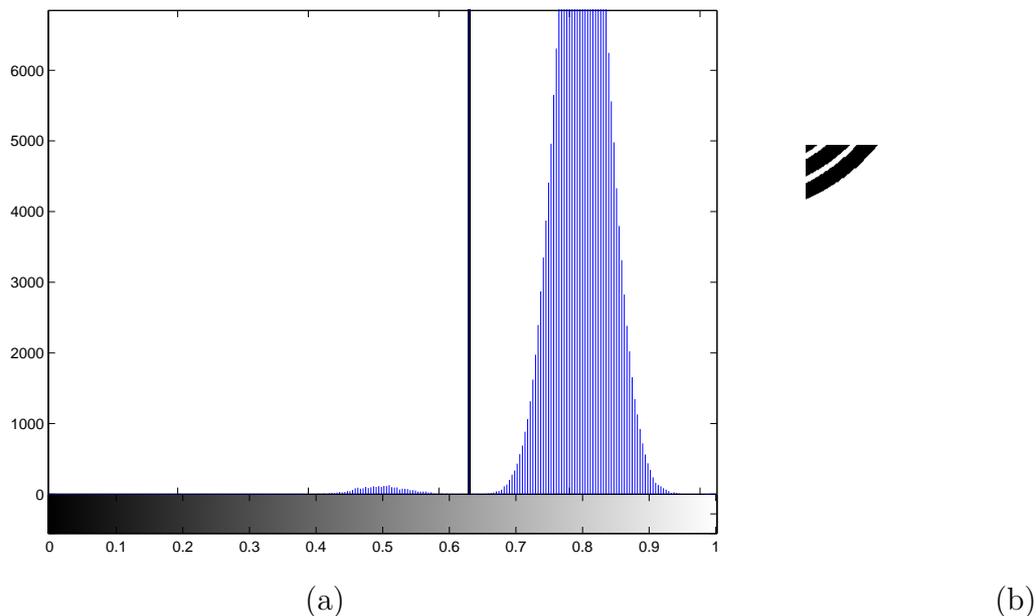


Figura 3.11: (a) Histograma da imagem 3.10-(a), com destaque para o limiar de binarização $L = 0,6274$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado o limiar apropriado, todos os *pixels* foram classificados corretamente.

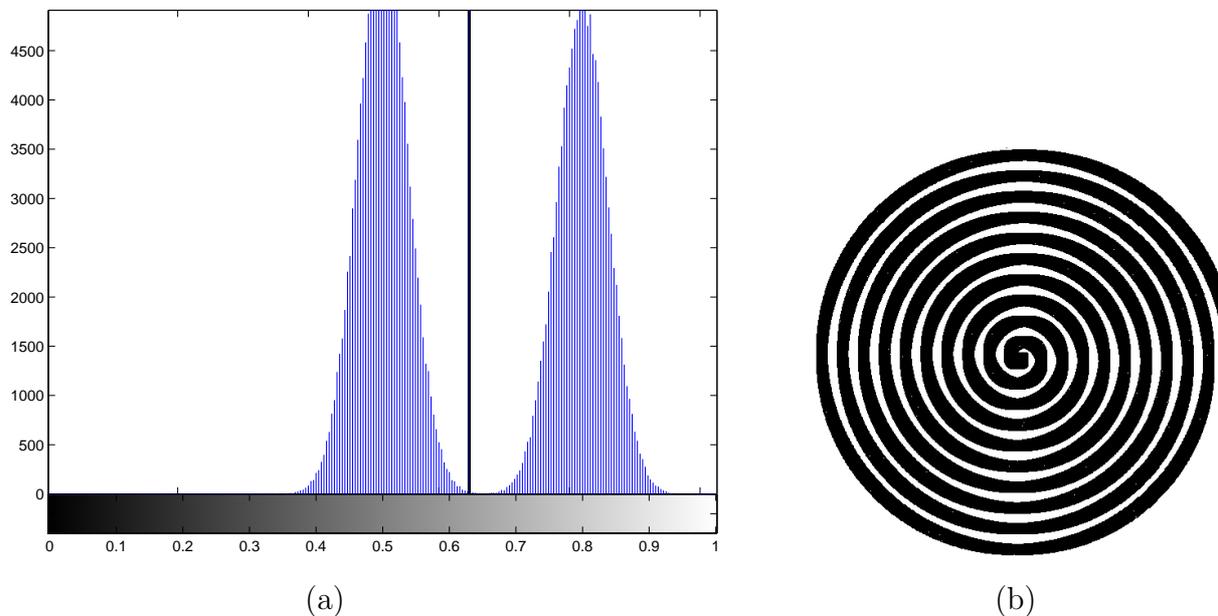
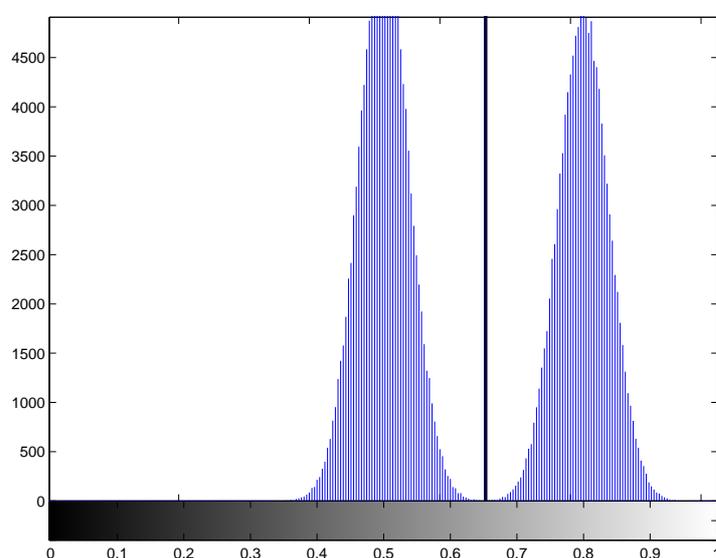
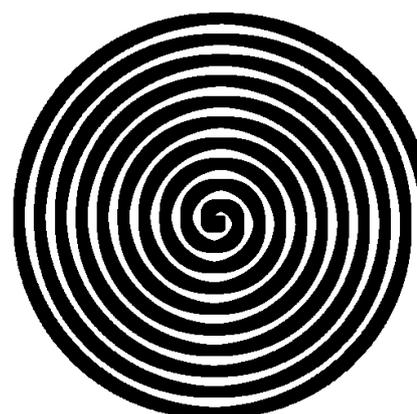


Figura 3.12: (a) Histograma da imagem 3.10-(c), com destaque para o limiar de binarização $L = 0,6274$ obtido para a imagem anterior — 3.10-(a). (b) Imagem binarizada utilizando o limiar destacado. Como não foi utilizado o limiar apropriado para a proporção de trajetória para fundo na imagem (p_T), diversos *pixels* foram classificados incorretamente.



(a)



(b)

Figura 3.13: (a) Histograma da imagem 3.10-(c), com destaque para o limiar de binarização $L = 0,6506$ calculado segundo a equação (3.73) considerando a proporção de trajetória para fundo (p_T) apropriada para esta imagem. (b) Imagem binarizada utilizando o limiar destacado. Como foi utilizado desta vez o limiar apropriado, todos os *pixels* foram classificados corretamente.

Outro aspecto que não deve ser desconsiderado em aplicações práticas da limiarização diz respeito às condições de iluminação da cena a ser analisada. Como se mostrou na seção 3.2, uma iluminação não-uniforme pode alterar significativamente o histograma da imagem, que deixa de ser caracterizada pelo modelo estatístico apresentado. Nesta situação, inviabiliza-se a aplicação direta do processo de limiarização aqui descrito.

Este problema sugere que se efetue um controle rigoroso das condições de iluminação em que é realizada a aquisição das imagens, de forma que todos os pontos da cena sejam iluminados aproximadamente com a mesma intensidade. No entanto, em muitas situações práticas não se tem controle da iluminação, ou a influência de fontes externas, aleatória, impede a uniformidade da iluminação independentemente da configuração da fonte principal. Assim, para garantir uma maior aplicabilidade dos algoritmos apresentados, optou-se por empregar técnicas de compensação de iluminação, cujo objetivo é obter, a partir das imagens reais, aquelas que seriam obtidas da mesma cena sob condições ideais de iluminação. As técnicas de compensação da iluminação não-uniforme investigadas são apresentadas no capítulo 5.

Capítulo 4

DETERMINAÇÃO DOS PARÂMETROS ESTATÍSTICOS DAS DISTRIBUIÇÕES GAUSSIANAS DA TRAJETÓRIA E DO FUNDO PARA CÁLCULO DO LIMIAR ÓTIMO DE BINARIZAÇÃO

4.1 DESCRIÇÃO DO PROBLEMA

A equação (3.73) apresenta, em forma fechada, a solução do problema de determinação do limiar ótimo L , supostas gaussianas as distribuições f_1 e f_2 . No entanto, devem ser conhecidos os parâmetros estatísticos destas distribuições, $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2$, e a proporção de objeto e de fundo na imagem, p_1 e $1 - p_1$. Estes parâmetros não são, em geral, conhecidos de antemão, e de um algoritmo absolutamente automático de determinação do limiar de binarização se espera que permita seu cálculo a partir unicamente da imagem de entrada. O método proposto neste capítulo objetiva precisamente o cálculo das 5 variáveis envolvidas.

Apesar de existirem estimadores para a média e para a variância associadas a um sinal, eles apenas permitem determinar, de forma direta, os parâmetros estatísticos globais da imagem, no caso correspondentes à densidade de mistura f (3.51), não às distribuições isoladas f_1 e f_2 , como se faz necessário. Uma possível abordagem para o cálculo de todos os parâmetros envolvidos consiste em definir uma função de erro quadrático (e_q) entre o histograma observado do sinal e a densidade de mistura. Este erro depende das variáveis de interesse $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2$ e p_1 , que devem ser escolhidas de forma a minimizá-lo. Assim:

$$e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1) = \sum_{n=1}^N \left\{ \left[p_1 f_g(\mu_1, \sigma_1^2, x_n) + (1 - p_1) f_g(\mu_2, \sigma_1^2, x_n) \right] - h(x_n) \right\}^2, \quad (4.1)$$

onde N é o número de níveis de quantização do sinal, $h(x_n)$ é a frequência relativa do n -ésimo nível (x_n) e $f_g(\mu, \sigma^2, x)$ é a distribuição gaussiana de média μ e variância σ^2 avaliada

no ponto x .

Nesta abordagem, o problema corresponde a determinar um ponto de mínimo da função e_q de 5 variáveis. Os pontos de mínimo de uma função contínua são aqueles em que as derivadas parciais de primeira ordem são nulas e as de segunda ordem são positivas [1]. Assim, uma alternativa para o cálculo dos parâmetros de interesse consiste em obter as raízes do sistema

$$\begin{aligned} \frac{\partial e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial \mu_1} &= \frac{\partial e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial \mu_2} = \frac{\partial e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial (\sigma_1^2)} = \\ &= \frac{\partial e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial (\sigma_2^2)} = \frac{\partial e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial p_1} = 0 \end{aligned} \quad (4.2)$$

sujeitas às inequações

$$\begin{aligned} \frac{\partial^2 e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial \mu_1^2} > 0, \quad \frac{\partial^2 e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial \mu_2^2} > 0, \quad \frac{\partial^2 e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial (\sigma_1^2)^2} > 0, \\ \frac{\partial^2 e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial (\sigma_2^2)^2} > 0, \quad \frac{\partial^2 e_q(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, p_1)}{\partial p_1^2} > 0. \end{aligned} \quad (4.3)$$

Entretanto, as derivadas em (4.2) conduzem, no caso das distribuições f_1 e f_2 serem gaussianas, a equações transcendentais cuja solução requer grande esforço computacional [8], sobretudo considerando o caso típico de imagens com 256 níveis de quantização.

Neste capítulo, propõe-se um algoritmo mais simples para a determinação automática dos parâmetros estatísticos descritos. A seção 4.2 apresenta inicialmente o equacionamento necessário à solução proposta, seguido da descrição das etapas que compõem o método. A seção 4.3 analisa os resultados da aplicação do algoritmo à determinação dos limiares de binarização de imagens reais e sintéticas, estas favorecendo a comparação direta entre os parâmetros calculados e os empregados em sua geração. Finalmente, a seção ?? apresenta a conclusão desta análise e uma avaliação da aplicabilidade do algoritmo a sistemas reais baseados em imagens binárias, como é o caso de manipuladores robóticos guiados por servo-visão aqui pesquisados.

4.2 DESCRIÇÃO DO ALGORITMO PROPOSTO

A solução proposta para a determinação automática dos parâmetros de interesse se baseia no uso de estimadores da média e da variância globais de sinais. A seção 4.2.1 aborda a relação entre estas grandezas e os parâmetros de interesse, e sugere de que maneira podem ser empregados para calculá-los. A seção 4.2.2 descreve a metodologia utilizada durante o particionamento da imagem, procedimento cuja necessidade é mostrada já em 4.2.1. As seções 4.2.3 e 4.2.4 descrevem por fim as duas etapas em que se divide o algoritmo.

4.2.1 Equacionamento

Inicialmente, considere-se a relação entre a média (μ) e a variância (σ^2) globais e os parâmetros que se deseja obter. Da definição da média [25], tem-se:

$$\mu = \int_{-\infty}^{\infty} x \cdot f(x) dx. \quad (4.4)$$

Para o caso da densidade de mistura (3.51), segue:

$$\mu = \int_{-\infty}^{\infty} x [p_1 f_1(x) + (1 - p_1) f_2(x)] dx = p_1 \int_{-\infty}^{\infty} x f_1(x) dx + (1 - p_1) \int_{-\infty}^{\infty} x f_2(x) dx \quad (4.5)$$

As duas últimas integrais correspondem às expressões para os valores esperados de f_1 e f_2 . Assim:

$$\mu = p_1 \mu_1 + (1 - p_1) \mu_2, \quad (4.6)$$

o que significa que o valor esperado global da imagem é a média ponderada dos valores esperados que se deseja calcular, tendo como pesos as proporções de objeto (p_1) e de fundo ($1 - p_1$) na imagem.

Da definição de variância [25], por sua vez, tem-se:

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx, \quad (4.7)$$

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 [p_1 f_1(x) + (1 - p_1) f_2(x)] dx, \quad (4.8)$$

que se reduz a:

$$\sigma^2 = p_1 [\sigma_1^2 + (\mu - \mu_1)^2] + (1 - p_1) [\sigma_2^2 + (\mu - \mu_2)^2]. \quad (4.9)$$

Nas equações (4.6) e (4.9), a média μ e a variância σ^2 globais da imagem I podem ser obtidas dos estimadores [11]:

$$\mu = \frac{1}{L \cdot C} \sum_{i=1}^L \sum_{j=1}^C I(i, j) \text{ e} \quad (4.10)$$

$$\sigma^2 = \frac{1}{L \cdot C - 1} \sum_{i=1}^L \sum_{j=1}^C (I(i, j) - \mu)^2, \quad (4.11)$$

onde L e C são o número de linhas e de colunas de I , respectivamente.

As equações (4.6) e (4.9) relacionam os parâmetros μ_1 , μ_2 , σ_1^2 , σ_2^2 e p_1 a serem calculados a μ e σ^2 , obtidos diretamente de (4.10) e (4.11). Entretanto, dispõe-se aqui de apenas 2 equações para 5 incógnitas, e o sistema é subdeterminado. Na primeira etapa do método aqui proposto, divide-se o suporte da imagem em um número específico de regiões, sendo

calculadas a média $\mu(r)$ e a variância $\sigma^2(r)$ em cada região r ; assim, as relações (4.6) e (4.9) permanecem válidas $\forall r$. A vantagem neste caso é que, embora a proporção $p_1(r)$ varie de região para região, os parâmetros $\mu_1, \sigma_1^2, \mu_2, \sigma_2^2$, intrínsecos às distribuições f_1 e f_2 , permanecem inalterados. Em outras palavras, cada nova região analisada contribui ao sistema com 2 novas equações ao passo que acrescenta apenas uma incógnita — $p_1(r)$. Conclui-se que, se R é o número total de regiões, há $4 + R$ incógnitas e $2R$ equações do tipo:

$$\begin{cases} \mu(r) = p_1(r)\mu_1 + (1 - p_1(r))\mu_2 & (a) \\ \sigma^2(r) = p_1(r) [\sigma_1^2 + (\mu(r) - \mu_1)^2] + (1 - p_1(r)) [\sigma_2^2 + (\mu(r) - \mu_2)^2] & (b) \end{cases} \quad (4.12)$$

Trata-se, no entanto, de um sistema de equações não-lineares. Na primeira etapa do algoritmo, propõe-se uma solução numérica do problema. Entretanto, conforme será abordado na seção 4.2.3, é mais conveniente expressar a relação (4.12-b) em termos dos momentos de segunda ordem $\lambda(r)$ de cada região, em substituição às variâncias. Da definição do momento de segunda ordem:

$$\lambda = \int_{-\infty}^{\infty} x^2 \cdot f(x) dx. \quad (4.13)$$

Para a densidade de mistura (3.51), obtém-se, em cada região r :

$$\lambda(r) = \int_{-\infty}^{\infty} x^2 \cdot [p_1(r)f_1(x) + (1 - p_1(r))f_2(x)] dx, \quad (4.14)$$

e o sistema (4.12) se reduz a:

$$\begin{cases} \mu(r) = p_1(r)\mu_1 + (1 - p_1(r))\mu_2 & (a) \\ \lambda(r) = p_1(r)\lambda_1 + (1 - p_1(r))\lambda_2 & (b) \end{cases} \quad (4.15)$$

Em (4.15-b), os momentos $\lambda(r)$ podem ser calculados a partir das estimativas de $\mu(r)$ e $\sigma^2(r)$:

$$\lambda(r) = \sigma^2(r) + [\mu(r)]^2. \quad (4.16)$$

Inversamente, uma vez calculados os momentos λ_1 e λ_2 associados às distribuições isoladas f_1 e f_2 , determinam-se as variâncias σ_1^2 e σ_2^2 a partir da mesma relação.

Obtidos μ_1, μ_2, σ_1^2 e σ_2^2 , todos os parâmetros necessários ao cálculo do limiar ótimo de binarização ficam determinados, uma vez que, conforme será exposto na seção 4.2.3, há uma expressão simples em forma fechada que relaciona p_1 àquelas grandezas e à média e variância globais da imagem.

4.2.2 Particionamento da imagem

Optou-se, por simplicidade, em dividir a imagem em regiões retangulares que não se sobrepõem, mas que são contíguas, formando, assim, uma partição do suporte da imagem. Em particular, note-se que, no sistema (4.15), há mais equações do que incógnitas se $R > 4$. Entretanto, o número de regiões não pode ser arbitrariamente grande, pois quanto maior R , menor o número de *pixels* em cada região, o que diminui a precisão das estimativas de $\mu(r)$ e $\sigma^2(r)$ a partir de (4.10) e (4.11). Observou-se que a divisão em $R_j = 3$ regiões na horizontal por $R_i = 3$ regiões na vertical, num total de $R = R_i \cdot R_j = 9$ regiões, foi suficiente para a convergência do algoritmo que será descrito a seguir, nos casos analisados.

A figura 4.1 ilustra o tipo de partição adotado; observe-se que, após a partição, são estimados o valor esperado $\mu(r)$ e a variância $\sigma^2(r)$ globais associados a cada região r , com uso dos estimadores (4.10) e (4.11).

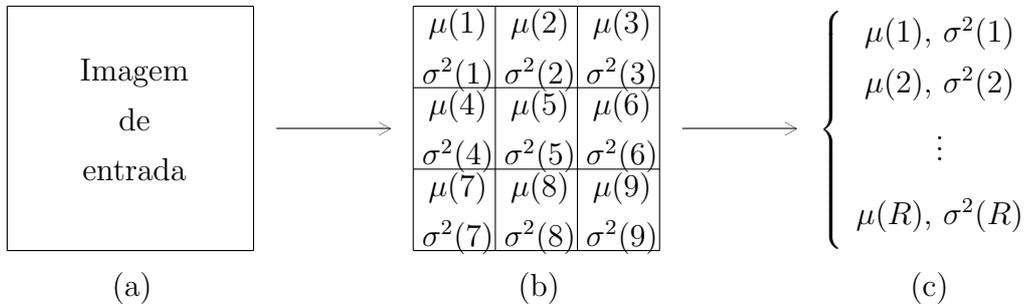


Figura 4.1: Divisão da imagem de entrada (a) em regiões retangulares contíguas sem sobreposição (b), para cálculo das médias e variâncias (c) que serão utilizadas na determinação dos parâmetros estatísticos das distribuições de probabilidade do objeto e do fundo.

4.2.3 Primeira etapa

Na proposta de uma solução iterativa ao sistema (4.12) surge o problema de se determinar uma estimativa inicial antes do início das iterações, seja ela referente às médias e variâncias associadas às distribuições f_1 e f_2 ou às proporções de objeto em cada região r da imagem, $p_1(r)$. No caso das médias μ_1 e μ_2 uma informação ainda não considerada pode ser utilizada como este objetivo: observe-se, da equação (4.15-a), que a média global $\mu(r)$ associada a cada região é a média ponderada de μ_1 e μ_2 . Desta forma:

$$\mu_1 \leq \mu(r) \leq \mu_2, \forall r, \quad (4.17)$$

se for arbitrado μ_1 como o menor valor entre μ_1 e μ_2 .

Como as desigualdades (4.17) se mantêm em todas as regiões, tem-se:

$$\mu_1 \leq \min_{1 \leq r \leq R} [\mu(r)], \quad (4.18)$$

$$\mu_2 \geq \max_{1 \leq r \leq R} [\mu(r)]. \quad (4.19)$$

Uma possível estimativa inicial de μ_1 e μ_2 consiste, pois, no menor e no maior valor dentre todas as médias estimadas das R regiões da imagem, respectivamente.

Estimativas deste tipo não são tão imediatas para as variâncias σ_1^2 e σ_2^2 . No entanto, elas surgem também para os momentos de segunda ordem λ_1 e λ_2 , se eles forem utilizados em substituição às variâncias.

Analogamente às relações obtidas para as médias μ_1 e μ_2 , tem-se, da equação (4.15-b):

$$\lambda_1 \leq \min_{1 \leq r \leq R} [\lambda(r)], \quad (4.20)$$

$$\lambda_2 \geq \max_{1 \leq r \leq R} [\lambda(r)]. \quad (4.21)$$

Uma observação adicional acerca do momento de ordem 2 pode ainda ser útil na escolha da estimativa inicial de λ_2 . Sabe-se que o momento de ordem 2 associado a uma distribuição é igual à variância mais o quadrado do valor esperado [25], ou seja, para a distribuição f_2 :

$$\lambda_2 = \mu_2^2 + \sigma_2^2. \quad (4.22)$$

Por ser a variância σ_2^2 não-negativa, segue:

$$\lambda_2 \geq \mu_2^2. \quad (4.23)$$

Considerando-se também a desigualdade (4.21):

$$\lambda_2 \geq \max(\mu_2^2, \max_{1 \leq r \leq R} [\lambda(r)]), \quad (4.24)$$

ou seja, a estimativa inicial de λ_2 pode ser tomada como o maior valor dentre μ_2^2 e todos os momentos $\lambda(r)$, $1 \leq r \leq R$.

Desta forma, dispõe-se também de uma estimativa inicial para λ_1 e λ_2 , desde que se possa estimar os momentos $\lambda(r)$ associados a cada região r . Felizmente, esta grandeza é obtida diretamente das estimativas $\mu(r)$ e $\sigma^2(r)$, conforme a equação (4.16).

Também as variâncias σ_1^2 e σ_2^2 , necessárias ao cálculo do limiar L em (3.73), podem ser obtidas de λ_1 e μ_1 , λ_2 e μ_2 , segundo a mesma relação.

Isto sugere que, em vez de se calcular μ_1 , μ_2 , σ_1^2 , σ_2^2 a partir diretamente das médias e variâncias em cada região da imagem, sejam inicialmente calculados os momentos de ordem 2, $\lambda(r)$, a partir daquelas grandezas, e, só então, μ_1 , μ_2 , λ_1 e λ_2 , a partir do sistema (4.15). Somente após o cálculo destes parâmetros, seriam então estimadas σ_1^2 e σ_2^2 .

Adotou-se uma solução iterativa a este sistema de equações não-lineares, baseada na observação de que, supostos conhecidos os valores de $p_1(r)$, $1 \leq r \leq R$, tem-se um sistema linear nas incógnitas μ_1 e μ_2 a partir da relação (a) em (4.15), e outro sistema linear,

em λ_1 e λ_2 , a partir da relação (b) em (4.15). Da mesma forma, supostos conhecidos μ_1 , μ_2 , λ_1 , λ_2 , a equação (4.15) corresponde a um único sistema linear nas incógnitas $p_1(r)$, $1 \leq r \leq R$.

Estes sistemas podem ser escritos respectivamente como:

$$\begin{cases} \mathbf{A} \cdot \mathbf{m} = \mathbf{B} & \text{(a)} \\ \mathbf{A} \cdot \mathbf{l} = \mathbf{C} & \text{(b)} \end{cases}, \quad (4.25)$$

onde:

$$\mathbf{A} = \begin{bmatrix} p_1(1) & 1 - p_1(1) \\ p_1(2) & 1 - p_1(2) \\ \vdots & \vdots \\ p_1(R) & 1 - p_1(R) \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mu(1) \\ \mu(2) \\ \vdots \\ \mu(R) \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \lambda(1) \\ \lambda(2) \\ \vdots \\ \lambda(R) \end{bmatrix}, \quad (4.26)$$

$$\mathbf{m} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \text{e} \quad \mathbf{l} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix},$$

e:

$$\mathbf{D} \cdot \mathbf{p} = \mathbf{E}, \quad (4.27)$$

com:

$$\mathbf{D}_{2R \times R} = \begin{bmatrix} \mu_1 - \mu_2 & 0 & 0 & \cdots & 0 \\ 0 & \mu_1 - \mu_2 & 0 & \cdots & 0 \\ 0 & 0 & \mu_1 - \mu_2 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \cdots & \mu_1 - \mu_2 \\ \lambda_1 - \lambda_2 & 0 & 0 & \cdots & 0 \\ 0 & \lambda_1 - \lambda_2 & 0 & \cdots & 0 \\ 0 & 0 & \lambda_1 - \lambda_2 & \cdots & 0 \\ & & \vdots & & \\ 0 & 0 & 0 & \cdots & \lambda_1 - \lambda_2 \end{bmatrix}, \quad (4.28)$$

$$\mathbf{p} = \begin{bmatrix} p_1(1) \\ p_1(2) \\ p_1(3) \\ \vdots \\ p_1(R) \end{bmatrix} \text{ e } \mathbf{E} = \begin{bmatrix} \mu(1) - \mu_2 \\ \mu(2) - \mu_2 \\ \mu(3) - \mu_2 \\ \vdots \\ \mu(R) - \mu_2 \\ \lambda(1) - \lambda_2 \\ \lambda(2) - \lambda_2 \\ \lambda(3) - \lambda_2 \\ \vdots \\ \lambda(R) - \lambda_2 \end{bmatrix}.$$

Esta abordagem sugere o seguinte método iterativo para resolução das equações (4.12), levando em conta as estimativas iniciais de μ_1 , μ_2 , λ_1 , λ_2 :

1. adotam-se ${}^0\mu_1 = \min_{1 \leq r \leq R} [\mu(r)]$, ${}^0\mu_2 = \max_{1 \leq r \leq R} [\mu(r)]$, ${}^0\lambda_1 = \min_{1 \leq r \leq R} [\lambda(r)]$ e ${}^0\lambda_2 = \max({}^0\mu_2^2, \max_{1 \leq r \leq R} [\lambda(r)])$ como estimativas iniciais de μ_1 , μ_2 , λ_1 e λ_2 ;
2. a partir de ${}^0\mu_1$, ${}^0\mu_2$, ${}^0\lambda_1$ e ${}^0\lambda_2$, calcula-se uma estimativa de \mathbf{p} pela solução do sistema linear (4.27);
3. a partir de \mathbf{p} estimado no item 2, obtêm-se novas estimativas de μ_1 e de μ_2 pela resolução do sistema linear (4.25)-(a), pelo método dos mínimos quadrados com a restrição de que as raízes sejam positivas;
4. a partir de \mathbf{p} estimado no item 2, obtêm-se novas estimativas de λ_1 e de λ_2 pela resolução do sistema linear (4.25)-(b), pelo método dos mínimos quadrados com a restrição:
$$\lambda_1 \leq \min_{1 \leq r \leq R} [\lambda(r)] \text{ e } \lambda_2 \geq \max(\mu_2^2, \max_{1 \leq r \leq R} [\lambda(r)]);$$

5. determina-se a diferença $\mathbf{d} = [\mu_1 \ \mu_2 \ \lambda_1 \ \lambda_2]^T - [{}^0\mu_1 \ {}^0\mu_2 \ {}^0\lambda_1 \ {}^0\lambda_2]^T$. Se o maior elemento de \mathbf{d} , em módulo, for maior do que um valor ϵ especificado, segundo um critério de parada, então adotam-se μ_1, μ_2, λ_1 e λ_2 como novos valores de ${}^0\mu_1, {}^0\mu_2, {}^0\lambda_1$ e ${}^0\lambda_2$ e reinicia-se o processo, a partir do item 2. Caso contrário, foi satisfeito o critério de parada.

Cálculo da Proporção p_1 de Objeto (Trajetória) na Imagem

Os métodos propostos na seção anterior permitem o cálculo de $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$. Para a determinação do limiar ótimo de binarização a partir da equação (3.73), resta apenas determinar p_1 .

Nas equações (4.6) e (4.9) constata-se que apenas este parâmetro permanece indeterminado. Como se trata de equações lineares nesta incógnita, uma alternativa é simplesmente isolá-la em qualquer uma das equações, o que fornece uma expressão em forma fechada para seu cálculo.

Em princípio, deveriam ser obtidos valores idênticos independentemente da equação a partir da qual se opte por calcular p_1 . No entanto, isto na prática não ocorre, em virtude dos procedimentos numéricos adotados no cálculo de todas as variáveis envolvidas. Optou-se, portanto, por aplicar no cálculo de p_1 o método dos mínimos quadrados, para satisfazer simultaneamente as duas equações com o menor erro quadrático possível. Reescrevendo (4.6) e (4.9):

$$\begin{cases} \delta_{1,2} \cdot p_1 = \delta_2 \\ (\alpha + \delta_1^2 - \delta_2^2) \cdot p_1 = \beta - \delta_2^2 \end{cases}, \quad (4.29)$$

onde $\alpha = \sigma_1^2 - \sigma_2^2$, $\beta = \sigma^2 - \sigma_2^2$, $\delta_{1,2} = \mu_1 - \mu_2$, $\delta_1 = \mu - \mu_1$ e $\delta_2 = \mu - \mu_2$.

Em formalismo matricial:

$$\begin{bmatrix} \delta_{1,2} \\ \alpha + \delta_1^2 - \delta_2^2 \end{bmatrix} \cdot p_1 = \begin{bmatrix} \delta_2 \\ \beta - \delta_2^2 \end{bmatrix}. \quad (4.30)$$

A solução pelo método dos mínimos quadrados é portanto:

$$p_1 = \left(\begin{bmatrix} \delta_{1,2} \\ \alpha + \delta_1^2 - \delta_2^2 \end{bmatrix}^T \cdot \begin{bmatrix} \delta_{1,2} \\ \alpha + \delta_1^2 - \delta_2^2 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} \delta_{1,2} \\ \alpha + \delta_1^2 - \delta_2^2 \end{bmatrix} \cdot \begin{bmatrix} \delta_2 \\ \beta - \delta_2^2 \end{bmatrix}, \quad (4.31)$$

o que conduz à solução em forma fechada aqui utilizada:

$$p_1 = \frac{\delta_{1,2} \cdot \delta_2 + [\alpha + \delta_1^2 - \delta_2^2] \cdot [\beta - \delta_2^2]}{\delta_{1,2}^2 + [\alpha + \delta_1^2 - \delta_2^2]^2}. \quad (4.32)$$

4.2.4 Segunda etapa

A primeira etapa do método proposto, apresentada na seção anterior, fornece uma primeira estimativa do limiar de binarização. Uma segunda etapa, que passa agora a ser descrita, se faz necessária à obtenção do valor ótimo deste limiar.

O objetivo é calcular com maior precisão as médias e variâncias associadas às distribuições de probabilidade do objeto (trajetória) e do fundo do que as fornecidas pela resolução ao sistema (4.12) ou (4.15) proposta. A diferença é que, neste estágio, já se dispõe de uma estimativa do limiar de binarização, ou seja, já é possível uma separação entre objeto e fundo na imagem, ainda que alguns *pixels* sejam classificados incorretamente. A figura 4.2 representa o histograma de uma imagem hipotética de um objeto sobre um fundo, com destaque para uma estimativa inicial do limiar de binarização, ainda distinta do limiar ótimo.

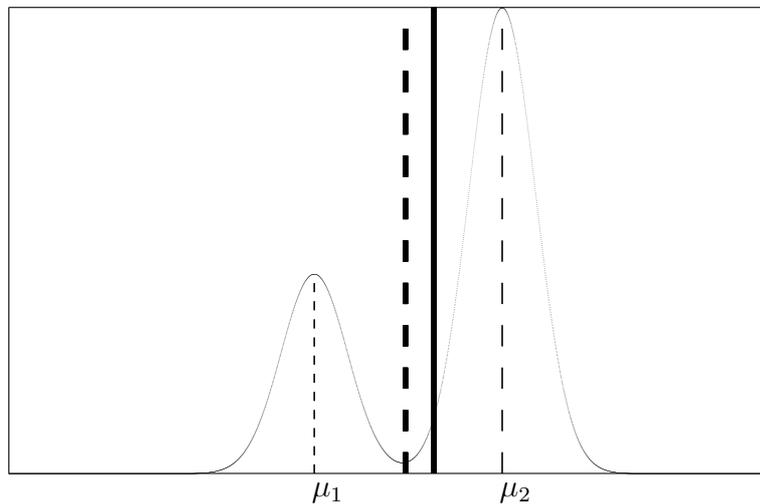


Figura 4.2: Histograma de uma imagem de um objeto sobre um fundo, como destaque para a estimativa inicial do limiar fornecida pelo método descrito na seção 4.2.3 (linha grossa contínua), ainda distinta do limiar ótimo (linha grossa seccionada).

A segunda etapa do método é baseada na separação das duas distribuições a partir de estimativas do limiar de binarização, a primeira das quais é obtida na etapa descrita na seção 4.2.3. O que se observou é que a aplicação dos estimadores da média (4.10) e da variância (4.11) apenas aos *pixels* associados a cada distribuição, segundo a binarização utilizando a primeira estimativa do limiar, conduziu a valores mais precisos daquelas grandezas do que os obtidos na primeira etapa e então utilizados no cálculo deste limiar. Conseqüentemente, uma segunda estimativa do limiar, mais próxima do valor ótimo, pôde ser obtida a partir da equação (3.73). Como a aplicação deste limiar conduz a menos classificações incorretas do que a do anterior, as médias e variâncias obtidas para cada distribuição a partir dos estimadores (4.10) e (4.11) são ainda mais precisas. Isto sugere a sucessiva aplicação dos estimadores aos *pixels* que, segundo os limiares calculados, pertençam a cada distribuição, e a atualização daqueles limiares com base nas novas médias

e variâncias obtidas. Já que a cada vez menos *pixels* são classificados incorretamente, o limiar calculado converge para o valor ótimo.

O processo iterativo fica assim esquematizado:

1. calcula-se a primeira estimativa L_0 do limiar de binarização, utilizando-se o método apresentado na seção 4.2.3;
2. determinam-se estimativas de μ_1 e σ_1^2 pela aplicação dos estimadores (4.10) e (4.11) aos *pixels* com nível de cinza inferior a L_0 ;
3. determinam-se estimativas de μ_2 e σ_2^2 pela aplicação dos estimadores (4.10) e (4.11) aos *pixels* com nível de cinza superior a L_0 ;
4. a partir de μ_1, σ_1^2, μ_2 e σ_2^2 , determina-se p_1 a partir da equação (4.32);
5. calcula-se uma nova estimativa do limiar L a partir da equação (3.73);
6. determina-se a diferença $d = L - L_0$. Se d for, em módulo, maior do que um valor ϵ especificado, segundo um critério de parada, então adota-se L como novo valor de L_0 e reinicia-se o processo, a partir do item 2. Caso contrário, foi satisfeito o critério de parada.

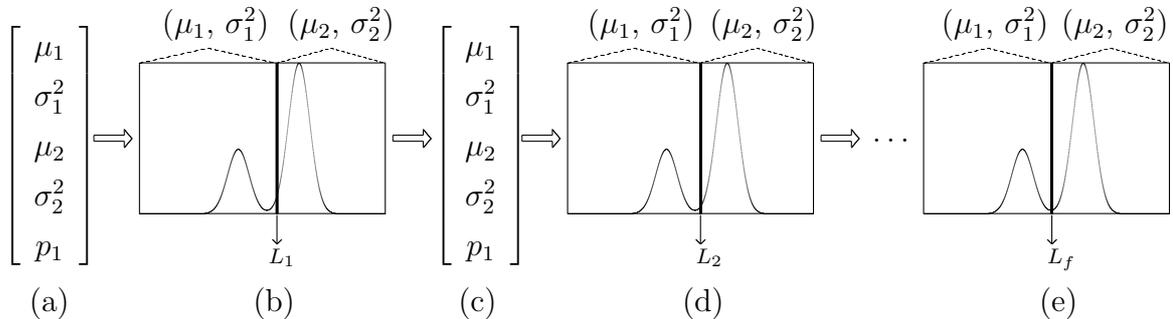


Figura 4.3: Segunda etapa do algoritmo proposto para cálculo automático dos parâmetros estatísticos das distribuições associadas à trajetória e ao fundo. Na seqüência de histogramas, observa-se que a cada iteração é obtida uma estimativa mais precisa do limiar ótimo de binarização.

A figura 4.4 esquematiza, em um diagrama de blocos, o algoritmo proposto, em suas duas etapas constituintes.

4.3 RESULTADOS DE TESTES DE AVALIAÇÃO DO ALGORITMO PROPOSTO

O algoritmo aqui proposto foi aplicado a imagens sintéticas e reais. As tabelas 4.1 e 4.2 mostram os parâmetros teóricos utilizados na geração de duas imagens sintéticas de exemplo, bem como os parâmetros correspondentes obtidos pelo algoritmo. Observa-se, nestas tabelas, que o método permitiu determinar todos os parâmetros com erro inferior a 3%.

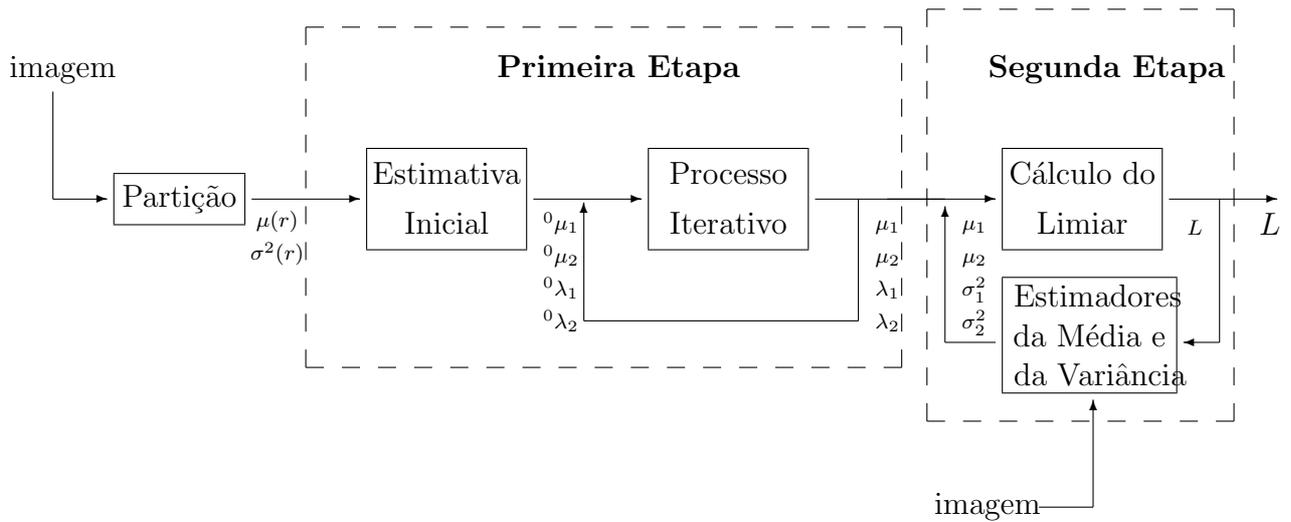


Figura 4.4: Diagrama de blocos do algoritmo proposto para cálculo automático dos parâmetros estatísticos das distribuições associadas à trajetória e ao fundo, para determinação do limiar ótimo de binarização L .

As figuras 4.5 e 4.6 ilustram o resultado da binarização destas imagens: em (a), apresenta-se uma imagem sintética degradada com ruído gaussiano aditivo; em (b), o seu histograma ressaltando-se o limiar calculado e utilizado na binarização; e em (c), a imagem binarizada obtida. Como o limiar obtido é bastante próximo do valor teórico, e como as duas distribuições não se sobrepõem, todos os *pixels* da imagem são classificados corretamente.

Como exemplo da aplicação do algoritmo a uma imagem real, apresenta-se na figura 4.7-(a) a imagem de uma trajetória sobre um fundo, juntamente com seu histograma com destaque para o limiar de binarização obtido (figura 4.7-(b)) e com o resultado da binarização (figura 4.7-(c)). Novamente as duas distribuições não se sobrepõem, e como o limiar calculado se situa entre elas, todos os *pixels* são classificados corretamente.

Por fim, apresenta-se, na figura 4.8-(a), uma imagem sintética altamente degradada como com ruído gaussiano aditivo, de tal forma que passa a haver sobreposição entre as duas distribuições. Neste caso, se observa (figura 4.8-(b)) que nem mesmo o limiar teórico é capaz de separar corretamente as duas distribuições, de forma que vários *pixels* são classificados incorretamente durante a binarização (figura 4.8-(c)). Entretanto, observou-se neste caso que a utilização de um filtro que reduza o ruído aditivo pode permitir separar as duas distribuições e utilizar com sucesso a mesma técnica para a binarização. O teste foi realizado utilizando o filtro de Wiener. Na figura 4.9-(a), mostra-se o resultado da aplicação deste filtro à imagem 4.8-(a). Em 4.9-(b), é apresentado o histograma correspondente, onde se nota que houve a redução do ruído aditivo e a conseqüente separação das duas distribuições. O novo limiar de binarização, calculado com o método aqui proposto, situa-se entre as duas distribuições. Em 4.9-(c), é apresentada a imagem binarizada

a partir deste limiar, com todos os *pixels* classificados corretamente.

Parâmetro	Teórico	Calculado	Erro (%)
μ_1	0,1000	0,0995	0,53
μ_2	0,2000	0,2001	0,05
$\sigma_1^2 \cdot 10^4$	2,0000	1,9597	2,02
$\sigma_2^2 \cdot 10^4$	2,0000	1,9923	0,39
λ_1	0,0102	0,0101	1,08
λ_2	0,0402	0,0402	0,09
p_1	0,0368	0,0368	0,05
<i>limiar (L)</i>	0,1435	0,1431	0,21

Tabela 4.1: Comparação entre os parâmetros teóricos utilizados na geração da imagem sintética 1 (figura 4.5) e os calculados.

Parâmetro	Teórico	Calculado	Erro (%)
μ_1	0,7000	0,6999	0,01
μ_2	0,8000	0,8001	0,01
$\sigma_1^2 \cdot 10^4$	1,7000	1,6903	0,57
$\sigma_2^2 \cdot 10^4$	1,5000	1,5027	0,18
λ_1	0,4902	0,4901	0,02
λ_2	0,6402	0,6403	0,02
p_1	0,1687	0,1687	0,00
<i>limiar (L)</i>	0,7489	0,7488	0,01

Tabela 4.2: Comparação entre os parâmetros teóricos utilizados na geração da imagem sintética 2 (figura 4.6) e os calculados.

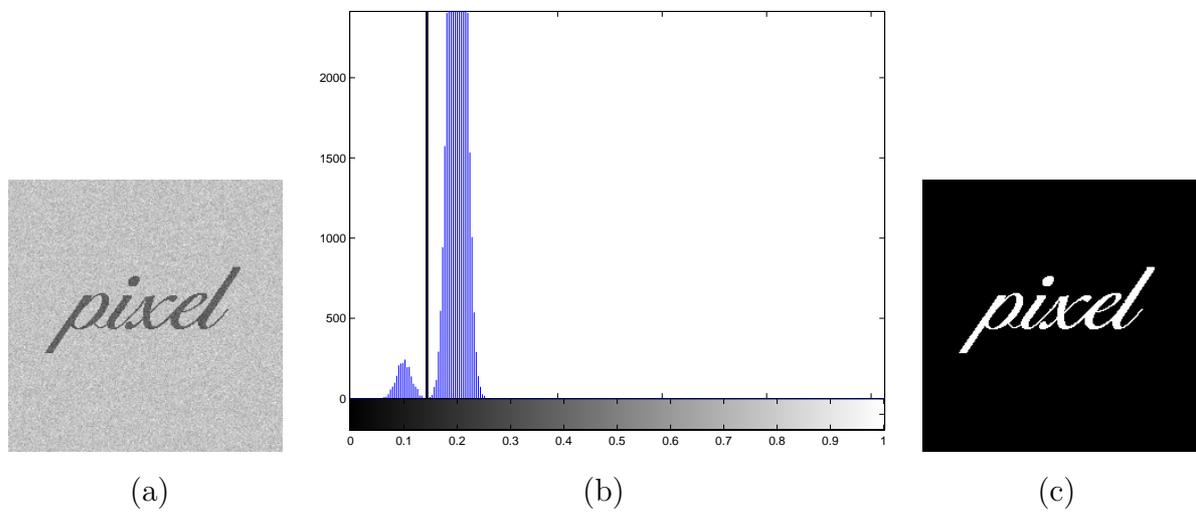


Figura 4.5: (a) Exemplo de uma imagem sintética com 2 regiões de reflectância afetadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar de binarização calculado. (c) Imagem binarizada a partir do limiar calculado.

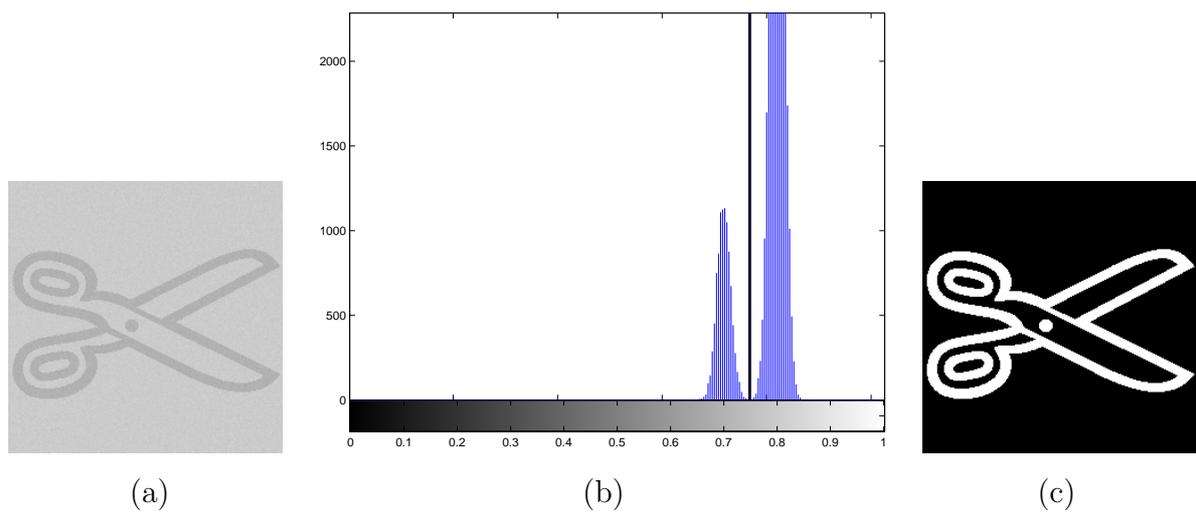


Figura 4.6: (a) Exemplo de uma imagem sintética com 2 regiões de reflectância afetadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar de binarização calculado. (c) Imagem binarizada a partir do limiar calculado.

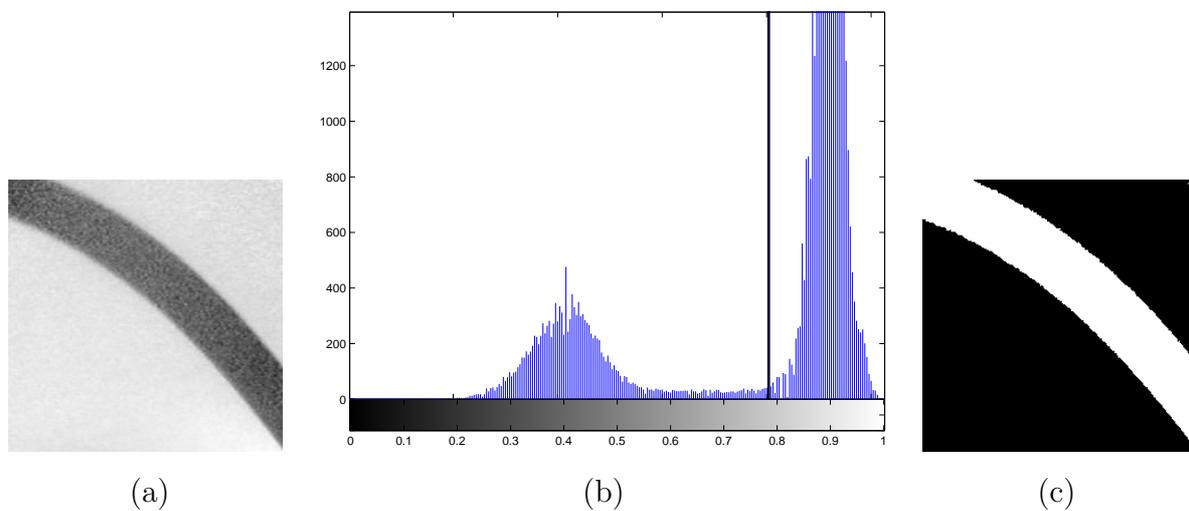


Figura 4.7: (a) Exemplo de uma imagem real de uma trajetória sobre um fundo. (b) Histograma correspondente, com destaque para o limiar calculado. (c) Imagem binarizada a partir do limiar calculado.

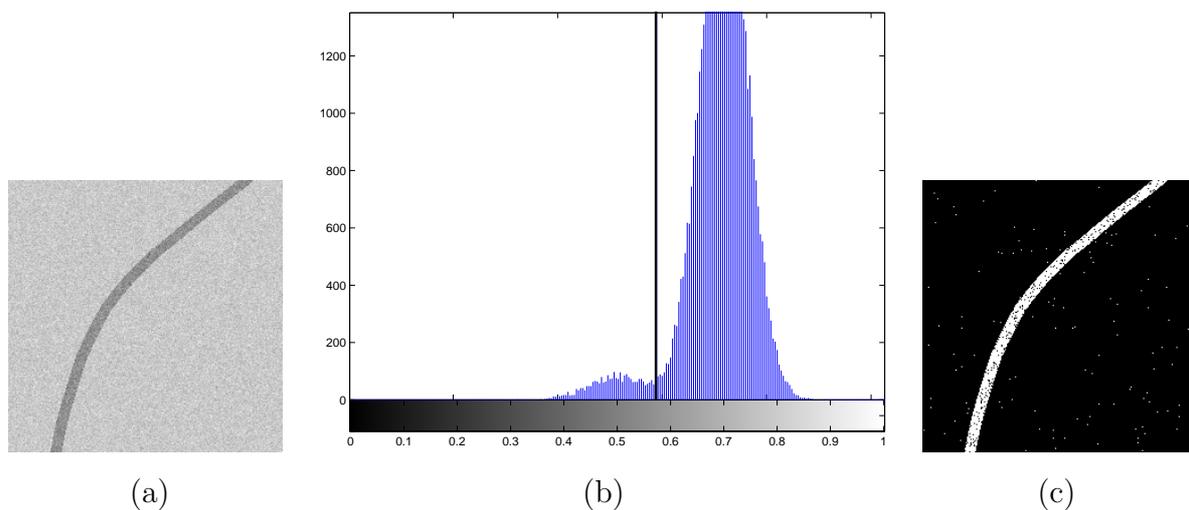


Figura 4.8: (a) Exemplo de uma imagem sintética com 2 regiões de reflectância altamente degradadas por ruído gaussiano aditivo. (b) Histograma correspondente, com destaque para o limiar teórico; observa-se a sobreposição das duas distribuições. (c) Imagem binarizada a partir do limiar teórico ótimo.

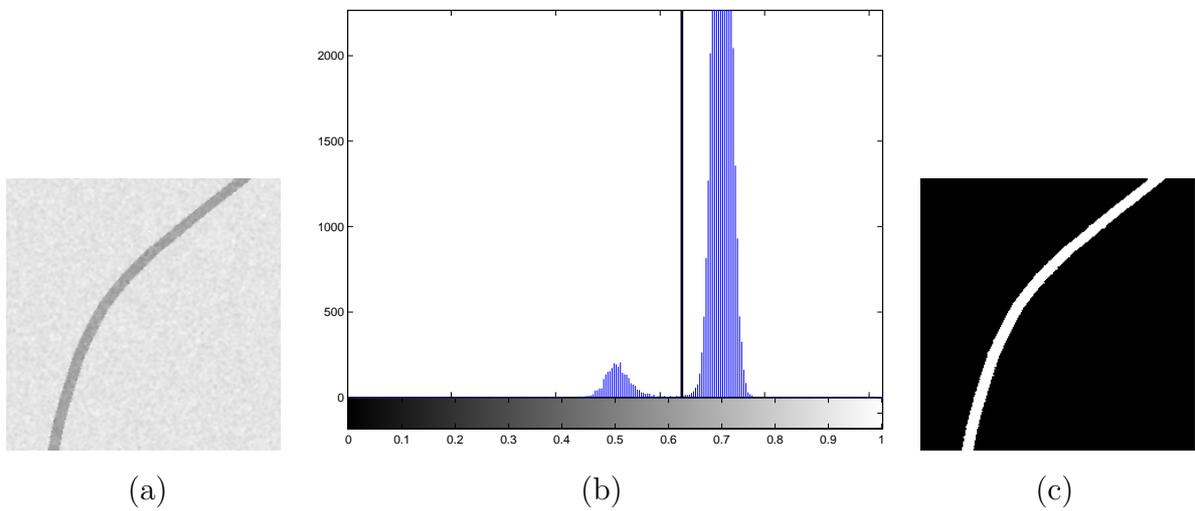


Figura 4.9: (a) Resultado da aplicação do filtro de Wiener à imagem 4.8. (b) Histograma correspondente, com destaque para o limiar calculado. (c) Imagem binarizada a partir do limiar calculado.

Capítulo 5

PRÉ-PROCESSAMENTO DAS IMAGENS DE ENTRADA

Conforme descrito no capítulo 3, a primeira etapa da extração das trajetórias a serem rastreadas pelo manipulador consiste num processo de binarização, ou seja, na classificação de cada região da cena com base em sua reflectância. Entretanto, não se dispõe a princípio de uma medida desta grandeza; a luminância em cada *pixel* de uma imagem, embora diretamente relacionada à reflectância do ponto que lhe corresponde, é ainda afetada por um fator $I_l(i, j)$ que depende da intensidade com que este ponto é iluminado, segundo a equação (3.43). Já que não foi possível, nos experimentos realizados, iluminar o campo de visão da câmera de forma uniforme, o que garantiria uma relação constante entre a luminância e a reflectância, constatou-se a necessidade de um pré-processamento das imagens, com o intuito de extrair a reflectância $R(i, j)$ a partir do produto $I_l(i, j) \cdot R(i, j)$ e, por assim dizer, compensar a não-uniformidade da iluminação.

Duas técnicas de pré-processamento específicas para este fim foram investigadas. A primeira, descrita na seção 5.2, baseia-se num método proposto por Land em 1964 [14] para sinais de domínio unidimensional, mas que pode ser estendido a sinais de domínio bidimensional, como imagens, pela substituição do operador de derivação por um operador rotacionalmente simétrico [13]. O segundo método, apresentado na seção 5.3, é de complexidade computacional bastante inferior, mas exige um controle mais rigoroso da iluminação do campo visual, a qual não deve variar com o tempo. As razões por que se adotou o segundo procedimento são então apresentadas na seção 5.4.

5.1 INTRODUÇÃO

Na seção 3.1, abordou-se o processo de formação de imagens de superfícies planas, constituídas de um número finito de regiões de reflectâncias constantes. A análise conduziu à equação (3.43), que estabelece que a luminância em um *pixel* (i, j) é igual ao produto da reflectância $R(i, j)$ da região em torno dele projetada pelo fator $I_l(i, j)$, diretamente proporcional à intensidade com que ela é iluminada. Conclui-se que, a não ser em situações

especiais em que um rígido controle das condições de iluminação permitam que toda a cena observada seja irradiada de forma uniforme, não se pode inferir a informação de reflectância almejada a partir diretamente do brilho no plano de retina.

Esta mesma constatação foi realizada experimentalmente por Land em 1959 [15]. Após distribuir nas paredes de uma sala folhas de papel em diferentes tons de cinza, Land iluminou de maneira estratégica o ambiente, de tal forma que a potência irradiada nas folhas mais escuras fosse maior do que nas mais claras. Um fotômetro foi ainda disposto em um ponto de observação pré-estabelecido, com o intuito de garantir que a potência nele irradiada a partir de cada folha fosse constante, independentemente de sua reflectância. Nestas condições, a imagem formada por uma câmera CCD apresentaria a mesma luminância em todos os *pixels*, comprovando que apenas a medição direta desta grandeza não provê a informação de reflectância procurada.

A conclusão mais importante do experimento de Land, no entanto, é o fato de que, mesmo nestas condições consideravelmente desfavoráveis, a imagem adquirida contém toda a informação necessária à extração dos valores das reflectâncias, senão de forma absoluta ao menos de cada região em relação às demais. De fato, observadores situados na posição indicada pelo fotômetro não tiveram dificuldade em distinguir as áreas mais claras das mais escuras. Assim, a imagem formada pode ser convenientemente segmentada de acordo com a reflectância em cada fragmento da superfície, mas uma etapa de processamento específica é necessária à extração desta informação a partir da distribuição de potência irradiada ao longo do plano de retina.

No contexto da binarização de imagens de trajetórias, uma imagem sintética permite ilustrar de forma bastante clara o problema apresentado por Land. Na figura ??-(a), é apresentada a imagem de uma trajetória de reflectância $\mu_t = 0,6$ sobre um fundo de reflectância $\mu_f = 0,9$. Supõem-se inicialmente condições ideais de iluminação, de tal sorte que todos os pontos das duas superfícies são irradiados com a mesma intensidade. Desta forma a luminância de cada *pixel* tem densidade de probabilidade dada pela média de duas gaussianas ponderada pelas proporções de trajetória e fundo na imagem, segundo a equação (3.51); conseqüentemente, o histograma da imagem compõe-se de duas curvas normais em torno das médias 0,6 e 0,9, como exemplifica a figura ??-(b). O processo de limiarização descrito nos capítulos 3 e 4 permite então separar as duas distribuições e classificar todos os *pixels* como pertencentes à trajetória ou ao fundo de forma automática e a partir apenas da luminância.

Um modelo mais realista leva em conta os efeitos de uma iluminação não necessariamente uniforme. Na figura ??, a mesma imagem passa a ser afetada por um padrão de iluminação que varia do mais escuro, à esquerda, ao mais claro, à direita (figura ??-(b)), simulando a utilização de uma fonte de luz que ilumina a cena da direita para a esquerda. Conforme a equação (3.43), o efeito da iluminação sobre a luminância é multiplicativo, de forma que se obtém a imagem da figura ??-(c).

Na nova imagem, observa-se que os *pixels* associados a alguns pontos da trajetória, à direita, têm maior luminância do que alguns *pixels* do fundo, à esquerda. A informação de reflectância, necessária à limiarização, foi claramente corrompida, já que a trajetória considerada é sempre mais escura do que o fundo. Como consequência, o histograma correspondente, mostrado na figura ??-(b), não corresponde mais ao modelo previsto pela equação (3.51), e o procedimento de limiarização proposto não pode ser aplicado diretamente à segmentação da imagem.

As duas técnicas apresentadas a seguir foram implementadas e avaliadas com o intuito de se verificar se, numa situação como a exemplificada, elas permitem compensar a não-uniformidade da iluminação e então classificar a imagem com uso do método de binarização descrito nos capítulos 3 e 4. Alguma informação adicional à luminância isolada de cada *pixel* se faz necessária, já que se deseja extrair um único fator, $R(i, j)$, a partir de um produto, $Il(i, j) \cdot R(i, j)$.

5.2 PRIMEIRA TÉCNICA DE COMPENSAÇÃO DA NÃO-UNIFORMIDADE DA ILUMINAÇÃO

O primeiro procedimento avaliado admite que, no segundo membro da equação (3.43), as contribuições referentes à reflectância, $R(i, j)$, e à iluminação, $Il(i, j)$, apresentam características espectrais distintas. De um lado, a iluminação é suposta variar de forma suave ao longo da imagem, ou seja, não há transições bruscas em $Il(i, j)$ e sua transformada de Fourier tem componentes concentradas nas baixas frequências. Por outro lado, considera-se que as superfícies que dão origem à imagem são formadas por regiões de reflectâncias constantes; assim, $R(i, j)$ é constante em cada segmento da imagem, o que dá origem a componentes de baixa frequência, mas varia bruscamente de um segmento para outro, apresentando portanto também componentes de alta frequência.

Desta forma, a técnica não se aplica, por exemplo, à análise de cenas em que ocorre a formação de sombras, que resultam em descontinuidades em Il . A hipótese de que $R(i, j)$ seja constante em cada segmento e apresente descontinuidade de um segmento a outro é satisfeita restringindo-se a análise às chamadas figuras de Mondrian [13], formadas por partições de superfícies em áreas contíguas de reflectâncias constantes. Felizmente, as imagens de trajetórias sobre fundos uniformes se enquadram nesta definição.

O processamento realizado se baseia num método proposto pelo próprio Land, designado *retinex*. Este método se restringe a sinais Il e R de domínio contínuo e unidimensional, mas é estendido a sinais de domínio discreto e bidimensional permitindo sua aplicação ao pré-processamento de imagens.

A idéia básica é que, como o sinal R a ser recuperado é constante por partes, sua derivada é formada por pares de impulsos presentes em cada descontinuidade, mas nula nos intervalos em que a função é constante. As figuras ??-(a) e ??-(b) exemplificam um

sinal R constante por partes e sua derivada. Já a função Il , suposta contínua em todos os pontos, não apresenta impulsos em sua derivada, conforme ilustram as figuras ??-(a) e ??-(b).

Assim, caso se possa estimar a partir do sinal de entrada $I = Il \cdot R$ uma função que seja a soma $f' + g'$ das derivadas de duas funções f e g que dependem exclusivamente de R e de Il e que preservem as características espectrais destes sinais, é possível separar desta soma as contribuições de f' e de g' , a primeira correspondendo a todos os impulsos presentes. A integração de f' conduz então à função f , que depende apenas de R e permite assim recuperar este sinal.

O cálculo direto da derivada I' de L não conduz a este resultado, já que $I' = Il' \cdot R + Il \cdot R'$, não correspondendo, portanto, à soma simples de R' e L' . Como, no entanto, o logaritmo de um produto é a soma dos logaritmos dos fatores, tem-se:

$$\log I = \log Il + \log R. \quad (5.1)$$

Felizmente, os logaritmos de Il e de R preservam as características que permitem separar com facilidade suas derivadas: o logaritmo de R é também constante por partes, de forma que sua derivada é formada por pares de impulsos nas descontinuidades e se anula nos demais pontos, e o logaritmo de Il não apresenta nenhuma descontinuidade, sendo sua derivada finita em todos os pontos.

Além disso, como a derivada da soma é a soma das derivadas das parcelas:

$$(\log_{10} I)' = (\log_{10} Il)' + (\log_{10} R)'. \quad (5.2)$$

Conseqüentemente, calculando-se o logaritmo do sinal de entrada e derivando-se o resultado, obtém-se um sinal que é a soma de duas funções que podem ser prontamente identificadas: $(\log_{10} R)'$ é o sinal formado por todos os impulsos em $(\log_{10} I)'$ e $(\log_{10} Il)'$ é a diferença entre $(\log_{10} I)'$ e $(\log_{10} R)'$. A figura ?? ilustra esta situação: em (c), é mostrado o sinal $(\log_{10} I)$, dado pela soma dos sinais em (a) — $(\log_{10} Il)$ — e em (b) — $(\log_{10} R)$. Na derivada de $(\log_{10} I)$ — figura ??-(f) — a contribuição de $(\log_{10} R)'$ — figura ??-(b) — é imediatamente identificada com todos os impulsos presentes.

Uma vez determinado $(\log_{10} R)'$, determina-se $(\log_{10} R)$, a menos de uma parcela k constante, por integração. O sinal R procurado é então obtido, a menos de uma constante multiplicativa 10^{-k} , pela aplicação da função inversa do logaritmo, ou seja, da potência de 10.

A figura 5.1 sintetiza o método *retinex* aqui descrito: inicialmente, calcula-se o logaritmo do sinal $I = Il \cdot R$ de entrada, o que conduz à função $\log_{10} Il + \log_{10} R$. Em seguida, o operador de derivação D é aplicado, gerando uma função em que a contribuição de R — $(\log_{10} R)'$ — é prontamente identificada como a soma de todos os impulsos ($\delta's$) presentes. A integral desta contribuição permite obter o logaritmo do sinal R , a menos da parcela

constante k . Finalmente, aplica-se a inversa do logaritmo ao sinal obtido, o que resulta em $R \cdot 10^{-k}$.

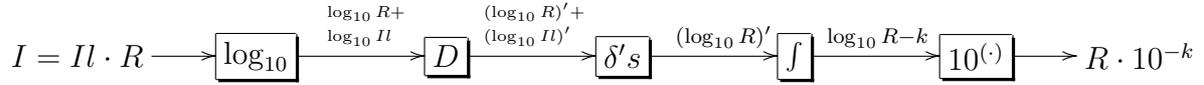


Figura 5.1: Diagrama de blocos que descreve o algoritmo *retinex*.

A aplicação do algoritmo *retinex* ao problema da compensação da não-uniformidade da iluminação exige que sejam feitas duas considerações. Em primeiro lugar, a imagem é uma função de domínio bidimensional. Assim, descontinuidades no sinal R , que permitem distingui-lo de Il ao se avaliar $\log_{10} I$, podem ocorrer tanto ao longo do eixo horizontal como do eixo vertical, ou ainda em alguma composição destas duas direções. Conseqüentemente, o operador derivada utilizado deve ser substituído por um operador que permita avaliar as variações ao longo de quaisquer direções de forma indistinta, ou seja, que seja rotacionalmente simétrico. Um exemplo de tal operador é o laplaciano [13][1], definido por:

$$\nabla^2 = D_{11} + D_{22}, \quad (5.3)$$

onde D_{ij} representa o operador derivada parcial sucessiva na i -ésima e j -ésima variáveis.

A segunda observação diz respeito ao fato de que sinais digitais têm domínio discreto. Conseqüentemente, o operador ∇^2 deve ainda ser substituído por um operador *lap* que permita calcular amostras do laplaciano do sinal de entrada a partir das amostras deste sinal, supondo-se finita sua banda passante e uma taxa de amostragem de acordo com o critério de Nyquist. Neste caso, no entanto, as descontinuidades em R não mais correspondem a impulsos em *lap*, já que elas geram componentes de frequências arbitrariamente altas que não são contempladas durante o processo de amostragem, que supõe uma banda finita. Em vez disso, obtêm-se nas descontinuidades de R valores tanto mais elevados quanto maior a taxa de amostragem, de forma que os sinais $\text{lap}(\log_{10} Il)$ e $\text{lap}(\log_{10} R)$ devem ser separados por um processo de limiarização, e não mais pela simples associação do primeiro a impulsos em $\text{lap}(\log_{10} I)$.

Por ser o laplaciano um operador linear e invariante ao deslocamento, sua implementação no domínio discreto pode ser efetuada pela convolução de sua resposta impulsional h_{lap} com o sinal de entrada. Um possível núcleo h_{lap} a ser utilizado é dado por [13]:

$$h_{lap} = \frac{1}{4 \cdot \epsilon_1 \cdot \epsilon_2} \cdot \begin{bmatrix} 1 & 4 & 1 \\ 4 & -20 & 4 \\ 1 & 4 & 1 \end{bmatrix}. \quad (5.4)$$

Após o processo de limiarização aplicado sobre o sinal $\text{lap}(\log_{10} I)$, deve então ser

realizada uma filtragem que corresponda aproximadamente a uma operação inversa do laplaciano, permitindo calcular o sinal $\log_{10} R$. O sinal obtido deve ser tal que:

$$h_{lap} * (\log_{10} R) = LI[\text{lap}(\log_{10} I)], \quad (5.5)$$

onde LI representa o sinal obtido do processo de limiarização.

Assim, para que $\log_{10} R$ possa ser estimado por um processo de filtragem linear, pela convolução de $LI[\text{lap}(\log_{10} I)]$ com um núcleo h_{inv_lap} , deve-se ter:

$$\log_{10} R = h_{inv_lap} * LI[\text{lap}(\log_{10} I)], \quad (5.6)$$

o que, substituído em (5.5), conduz a:

$$h_{lap} * \{h_{inv_lap} * LI[\text{lap}(\log_{10} I)]\} = LI[\text{lap}(\log_{10} I)]. \quad (5.7)$$

Por ser a convolução uma operação comutativa [24], segue:

$$[h_{lap} * h_{inv_lap}] * LI[\text{lap}(\log_{10} I)] = LI[\text{lap}(\log_{10} I)]. \quad (5.8)$$

Mas a função que convoluída com qualquer sinal resulta no próprio sinal é o impulso unitário δ . Portanto:

$$h_{inv_lap} * h_{lap} = \delta. \quad (5.9)$$

Conclui-se que o núcleo a ser utilizado deve ser tal que, convoluído com o núcleo em (5.4), resulte no impulso δ . Uma possível abordagem consiste em adotar um suporte finito para h_{inv_lap} , sendo então calculados, pelo método dos mínimos erros quadrados, os elementos constituintes deste núcleo que melhor satisfazem a equação (5.9). O que se observa, no entanto, é que a adoção de um suporte finito para h_{inv_lap} (filtro FIR) não permite satisfazer com exatidão esta equação, mas apenas de forma aproximada, segundo o critério adotado. Isto já era esperado, já que o laplaciano não é, na verdade, um operador inversível (não é possível, por exemplo, recuperar o nível DC de uma imagem a partir de seu laplaciano, o mesmo problema apresentado pela derivada de sinais de domínio unidimensional). Esta característica será ainda discutida na seção 5.4.

A aplicação deste método para um núcleo 3x3 conduz a:

$$h_{inv_lap} = \begin{bmatrix} -0,0074 & -0,0171 & -0,0112 \\ -0,0152 & -0,0639 & -0,0192 \\ -0,0085 & -0,0182 & -0,0116 \end{bmatrix}, \quad (5.10)$$

que convoluído com h_{lap} resulta em:

$$h_{lap} * h_{inv_lap} = \begin{bmatrix} -0.0074 & -0.0468 & -0.0871 & -0.0618 & -0.0112 \\ -0.0449 & -0.0449 & -0.0215 & 0.0141 & -0.0638 \\ -0.0767 & -0.0506 & 0.9605 & 0.0014 & -0.0994 \\ -0.0492 & -0.0273 & -0.0071 & 0.0189 & -0.0656 \\ -0.0085 & -0.0522 & -0.0927 & -0.0646 & -0.0116 \end{bmatrix}. \quad (5.11)$$

As alterações aqui apresentadas, que permitem adequar o método *retinex* ao pré-processamento de imagens digitais, conduzem ao algoritmo esquematizado na figura 5.2. Inicialmente, é calculado o logaritmo da imagem de entrada, *pixel a pixel*. Isto conduz à soma dos logaritmos da reflectância (R) e do fator devido à iluminação (Il) em cada *pixel*. Em seguida, a convolução com o núcleo h_{lap} conduz ao sinal $h_{lap} * \log_{10} Il + h_{lap} * \log_{10} R$, em que a contribuição devida à reflectância pode ser isolada por um processo de limiarização. O novo sinal é então convoluído com h_{inv_lap} , permitindo calcular de forma aproximada $\log_{10} R$, a menos de uma parcela constante k . Finalmente, o cálculo da potência de 10 do novo sinal, *pixel a pixel*, resulta na estimativa R das reflectâncias, a menos da constante multiplicativa 10^{-k} .

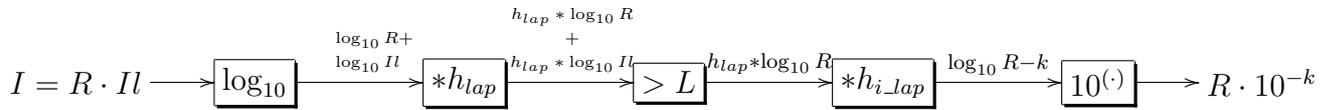


Figura 5.2: Diagrama de blocos que descreve a primeira técnica avaliada de compensação da não-uniformidade da iluminação durante a formação das imagens de trajetórias.

5.3 SEGUNDA TÉCNICA DE COMPENSAÇÃO DA NÃO-UNIFORMIDADE DA ILUMINAÇÃO

A segunda técnica de compensação da não-uniformidade da iluminação tem por base a hipótese de que a influência da potência radiada pelas fontes de luz não se altera com o tempo. Em outras palavras, admite-se que há um controle da iluminação do ambiente em que se dá a formação das imagens analisadas, de forma que, para cada *pixel* (i, j), o fator $Il(i, j)$ é o mesmo durante todo o processo de rastreamento.

Nesta situação, caso se possa determinar, em um procedimento empírico prévio, o valor de $Il(i, j)$ para cada posição (i, j), a reflectância do ponto cuja imagem nela se forma pode ser determinada dividindo-se sua luminância, $I(i, j)$, por aquele valor, segundo a equação (3.43).

Um experimento simples permite determinar $Il(i, j)$ a menos de uma constante multiplicativa. Submete-se uma superfície plana lisa, de reflectância constante, às mesmas

condições de iluminação em que ocorrerá posteriormente o rastreamento de trajetórias. Adquire-se então uma imagem desta superfície. Segundo a equação (3.43), a imagem obtida é dada por:

$$P(i, j) = k \cdot Il(i, j), \quad (5.12)$$

onde k é a reflectância da superfície, igual para todos os pontos considerados.

Esta imagem é adotada então como um *padrão de iluminação*. Os efeitos da não-uniformidade da iluminação, durante a formação da imagem de uma trajetória a ser rastreada, são compensados por meio de sua divisão pelo padrão, sendo obtida a nova imagem

$$C(i, j) = \frac{I(i, j)}{P(i, j)} = \frac{Il(i, j) \cdot R(i, j)}{k \cdot Il(i, j)} = \frac{1}{k} \cdot R(i, j). \quad (5.13)$$

Assim, esta técnica também permite determinar a reflectância $R(i, j)$ a menos de uma constante multiplicativa.

5.4 ESCOLHA DO MÉTODO DE PRÉ-PROCESSAMENTO DAS IMAGENS UTILIZADO

O método de compensação da não-uniformidade da iluminação baseado no algoritmo *retinex* apresenta uma clara vantagem em relação ao segundo método apresentado. Por não exigir que as condições de iluminação se mantenham constantes, é aplicável em sistemas em que se dispõe de um controle bem menos rígido destas condições, desde que se evite a formação de sombras. Ainda assim, várias desvantagens da primeira técnica levaram à adoção, no sistema de rastreamento implementado, do segundo procedimento.

Por um lado, observa-se, na figura 5.2, a necessidade de se realizar um processo de limiarização na imagem resultante da aplicação do filtro h_{lap} , para que se efetue a separação das componentes $h_{lap} * \log_{10} R$ e $h_{lap} * \log_{10} Il$. Desta forma, permanece o problema inicial de se determinar um limiar apropriado. Como o pré-processamento das imagens tem por objetivo permitir o posterior cálculo automático do limiar de binarização, deve-se utilizar um procedimento de compensação da iluminação não-uniforme que não exija também uma etapa de limiarização, como é o caso da segunda técnica.

Outro problema encontrado durante a implementação da primeira técnica foi o fato de que ela utiliza um filtro linear (h_{inv_lap}) com o objetivo de estimar em situações especiais uma imagem a partir do sinal que dela resultaria após a aplicação do laplaciano. Em outras palavras, este filtro deve realizar de forma aproximada, e a menos de um nível DC, o processo inverso de um operador que, a rigor, é não-inversível. Considerando que, no caso de imagens de trajetórias, o sinal obtido após a aplicação do laplaciano e da limiarização só deve apresentar valores significativamente diferentes de zero nos *pixels* de

transição entre fundo e trajetória, o filtro em questão deve obrigatoriamente apresentar um núcleo de dimensões superiores à máxima largura da trajetória na imagem (se isto não ocorre, o filtro não permite discriminar os *pixels* mais internos da trajetória daqueles do fundo, já que, nos dois casos, atua apenas em sinais de mesma magnitude). Este fato se reflete na constatação de que a convolução do núcleo h_{lap} — equação (5.4) — com um filtro de suporte finito — por exemplo, equação (5.10) — nunca é exatamente igual ao impulso de domínio bidimensional, mas se aproxima cada vez mais desta situação ideal à medida que se aumenta o suporte de h_{inv_lap} . Conseqüentemente, apenas um filtro IIR permitiria diferenciar as reflectâncias do fundo e da trajetória independentemente de suas dimensões na imagem, mas a necessidade de se utilizarem grandes suportes mesmo numa aproximação já resulta em grande complexidade computacional, além de resultar em grandes perdas das bordas das imagens por não se dispor de informação completa de sua vizinhança durante o processo de filtragem.

Mesmo deconsiderando-se toda a complexidade computacional inerente à aplicação do filtro h_{inv_lap} , o método baseado no algoritmo *retinex* continua a exigir um esforço bastante superior. Enquanto a primeira técnica envolve uma série de processos de filtragem não-lineares (aplicação do logaritmo, da potência de 10, limiarização) e lineares (convolução com os núcleos h_{lap} e h_{inv_lap}), a segunda exige apenas a divisão da imagem de entrada por um padrão pré-armazenado. Com efeito, $M \cdot N$ divisões se fazem necessárias durante a compensação da iluminação não-uniforme pelo segundo método, enquanto que, no primeiro, o dobro de operações é realizado somente durante a aplicação do logaritmo e da potência de 10.

Por estas razões, a segunda técnica foi a adotada no sistema de rastreamento implementado. Para garantir que as condições de iluminação se mantivessem constantes, o *plotter* foi colocado no interior de uma caixa, sob cuja tampa foram posicionadas as trajetórias a serem rastreadas. Isto permitiu reduzir a influência da iluminação externa. Além disso, a lâmpada utilizada para iluminar a cena foi fixada à câmera, para que as condições de iluminação permanecessem independentes da posição do atuador.

Capítulo 6

EXTRAÇÃO DA TRAJETÓRIA A PARTIR DA IMAGEM BINARIZADA

Os algoritmos de visão computacional, de forma genérica, podem ser divididos em duas etapas principais: a análise de imagens e a análise de cenas [13]. Na primeira etapa, objetiva-se extrair das imagens de entrada apenas as informações que são úteis, de forma direta, ao processo em execução. Informações relacionadas a objetos que não dizem respeito ao processo, a ruídos aditivos ou a diferenças de luminância no fundo da cena, bem como quaisquer informações consideradas redundantes, não devem estar presentes na imagem de saída, já que levariam a esforço computacional desnecessário. Já que, nesta etapa, tanto a entrada como a saída são imagens, ela se resume a uma ou mais técnicas de processamento de imagens.

A segunda etapa, por sua vez, tem por objetivo construir uma representação simbólica adequada da cena. Trata-se de traduzir a informação extraída na primeira etapa numa forma adequada à realização, em tempo hábil, da tarefa de interesse. Assim, a principal diferença entre a análise de imagens e a análise de cenas é que, enquanto que na primeira tanto a entrada como a saída são imagens, na segunda a saída é uma descrição simbólica simples, altamente específica e determinada pela natureza da aplicação visada.

Neste sentido, todas as técnicas apresentadas nos capítulos 3 a ?? compõem tão-somente a primeira etapa do algoritmo de visão computacional utilizado pelo sistema aqui pesquisado. Da imagem de entrada, obtém-se uma imagem binária, em que cada *pixel* apresenta a luminância 0 ou 1 conforme corresponda ao fundo da cena ou à trajetória, respectivamente. Embora esta informação seja suficiente ao seguimento da trajetória, e esteja livre de dados desnecessários tais como as variações de luminância causadas por gradientes de iluminação, ela não explicita algumas relações essenciais ao problema.

A primeira destas relações é a ordem em que devem ser percorridos os diversos pontos da trajetória. A imagem binária pode ser analisada como um conjunto de pontos, especificados pela luminância 1, porém sem nenhuma relação de ordem estabelecida. Embora um observador humano seja capaz de determiná-la de forma quase instantânea a partir da

visualização da imagem binarizada, desde que se arbitre um sentido de percurso, não se trata de informação evidente. Um algoritmo específico deve ser empregado para torná-la explícita, antes que se proceda ao comando do manipulador robótico.

A segunda relação diz respeito à definição de instantes de tempo em que cada ponto extraído deverá ser percorrido pelo manipulador. Mesmo em tarefas em que não é previamente definido o instante em que o manipulador ocupa cada posição na trajetória, bastando que toda ela seja percorrida num sentido específico, é necessário, normalmente, garantir uma velocidade constante em módulo ao longo de todo o trajeto. Um exemplo são os robôs de soldagem; se nenhuma restrição é feita acerca do tempo de percurso da trajetória, nos segmentos percorridos a uma velocidade maior, menos solda é depositada pelo manipulador. É necessário garantir uma velocidade uniforme ao longo de toda a trajetória para que a quantidade de solda depositada seja a mesma em todos os pontos.

Neste sentido, mesmo estabelecida uma seqüência de pontos que compõem a trajetória, o controle propriamente dito do manipulador não está pronto para ser realizado. Enviar o manipulador a cada ponto da seqüência, a intervalos regulares, não garante uma velocidade uniforme, já que a distância entre pontos consecutivos extraídos da imagem binarizada não será, em geral, constante.

Este capítulo trata do algoritmo utilizado para resolver o primeiro problema levantado. O algoritmo de busca circular, proposto por Bauchspiess em 1995 [3], permite extrair da imagem binarizada os pontos a serem percorridos pelo manipulador, em seqüência. Uma descrição detalhada do processo é apresentada na seção 6.1. Considerações de ordem prática acerca da implementação do algoritmo são então levantadas na seção 6.2.

A solução ao segundo problema, por sua vez, exige o cálculo de pontos igualmente espaçados ao longo da trajetória, de forma a garantir uma velocidade uniforme do manipulador ao ser enviado à nova seqüência de posições. Em outras palavras, deve-se gerar uma seqüência de pontos da trajetória de tal forma que a distância entre dois pontos consecutivos seja sempre a mesma. No entanto, já que a distância a ser considerada é medida no espaço de trabalho do atuador, e não em número de *pixels* na imagem da trajetória, o primeiro passo é converter as coordenadas dos *pixels* extraídos em posições, em milímetros, do atuador. A técnica empregada para resolver este problema é descrita, portanto, no final do capítulo 7, que trata especificamente desta conversão.

Os dois procedimentos mencionados compõem a análise de cenas, no algoritmo de visão computacional empregado. Ao contrário das técnicas de processamento de imagens até então apresentadas, com aplicações em diferentes áreas, eles são bastante específicos ao problema de rastreamento de trajetórias, o que é característico desta etapa dos algoritmos de visão.

6.1 O ALGORITMO DE BUSCA CIRCULAR

No algoritmo de busca circular, a determinação de cada ponto da seqüência a ser seguida pelo manipulador é feita a partir do ponto imediatamente anterior e da reta tangente à trajetória neste ponto. A tangente no novo ponto pode ser então aproximada pela reta que o une ao anterior, desde que eles sejam suficientemente próximos. Desta forma, obtêm-se os elementos necessários à determinação do terceiro ponto da seqüência, e assim por diante: pela aplicação repetidas vezes do mesmo tipo de busca, é extraído um número qualquer de pontos, na seqüência em que devem ser percorridos pelo manipulador.

A figura 6.1 ilustra como opera a busca circular na determinação do próximo ponto da trajetória. O ponto A destacado exemplifica um ponto já extraído pelo algoritmo numa etapa anterior, ou ainda o primeiro ponto da seqüência, fornecido pelo usuário. O próximo ponto, B , é obtido por varredura de uma circunferência centrada em A , até que seja encontrada a intersecção com a trajetória. É imprescindível que a varredura se inicie a partir de um ponto apropriado, e que seja feita num sentido também apropriado, para que seja determinada a intersecção correta (ponto B , e não C). Para tanto, é necessário levar em conta a direção de percurso da trajetória, representada pela reta tangente no ponto A . Considera-se a semi-reta com origem em A que define um pequeno ângulo α em relação à tangente; a varredura se dá então em sentido anti-horário, a partir do ponto em que a semi-reta intercepta a circunferência.

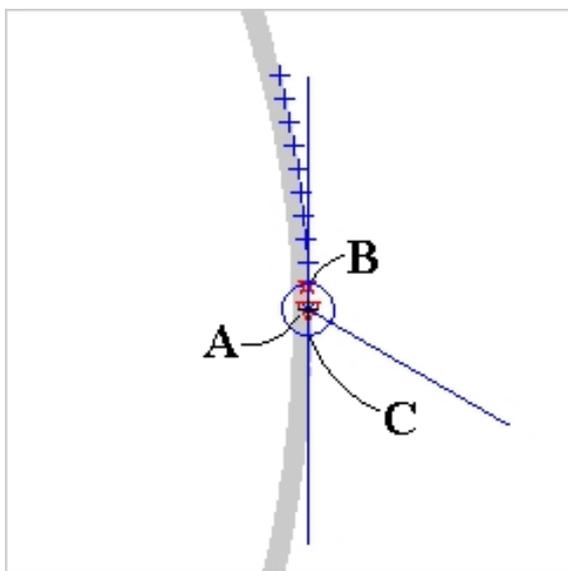


Figura 6.1: Descrição do algoritmo de busca circular, com a direção de varredura das circunferências em sentido anti-horário.

Observe-se que a adoção do sentido anti-horário para a varredura é até certo ponto arbitrária. Resultado semelhante é obtido se a semi-reta for traçada a partir de um mesmo ângulo α em relação à tangente medido no sentido horário, sendo a varredura ao longo

da circunferência realizada também em sentido horário. Neste caso, porém, a trajetória é percorrida pelo outro lado da linha.

Note-se que dois parâmetros devem ser determinados previamente à aplicação do algoritmo de busca circular: o raio R da circunferência centrada em A e o ângulo α . O valor de R determina a distância entre os pontos extraídos da trajetória; quanto maior R , maior a distância entre eles. Assim, como é realizada uma interpolação dos pontos extraídos na busca circular, conforme abordado na seção 7.3, adotou-se um raio R de 10 *pixels* no início da busca, levando a uma distância considerável entre pontos consecutivos. Cabe observar, no entanto, que como a trajetória pode apresentar falhas, caso haja classificações incorretas durante o processo de binarização, eventualmente a circunferência poderá não interceptá-la em ponto algum; nesta situação, o algoritmo prossegue aumentando-se o raio de busca, até que seja encontrada a primeira intersecção.

Quanto à determinação do valor apropriado de α , ela deve ser realizada tendo em vista uma solução de compromisso. Por um lado, a adoção de um ângulo muito pequeno aumenta o tempo gasto, em média, até que seja extraído cada ponto da trajetória, já que a varredura é feita num arco de circunferência de comprimento maior. Assim, um tempo de busca menor é obtido aumentando-se o valor de α . Por outro lado, um valor alto de α limita a curvatura máxima das trajetórias que podem ser extraídas a partir da busca circular. Empiricamente, determinou-se um valor de $\frac{\pi}{6}$ para o ângulo α , o que se mostrou suficiente para a extração das trajetórias com as curvaturas analisadas nesta pesquisa.

A determinação do número apropriado de *pixels* a serem extraídos durante a busca circular, por sua vez, é feita no capítulo 7.

6.2 CONSIDERAÇÕES SOBRE DETALHES DE IMPLEMENTAÇÃO DO ALGORITMO DE BUSCA CIRCULAR

A programação do algoritmo de busca descrito conduziu a alguns problemas de ordem prática, que tiveram de ser contornados para que se obtivesse uma implementação suficientemente eficiente do ponto de vista do tempo de execução.

O primeiro destes problemas corresponde à definição da área de varredura, durante a busca de cada ponto da trajetória. Como se adotou uma área de busca formada por *pixels* dispostos em uma circunferência, como ilustra a figura 6.1, a primeira alternativa considerada foi o cálculo dos *pixels* equidistantes do ponto já extraído da trajetória, durante a busca de cada ponto seguinte. No entanto, este cálculo, que envolve a geração de uma nova circunferência a cada etapa de busca, exige grande esforço computacional, e o tempo de execução correspondente é proibitivamente elevado. A solução encontrada foi a geração de imagens de circunferência de diversos raios e centradas no ponto $(0, 0)$, sendo armazenadas as coordenadas dos *pixels* correspondentes em um banco de dados disponível em tempo de execução ao algoritmo de busca. Desta forma, as coordenadas dos *pixels*

em que é feita a varredura são obtidas somando-se as coordenadas do centro da região de busca às da circunferência de raio correspondente armazenada.

O segundo problema diz respeito ao cálculo da interseção I entre a circunferência e a semi-reta que define um ângulo α em relação à tangente à trajetória no centro (i, j) da região de busca (figura ??). Neste caso, a técnica adotada foi o cálculo do ângulo $\theta(i_c, j_c)$ entre a linha horizontal na imagem e a semi-reta que parte de (i, j) a cada ponto (i_c, j_c) na circunferência. Conforme mostra a figura ??, os ângulos α , δ e $\theta(I)$ são os ângulos internos de um triângulo, onde δ é a inclinação da reta tangente e $\theta(I)$ é o valor de $\theta(i_c, j_c)$ na intersecção procurada. Como a soma dos ângulos internos de um triângulo é π , esta intersecção é obtida como sendo o ponto da circunferência para o qual o valor de θ calculado mais se aproxima de $\pi - (\alpha + \delta)$.

Outro problema de ordem prática encontrado foi o do estabelecimento de um sentido apropriado de varredura das circunferências. Conforme mencionado na seção anterior, o sentido pode ser tanto o horário quanto o anti-horário, a depender do sentido em que se define o ângulo α em relação à tangente à trajetória. A solução adotada para garantir uma direção correta de varredura foi a geração e o armazenamento das coordenadas das circunferências já em sentido anti-horário. Assim, a busca começa a partir da intersecção obtida entre a circunferência e a semi-reta com ângulo α em relação à tangente e se dá em direção ao último ponto armazenado. Se necessário, ou seja, caso não se encontre a intersecção com a trajetória, a busca deve reiniciar então a partir do primeiro ponto da circunferência armazenado.

Capítulo 7

MÓDULO DE CONTROLE DA POSIÇÃO DO TCP

Neste capítulo, é abordado o cálculo das referências de entrada x e y do sistema de posicionamento do TCP. Inicialmente, na seção 7.1, avalia-se a relação entre um deslocamento arbitrário do TCP, em relação aos dois eixos, e a variação na imagem das coordenadas do *pixel* associado a um ponto fixo na trajetória. Obtém-se uma relação linear entre aquelas grandezas, e um procedimento experimental, descrito na seção 7.1.2, leva à determinação da matriz de calibração que descreve a transformação.

Na seção 7.2, procede-se ao cálculo das posições desejadas do TCP associadas a cada *pixel* extraído na busca circular. Mostra-se que estas posições podem ser obtidas convertendo-se as diferenças entre as coordenadas daqueles *pixels* e um *pixel* de referência em deslocamentos necessários do TCP, que em seguida se somam à sua posição atual.

Em seguida, a seção 7.3 trata da obtenção das referências de entrada propriamente ditas. Um processo de interpolação garante a obtenção de posições de referência uniformemente espaçadas, garantindo uma velocidade uniforme do TCP ao ser enviado a estas posições a intervalos regulares de tempo.

Finalmente, considera-se, na seção 7.4, o problema da determinação do número de posições de referência a serem calculadas de cada imagem de entrada e, conseqüentemente, do número de *pixels* a serem extraídos na busca circular, o que determina a relação entre a taxa efetiva de aquisição de imagens e a taxa de atualização das referências de entrada do sistema de posicionamento.

7.1 DETERMINAÇÃO DA RELAÇÃO ENTRE O DESLOCAMENTO DO ATUADOR E A VARIAÇÃO DAS COORDENADAS DA IMAGEM DE UM PONTO FIXO NA TRAJETÓRIA

Nesta seção, é abordada a relação entre a posição da câmera e as coordenadas (i, j) do *pixel* em que se forma a imagem de um ponto (x, y) no plano que contém a trajetória a ser

rastreada. Em especial, demonstra-se que há uma relação linear entre os deslocamentos Δx e Δy do atuador ao longo dos eixos x e y , respectivamente, e as variações Δi e Δj do *pixel* correspondente a um ponto fixo naquele plano. Esta relação, expressa por uma matriz $\mathbf{A}_{2,2}$ a ser determinada experimentalmente, será necessária ao cálculo, pelo módulo de visão computacional, das posições de referência R do atuador a partir das distâncias, em número de *pixels*, dos pontos da trajetória ao ponto fixo de referência (i_{ref}, j_{ref}) . Em outras palavras, conhecida a matriz \mathbf{A} , pode-se calcular os deslocamentos necessários do atuador, em número de *bits*, a partir da distância em *pixels* de cada ponto a ser rastreado ao ponto alvo na imagem.

Inicialmente, é estabelecida, na seção 7.1.1, a expressão matemática que define a matriz \mathbf{A} , em função de parâmetros do sistema óptico tais como a distância do plano de retina ao centro óptico, a orientação dos eixos i e j do plano de imagem em relação ao sistema de coordenadas do atuador e outros. Apesar de comprovar que as grandezas Δi e Δj dependem exclusivamente de Δx e Δy , segundo uma relação linear, esta análise não é suficiente ao cálculo desta relação, porquanto não são conhecidos de antemão os valores precisos daqueles parâmetros. Assim, é proposto, na seção 7.1.2, um método experimental para o cálculo da matriz \mathbf{A} , utilizado ao longo desta pesquisa.

7.1.1 Expressão matemática para a matriz de calibração em função de parâmetros do sistema óptico

Considere-se inicialmente o problema de se relacionar a posição (x, y) de um ponto no plano da trajetória às coordenadas (i, j) de sua imagem, suposta fixa a posição do atuador no espaço de trabalho. Em sistemas de aquisição de imagens genéricos, em que os objetos analisados apresentam três graus de liberdade e o sistema de coordenadas externo à câmera pode se situar numa posição e orientação arbitrárias em relação ao eixo óptico, a relação geométrica em questão é comumente avaliada em termos de uma matriz de transformação perspectiva $\tilde{\mathbf{P}}_{3,4}$ [18][6][9]:

$$\begin{bmatrix} U \\ V \\ S \end{bmatrix} = \tilde{\mathbf{P}}_{3,4} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \quad (7.1)$$

onde x, y, z são as coordenadas de um ponto num sistema externo arbitrário e U, V, S definem a posição de sua imagem (i, j) em coordenadas homogêneas, ou seja:

$$j = \frac{U}{S} \text{ e } i = \frac{V}{S}. \quad (7.2)$$

A determinação da matriz $\tilde{\mathbf{P}}_{3,4}$ requer um procedimento experimental específico, denominado *calibração*[9][18], que constitui um dos problemas centrais em muitos sistemas

de visão computacional. A hipótese de que os objetos analisados apresentam três graus de liberdade torna necessária a construção de um padrão de pontos de referência não-coplanares, de posições bem conhecidas. Em [9], são apresentadas as características necessárias ao padrão de calibração, bem como um método experimental de determinação da matriz $\tilde{\mathbf{P}}_{3,4}$.

No sistema de rastreamento de trajetórias aqui analisado, no entanto, o problema de calibração pode ser bastante simplificado, se levado em conta o fato de que todos os pontos da trajetória se situam em um plano paralelo ao de imagem, ou seja, apresentam apenas dois graus de liberdade. A figura 7.1 descreve esta situação, com d a distância, fixa, entre o plano da trajetória \mathcal{T} e o centro óptico C .

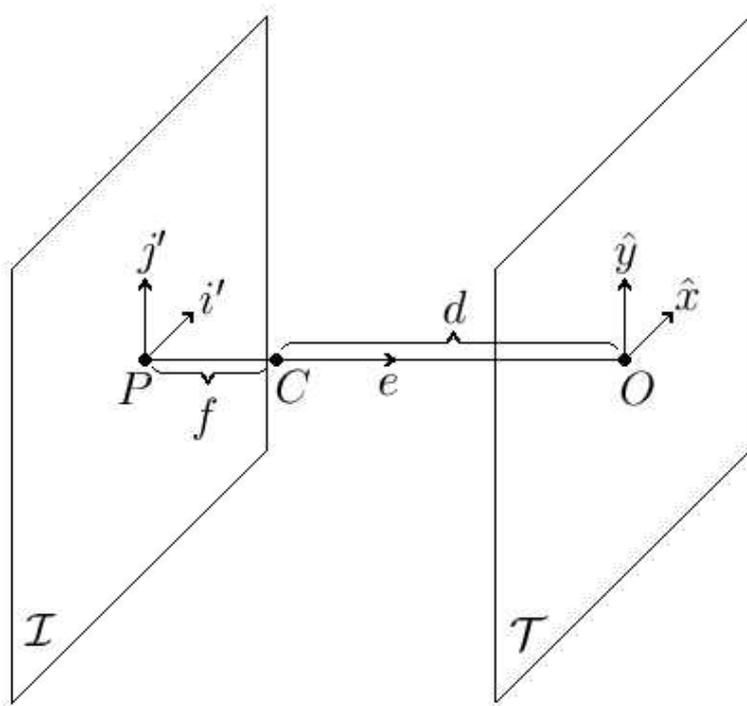


Figura 7.1: Sistema óptico utilizado durante a formação de imagens de pontos em um plano \mathcal{T} paralelo ao plano de retina \mathcal{I} .

Nesta situação, considere-se em \mathcal{T} o sistema de coordenadas $\hat{x}\hat{y}$ definido pelos eixos \hat{x} e \hat{y} paralelos, respectivamente, a i e j e com origem O na interseção entre o eixo óptico e e \mathcal{T} . Conforme destacado na figura 7.2, o fato de serem paralelos os eixos j' e \hat{y} , bem como o alinhamento entre C , O e o ponto principal do plano de imagem (P), garantem que a relação entre a coordenada \hat{y} de um ponto A em \mathcal{T} e a coordenada j' do *pixel* A' correspondente é de proporcionalidade direta:

$$-\frac{j'}{f} = \frac{\hat{y}}{d}, \quad (7.3)$$

o que se depreende da avaliação da tangente do ângulo α a partir dos triângulos CPA' e COA .

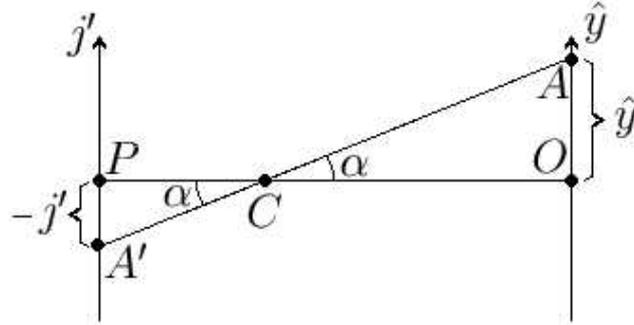


Figura 7.2: Visão unidimensional dos eixos j' e \hat{y} do sistema óptico da figura 7.1, com destaque para a transformação perspectiva durante a formação da imagem A' de um ponto A .

Analogamente, obtém-se para a relação entre i' e \hat{x} :

$$-\frac{i'}{f} = \frac{\hat{x}}{d}. \quad (7.4)$$

A relação entre as coordenadas de um ponto em \mathcal{T} e sua imagem em \mathcal{I} se torna mais complexa devido ao fato de que, devido às limitações do aparato experimental utilizado, não se garante o alinhamento entre os eixos i e j da câmera e os eixos x' e y' do espaço de trabalho do atuador, com a precisão que se faz necessária. A figura 7.3-(a) apresenta a relação mais genérica possível entre o sistema hipotético $\hat{x}\hat{y}$ e o sistema real $x'y'$, transladado e rotacionado em relação ao primeiro. Para facilitar a análise seguinte, considera-se ainda um sistema x_1y_1 , com os eixos paralelos a \hat{x} e \hat{y} porém com origem comum a $x'y'$, conforme a figura 7.3-(b).

A avaliação do seno de $\frac{\pi}{2} - \alpha$ no triângulo ABC , com α o ângulo de rotação do sistema $x'y'$ em relação a x_1y_1 , conduz a:

$$\text{sen} \left(\frac{\pi}{2} - \alpha \right) = \frac{y'}{y_1 - \lambda}. \quad (7.5)$$

Mas: $\lambda = x_1 \cdot \text{tg}\alpha$ e $\text{sen} \left(\frac{\pi}{2} - \alpha \right) = \cos \alpha$. Logo:

$$\cos \alpha = \frac{y'}{y_1 - x_1 \cdot \text{tg}\alpha}; \quad (7.6)$$

$$y' = y_1 \cos \alpha - x_1 \text{sen}\alpha. \quad (7.7)$$

Da avaliação do seno de $\frac{\pi}{2} - \alpha$ a partir do triângulo ADE , por sua vez, obtém-se:

$$\text{sen} \left(\frac{\pi}{2} - \alpha \right) = \frac{x_1}{x' - \gamma}. \quad (7.8)$$

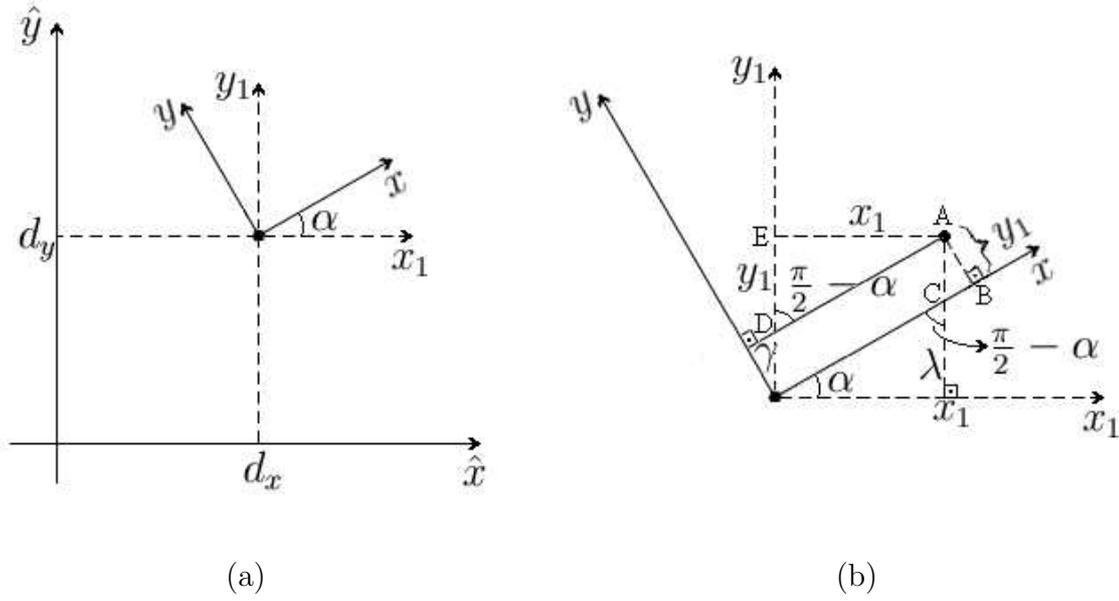


Figura 7.3: Relação genérica entre a orientação e a posição de dois sistemas de coordenadas no plano da trajetória, um arbitrário $(x'y')$ e o outro com os eixos paralelos aos do plano de retina $(\hat{x}\hat{y})$.

Mas: $\gamma = y'tg\alpha$. Logo:

$$\cos \alpha = \frac{x_1}{x' - y'tg\alpha}; \quad (7.9)$$

$$x_1 = x' \cos \alpha - y' \text{sen} \alpha. \quad (7.10)$$

Substituindo a equação (7.7) em (7.10), obtém-se:

$$x_1 = x' \cos \alpha - y_1 \text{sen} \alpha \cos \alpha + x_1 \text{sen}^2 \alpha; \quad (7.11)$$

$$x' \cos \alpha = (1 - \text{sen}^2 \alpha)x_1 + \text{sen} \alpha \cos \alpha y_1; \quad (7.12)$$

$$x' = x_1 \cos \alpha + y_1 \text{sen} \alpha. \quad (7.13)$$

As equações (7.7) e (7.13) podem ser rescritas, na forma matricial, como:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & \text{sen} \alpha \\ -\text{sen} \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}. \quad (7.14)$$

Como $x_1 = \hat{x} - d_x$ e $y_1 = \hat{y} - d_y$, segue:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \alpha & \text{sen} \alpha \\ -\text{sen} \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} - \begin{bmatrix} d_x \cos \alpha + d_y \text{sen} \alpha \\ -d_x \text{sen} \alpha + d_y \cos \alpha \end{bmatrix}. \quad (7.15)$$

Substituindo as equações (7.3) e (7.4) em (7.15), tem-se:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \left(-\frac{d}{f}\right) \begin{bmatrix} \cos \alpha & \text{sen} \alpha \\ -\text{sen} \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} i' \\ j' \end{bmatrix} - \begin{bmatrix} d_x \cos \alpha + d_y \text{sen} \alpha \\ -d_x \text{sen} \alpha + d_y \cos \alpha \end{bmatrix}. \quad (7.16)$$

A equação (7.16) relaciona a posição de um ponto A , expressa em termos das distâncias na horizontal (x') e na vertical (y') à origem O , à posição de sua imagem A' , expressa em distâncias na horizontal (i') e na vertical (j') ao ponto principal P do plano de retina. Interessa, no entanto, estabelecer esta mesma relação com a posição de A expressa em termos das palavras de 8 bits x e y com que o circuito da figura A.4 comanda o atuador, e com a posição de A' em termos dos índices adimensionais i e j que identificam o *pixel* correspondente. Como a relação entre x, y e x', y' , bem como a relação entre i, j e i', j' , é de primeiro grau (um aumento em x', y', i' ou j' implica um aumento na mesma proporção em x, y, i ou j , respectivamente), tem-se:

$$\begin{cases} x' = x \cdot k_x + x_0 \\ y' = y \cdot k_y + y_0 \end{cases} \text{ e} \quad (7.17)$$

$$\begin{cases} i' = i \cdot k_i + i_0 \\ j' = j \cdot k_j + j_0 \end{cases}, \quad (7.18)$$

com $k_x, k_y, x_0, y_0, k_i, k_j, i_0$ e j_0 constantes reais.

Da equação (7.16) obtém-se portanto:

$$\begin{bmatrix} xk_x + x_0 \\ yk_y + y_0 \end{bmatrix} = \left(-\frac{d}{f}\right) \begin{bmatrix} \cos \alpha & \text{sen} \alpha \\ -\text{sen} \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} ik_i + i_0 \\ jk_j + j_0 \end{bmatrix} - \begin{bmatrix} d_x \cos \alpha + d_y \text{sen} \alpha \\ -d_x \text{sen} \alpha + d_y \cos \alpha \end{bmatrix}, \quad (7.19)$$

o que se reduz a:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -\frac{d}{f} \cdot \frac{k_i}{k_x} \cdot \cos \alpha & -\frac{d}{f} \cdot \frac{k_j}{k_x} \cdot \text{sen} \alpha \\ \frac{d}{f} \cdot \frac{k_i}{k_y} \cdot \text{sen} \alpha & -\frac{d}{f} \cdot \frac{k_j}{k_y} \cdot \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} i \\ j \end{bmatrix} + \begin{bmatrix} \frac{d}{fk_x} (-i_0 \cos \alpha - j_0 \text{sen} \alpha) - \frac{1}{k_x} d_x \cos \alpha - \frac{1}{k_x} d_y \text{sen} \alpha \\ \frac{d}{fk_y} (i_0 \text{sen} \alpha - j_0 \cos \alpha) + \frac{1}{k_y} d_x \text{sen} \alpha - \frac{1}{k_y} d_y \cos \alpha \end{bmatrix}. \quad (7.20)$$

Fazendo:

$$\mathbf{A} = \begin{bmatrix} \frac{d}{f} \cdot \frac{k_i}{k_x} \cdot \cos \alpha & \frac{d}{f} \cdot \frac{k_j}{k_x} \cdot \text{sen} \alpha \\ -\frac{d}{f} \cdot \frac{k_i}{k_y} \cdot \text{sen} \alpha & \frac{d}{f} \cdot \frac{k_j}{k_y} \cdot \cos \alpha \end{bmatrix} \text{ e} \quad (7.21)$$

$$\mathbf{B} = \begin{bmatrix} \frac{d}{fk_x} (-i_0 \cos \alpha - j_0 \text{sen} \alpha) - \frac{1}{k_x} d_x \cos \alpha - \frac{1}{k_x} d_y \text{sen} \alpha \\ \frac{d}{fk_y} (i_0 \text{sen} \alpha - j_0 \cos \alpha) + \frac{1}{k_y} d_x \text{sen} \alpha - \frac{1}{k_y} d_y \cos \alpha \end{bmatrix}, \quad (7.22)$$

tem-se ainda, da equação (7.20):

$$\begin{bmatrix} x \\ y \end{bmatrix} = -\mathbf{A} \cdot \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{B}. \quad (7.23)$$

Esta relação foi obtida supondo a câmera fixa na posição $(0, 0)$ do *plotter*. Para a câmera numa posição arbitrária (x_c, y_c) , a relação entre (x, y) e (i, j) pode ser obtida fazendo-se a mudança de variáveis $\bar{x} = x - x_c$ e $\bar{y} = y - y_c$, de forma que a equação (7.23) se torna:

$$\begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} = -\mathbf{A} \cdot \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{B}. \quad (7.24)$$

Em particular, se a câmera é deslocada da posição (x_1, y_1) para a posição (x_2, y_2) , a posição da imagem de (x, y) varia de (i_1, j_1) para (i_2, j_2) segundo as relações:

$$\left\{ \begin{array}{l} \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix} = -\mathbf{A} \cdot \begin{bmatrix} i_1 \\ j_1 \end{bmatrix} + \mathbf{B} \\ \begin{bmatrix} x - x_2 \\ y - y_2 \end{bmatrix} = -\mathbf{A} \cdot \begin{bmatrix} i_2 \\ j_2 \end{bmatrix} + \mathbf{B} \end{array} \right. . \quad (7.25)$$

Subtraindo a primeira equação da segunda, tem-se:

$$\begin{bmatrix} x - x_2 \\ y - y_2 \end{bmatrix} - \begin{bmatrix} x - x_1 \\ y - y_1 \end{bmatrix} = -\mathbf{A} \cdot \begin{bmatrix} i_2 \\ j_2 \end{bmatrix} + \mathbf{B} + \mathbf{A} \cdot \begin{bmatrix} i_1 \\ j_1 \end{bmatrix} - \mathbf{B}, \quad (7.26)$$

o que se reduz a:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}, \quad (7.27)$$

com $\Delta x = x_2 - x_1$, $\Delta y = y_2 - y_1$, $\Delta i = i_2 - i_1$ e $\Delta j = j_2 - j_1$.

A equação (7.27) relaciona a variação $(\Delta x, \Delta y)$ na posição da câmera, em palavras de comando de 8 *bits*, à conseqüente variação $(\Delta i, \Delta j)$ na posição da imagem de um ponto da trajetória, em número de *pixels*. Esta relação é utilizada pelo módulo de visão computacional para, a partir dos pontos da trajetória extraídos de uma imagem binarizada, determinar o quanto se deve deslocar o atuador para que aqueles pontos da imagem passem a se situar sobre uma posição de referência (i_{ref}, j_{ref}) pré-estabelecida.

7.1.2 Calibração do sistema óptico

No sistema de rastreamento de trajetórias implementado, a calibração do módulo de aquisição de imagens consiste em determinar, em um procedimento experimental específico, os quatro elementos da matriz \mathbf{A} , que relaciona $(\Delta x, \Delta y)$ a $(\Delta i, \Delta j)$ segundo a equação (7.27). A avaliação direta desta matriz, com base na igualdade (7.21), não é

possível, já que os valores de f , d e α não são conhecidos de antemão. De fato, a faixa de possíveis valores da distância f entre o centro óptico e o plano de imagem é fornecida pelo fabricante da câmera, mas o valor exato desta grandeza depende do ajuste manual da posição da lente. Já d e α dependem da posição e orientação da câmera em relação aos eixos x e y do espaço de trabalho do atuador, e enquanto há uma grande dificuldade prática em medir o ângulo α de inclinação entre o plano de imagem e o de trajetória, a distância d simplesmente não pode ser medida de forma direta com boa precisão, já que não se tem acesso ao centro óptico no interior da câmera.

Assim, o procedimento de calibração empregado consistiu em capturar diferentes imagens de um mesmo ponto de referência, representado por uma cruz, a partir de diferentes posições. Deslocando-se o atuador (e conseqüentemente a câmera nele fixada) entre as $N + 1$ posições (x_0, y_0) a (x_N, y_N) , foram calculados os N deslocamentos $(\Delta x_k, \Delta y_k) = (x_k - x_0, y_k - y_0)$ e as conseqüentes variações na posição do *pixel* contendo o centro da cruz, dadas por $(\Delta i_k, \Delta j_k) = (i_k - i_0, j_k - j_0)$. A matriz \mathbf{A} a ser calculada deve ser tal que:

$$\begin{bmatrix} \Delta x_k \\ \Delta y_k \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} \Delta i_k \\ \Delta j_k \end{bmatrix}, \quad \forall k \in 1, 2, \dots, N, \quad (7.28)$$

ou seja:

$$\begin{bmatrix} \Delta x_1 & \Delta x_2 & \dots & \Delta x_N \\ \Delta y_1 & \Delta y_2 & \dots & \Delta y_N \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} \Delta i_1 & \Delta i_2 & \dots & \Delta i_N \\ \Delta j_1 & \Delta j_2 & \dots & \Delta j_N \end{bmatrix}. \quad (7.29)$$

Designando por Δp_a as variações nas posições do atuador e por Δp_i as variações nas posições da imagem do ponto de referência, ou seja:

$$\Delta p_a = \begin{bmatrix} \Delta x_1 & \Delta x_2 & \dots & \Delta x_N \\ \Delta y_1 & \Delta y_2 & \dots & \Delta y_N \end{bmatrix} \text{ e } \Delta p_i = \begin{bmatrix} \Delta i_1 & \Delta i_2 & \dots & \Delta i_N \\ \Delta j_1 & \Delta j_2 & \dots & \Delta j_N \end{bmatrix}, \quad (7.30)$$

obtém-se:

$$\Delta p_a = \mathbf{A} \cdot \Delta p_i. \quad (7.31)$$

A equação (7.31) constitui um sistema linear em \mathbf{A} . A solução pelo método dos mínimos quadrados [31] [30] conduz à matriz de calibração procurada:

$$\mathbf{A} = \Delta p_a \cdot \Delta p_i^T \cdot (\Delta p_i \cdot \Delta p_i^T)^{-1}. \quad (7.32)$$

Da aplicação do método aqui descrito ao sistema implementado, obteve-se:

$$\mathbf{A} = \begin{bmatrix} 0,1610 & 0,0127 \\ 0,0164 & -0,2411 \end{bmatrix}. \quad (7.33)$$

Assim, a transformação dada pela equação (7.27) fica:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0,1610 & 0,0127 \\ 0,0164 & -0,2411 \end{bmatrix} \cdot \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix}. \quad (7.34)$$

A equação (7.34) permite determinar, em particular, que uma variação de 0,1610 na coordenada x e de 0,2411 na coordenada y correspondem, respectivamente, à variação de uma unidade nas coordenadas i e j do *pixel* em que se forma a imagem de um ponto fixo na trajetória. Da relação entre as variações das entradas nos dois canais do *plotter* e os deslocamentos, em milímetros, do TCP — equações (A.1) e (A.2) — conclui-se que uma mudança na posição do TCP de apenas 0.1441mm, se fosse possível, resultaria na variação de um *pixel* na imagem obtida. Esta informação é importante para a avaliação do erro mínimo de posicionamento que pode ser detectado pelo sistema de visão implementado, conforme será discutido no capítulo 8.

7.2 CÁLCULO DAS POSIÇÕES DESEJADAS DO TCP CORRESPONDENTES AOS *PIXELS* EXTRAÍDOS NA BUSCA CIRCULAR

No sistema de rastreamento implementado (figura 2.6), a função do elemento de controle é determinar as coordenadas x e y de referência de entrada para o circuito de posicionamento do atuador. A informação para tanto disponível é representada pelas coordenadas i e j dos L *pixels* extraídos de cada imagem durante a aplicação do algoritmo de busca circular. Trata-se, portanto, de obter, para cada imagem, L posições ao longo da trajetória desejada do atuador, a partir da projeção no plano de retina da câmera dos pontos correspondentes da trajetória de entrada.

Conforme já indicado no capítulo 2, o objetivo de todo o sistema é posicionar o atuador de forma que sucessivos pontos ao longo da linha traçada sejam, nas sucessivas imagens adquiridas, projetados num mesmo *pixel* de referência. Desta forma, numa única imagem, a distância em número de *pixels* entre um ponto considerado na trajetória e a referência adotada (por exemplo, o centro da imagem) pode ser interpretada como um sinal de erro entre a posição atual do TCP e a posição desejada em diversos instantes futuros.

Neste sentido, deve-se inicialmente converter as distâncias entre os *pixels* obtidos na busca circular ao *pixel* de referência em deslocamentos sucessivos do TCP, em termos das coordenadas x e y de entrada do sistema de posicionamento. A relação entre os dois sinais é dada pela matriz de calibração \mathbf{A} , conforme a equação (7.27).

Assim, sejam i_1, i_2, \dots, i_L e j_1, j_2, \dots, j_L as coordenadas dos L *pixels* extraídos durante a busca circular. O ponto de partida — (i_1, j_1) — representa a posição já atingida pelo TCP; os *pixels* seguintes — (i_2, j_2) a (i_L, j_L) — representam as posições futuras a

serem percorridas, e são eles que definem os próximos deslocamentos necessários. Da equação (7.27):

$$\begin{bmatrix} \Delta x_2 & \Delta x_3 & \cdots & \Delta x_L \\ \Delta y_2 & \Delta y_3 & \cdots & \Delta y_L \end{bmatrix} = \mathbf{A} \cdot \begin{bmatrix} i_2 - i_{ref} & i_3 - i_{ref} & \cdots & i_L - i_{ref} \\ j_2 - j_{ref} & j_3 - j_{ref} & \cdots & j_L - j_{ref} \end{bmatrix}, \quad (7.35)$$

onde i_{ref} e j_{ref} são as coordenadas do *pixel* de referência (centro da imagem) e $(\Delta x_k, \Delta y_k)$ é o deslocamento do TCP em relação à posição atual necessário para que ele atinja o ponto representado pelo *pixel* (i_k, j_k) .

Como o rastreamento tem início a partir de uma posição conhecida do TCP, conforme ressaltado no capítulo 2, é possível determinar as novas posições a partir dos deslocamentos assim calculados para cada imagem. Se (\hat{x}, \hat{y}) é a posição do TCP quando da aquisição da imagem da qual foram obtidos os deslocamentos $(\Delta x_2, \Delta y_2)$ a $(\Delta x_L, \Delta y_L)$, tem-se:

$$\begin{bmatrix} x_2 & x_3 & \cdots & x_L \\ y_2 & y_3 & \cdots & y_L \end{bmatrix} = \begin{bmatrix} x_2 + \hat{x} & x_3 + \hat{x} & \cdots & x_L + \hat{x} \\ y_2 + \hat{y} & y_3 + \hat{y} & \cdots & y_L + \hat{y} \end{bmatrix}, \quad (7.36)$$

onde (x_2, y_2) a (x_L, y_L) são as posições no espaço de trabalho representadas pelo *pixels* (i_2, j_2) a (i_L, j_L) , respectivamente.

7.3 CÁLCULO DAS REFERÊNCIAS DE ENTRADA DO SISTEMA DE POSICIONAMENTO UNIFORMEMENTE ESPAÇADAS

Uma vez determinadas as coordenadas x e y dos pontos correspondentes aos L *pixels* extraídos na busca circular, deve-se obter uma seqüência de pontos igualmente espaçados ao longo de uma linha contínua que une os pontos originais. O objetivo, conforme ressaltado no capítulo 6, é garantir uma velocidade uniforme do manipulador robótico, ao ser enviado a intervalos regulares de tempo aos pontos da nova seqüência gerada.

A seqüência de pontos equidistantes pode ser obtida interpolando-se as coordenadas x_k e y_k dos $L - 1$ pontos obtidos da equação (7.36), $2 \leq k \leq L$, em função da distância ao ponto de partida a ser percorrida ao longo da trajetória. A aplicação das 2 funções interpoladoras obtidas a distâncias uniformemente crescentes fornece então a seqüência de pontos equidistantes procurada.

Alternativamente, o processo de interpolação pode ser efetuado diretamente sobre as variações Δx_k e Δy_k nas coordenadas x e y — equação (7.35), antes que se efetue a soma destas variações à posição atual do atuador para o cálculo das entradas do sistema de posicionamento. Esta abordagem, adotada no sistema implementado, é apresentada a seguir.

A variável associada à abscissa, no processo de interpolação, é a distância d_k do k -ésimo ao primeiro pontos, medida ao longo da trajetória. Uma estimativa desta distância é obtida pela soma das distâncias entre todos os pontos consecutivos extraídos, do primeiro ao k -ésimo. Assim:

$$d_k = \sum_{n=2}^k \sqrt{[D_x \cdot (\Delta x_n - \Delta x_{n-1})]^2 + [D_y \cdot (\Delta y_n - \Delta y_{n-1})]^2}, \quad (7.37)$$

onde D_x e D_y são as variações em milímetros da posição do atuador para um deslocamento unitário, em palavras de comando de 8 *bits*, nas coordenadas x e y , respectivamente¹.

A interpolação dos L pares de pontos $(d_k, \Delta x_k)$ e $(d_k, \Delta y_k)$ conduz a duas funções contínuas $p_{\Delta x}(d)$ e $p_{\Delta y}(d)$, cujos valores podem ser avaliados em qualquer ponto d no intervalo $[d_1, d_L]$. Em particular, pode-se obter os M pares de coordenadas dados por:

$$\Delta x'_k = p_{\Delta x} \left(\frac{k}{M-1} \cdot d_L \right) \text{ e} \quad (7.38)$$

$$\Delta y'_k = p_{\Delta y} \left(\frac{k}{M-1} \cdot d_L \right), \quad (7.39)$$

$\forall 0 \leq k \leq M-1$. Nesta situação, a distância entre dois pontos consecutivos na nova seqüência $\langle (x'_k, y'_k) \rangle$ gerada é constante, dada por:

$$d \left((x'_k, y'_k), (x'_{k-1}, y'_{k-1}) \right) = \frac{1}{M-1} \cdot d_L. \quad (7.40)$$

A função interpoladora adotada foi o polinômio de 3^o grau. O apêndice C mostra como podem ser calculados os coeficientes do polinômio de grau arbitrário que melhor interpola L pontos de coordenadas conhecidas (α_k, β_k) , $1 \leq k \leq L$.

7.4 DETERMINAÇÃO DO NÚMERO DE POSIÇÕES DE REFERÊNCIA EXTRAÍDAS DE CADA IMAGEM

Considerando-se que o envio do TCP às posições de referência calculadas a partir de uma única imagem é realizado em malha aberta, só sendo fechada a malha nos instantes de aquisição de novas imagens, o ideal é reduzir ao máximo o valor de L , na equação (7.36). Entretanto, a adoção de uma taxa de atualização das entradas do sistema de posicionamento igual à taxa de aquisição de imagens se mostrou inadequada ao rastreamento: mesmo a máxima taxa de aquisição de imagens permitida pela câmera (ver apêndice A) é insuficiente para garantir, nesta condição, que o TCP não desacelere consideravelmente ao atingir cada posição especificada, sendo posteriormente acelerado à posição seguinte. As desvantagens desta variação na velocidade do TCP são discutidas no capítulo 6.

¹Das equações (A.1) e (A.2), $D_x = 0,8953mm$ e $D_y = 0,5977mm$.

Experimentalmente, constatou-se que a extração de cerca de 4 pontos por imagem ($L = 4$) foi suficiente para a obtenção de uma taxa adequada de atualização das referências de entrada. Ainda assim, optou-se pela extração de 10 pontos durante a busca circular, e de um cálculo de um igual número de referências de entrada após o processo de interpolação apresentado na seção 7.3. Isto permitiu a avaliação de uma série de alternativas durante os testes dos algoritmos de visão implementados, sem a preocupação de otimizar o tempo de execução de cada etapa (de fato, os processamentos realizados em cada imagem adquirida reduzem a taxa efetiva de aquisição, de forma que se deve otimizar os algoritmos de visão sempre que se queira minimizar o número de referências obtidas por imagem). A adoção de um número maior de pontos permitiu, no entanto, que o rastreamento fosse realizado com boa precisão frente à resolução espacial do *plotter*, como será analisado no capítulo 8; a avaliação apenas a cada 10 pontos da diferença entre a posição efetiva do TCP e a posição desejada no mesmo instante não se mostrou uma fonte importante de erros.

Capítulo 8

RESULTADOS EXPERIMENTAIS

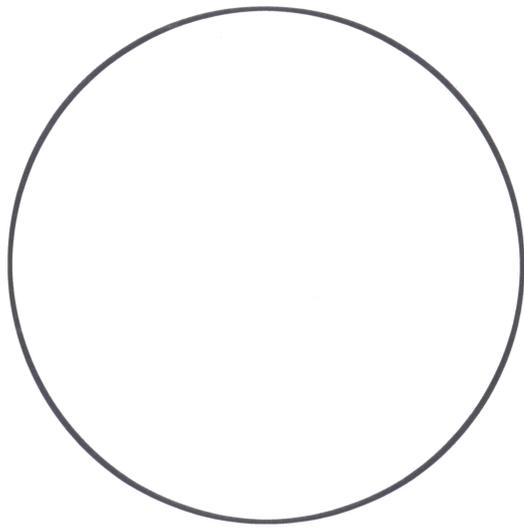
O sistema proposto nos capítulos anteriores foi submetido ao rastreamento de diferentes trajetórias de entrada. Seu desempenho foi então avaliado a partir da comparação destas trajetórias com aquelas efetivamente seguidas pelo TCP do *plotter*, registradas em papel milimetrado por uma caneta fixa neste dispositivo durante todo o rastreamento. Em alguns casos, a adoção de trajetórias definidas a partir de expressões matemáticas analíticas permitiu avaliar o erro numérico entre os pontos que as constituem e as referências de entrada do sistema de posicionamento, calculadas pelos algoritmos de visão computacional ao longo do trajeto da câmera.

Este capítulo trata dos resultados experimentais obtidos para quatro destas trajetórias, apresentadas na seção 8.1. Para uma das trajetórias consideradas, são ilustrados os resultados intermediários de todas as etapas envolvidas, desde o pré-processamento das imagens até o cálculo de referências de posição uniformemente espaçadas¹. As seções 8.2 a 8.5 tratam individualmente destas etapas. Finalmente, a seção 8.6 apresenta todas as referências de entrada do sistema de posicionamento calculadas durante os rastreamentos, juntamente com as trajetórias efetivamente seguidas pelo TCP em cada caso. Os erros observados em relação às trajetórias de entrada são então analisados tendo em vista a precisão do *plotter* e a correção das posições pelo sistema de visão.

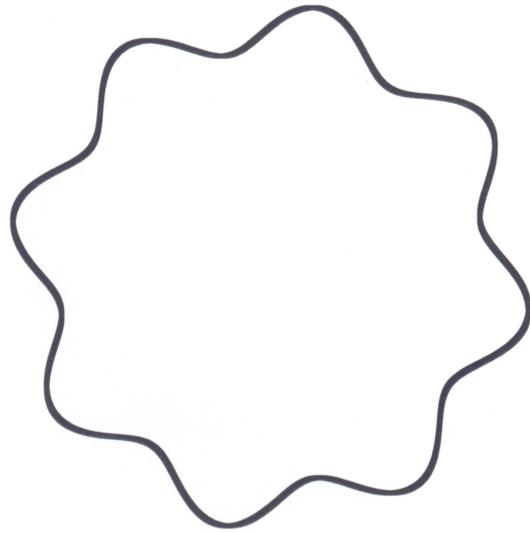
8.1 APRESENTAÇÃO DE ALGUMAS TRAJETÓRIAS DE ENTRADA EMPREGADAS

Os resultados de rastreamento aqui apresentados referem-se a quatro trajetórias de entrada. A figura 8.1 apresenta as imagens digitalizadas das quatro curvas reais apresentadas ao sistema. Observe-se que não se trata, aqui, de imagens adquiridas pela câmera durante o rastreamento, mas sim de toda a curva posicionada sobre o *plotter* como descrito na figura 2.2.

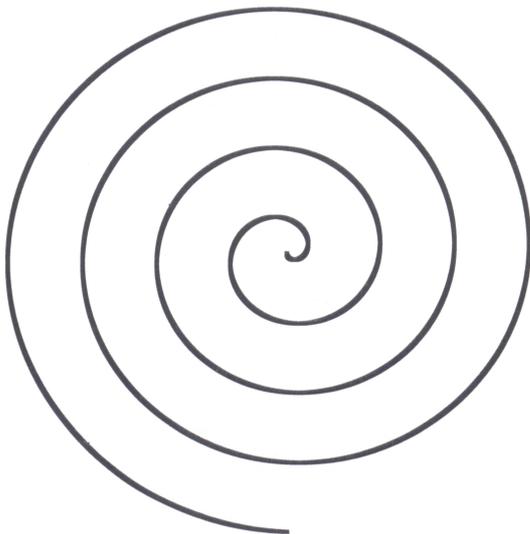
¹Para as outras três trajetórias, os resultados intermediários são apresentados no apêndice D.



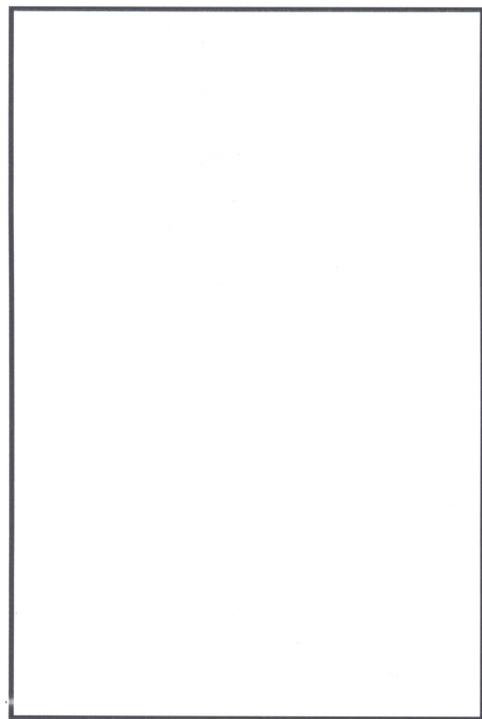
(a)



(b)



(c)



(d)

Figura 8.1: Imagens digitalizadas de quatro trajetórias submetidas ao sistema de rastreamento: (a) circunferência com $60,5mm$ de raio; (b) curva fechada com ondulações; (c) espiral; (d) retângulo com $152mm$ e $100mm$ de lados.

A escolha destas trajetórias foi feita de forma a favorecer uma análise do desempenho do sistema. Três delas são trajetórias fechadas, o que significa que o rastreamento pode prosseguir indefinidamente, sendo realizadas diversas voltas ao longo do percurso; desta forma, é possível verificar se os resultados obtidos se reproduzem ou se há um efeito acumulativo de erros ao longo do rastreamento. A outra entrada, a espiral da figura 8.1-(c), exige que o TCP se movimente em todas as direções possíveis e em uma grande região do espaço de trabalho do *plotter*, já que o raio aumenta do início ao fim da trajetória.

Dentre as trajetórias fechadas, adotou-se uma circunferência, uma curva com ondulações e um retângulo. A primeira também exige que o TCP se desloque em todas as direções, enquanto que a segunda exige mudanças rápidas de direção devido às várias ondulações presentes. A dificuldade inerente ao rastreamento do retângulo, por sua vez, diz respeito aos vértices presentes, que teoricamente exigiriam uma variação instantânea de direção.

Na realização dos testes, as trajetórias fechadas foram percorridas durante cerca de três voltas completas, sendo então interrompido o rastreamento. No caso da espiral, o rastreamento teve início no centro e foi interrompido assim que o TCP atingiu o ponto mais distante da curva.

Nas seções 8.2 a 8.5, são ilustradas todas as etapas intermediárias do rastreamento para uma das imagens adquiridas da trajetória da figura 8.1-(a).

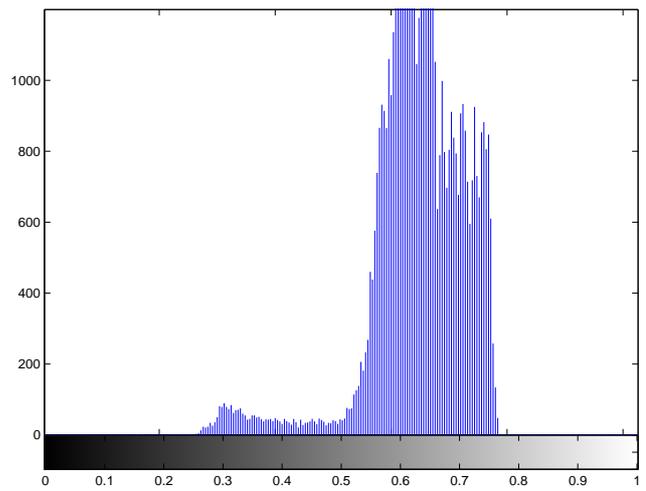
8.2 RESULTADOS DE PRÉ-PROCESSAMENTO DAS IMAGENS

Conforme descrito nos capítulos anteriores, a primeira etapa do rastreamento, após a aquisição de uma imagem da trajetória de entrada, é a compensação da não-uniformidade da iluminação durante sua geração. O objetivo é obter uma nova imagem que represente uma medida da reflectância de cada ponto da cena projetado no plano de retina. Na imagem pré-processada deve ser então possível a separação das duas distribuições de luminâncias associadas à trajetória e ao fundo, por meio do algoritmo automático proposto nos capítulos 3 e 4.

Na figura 8.2-(a), é ilustrada uma das imagens adquiridas da trajetória 8.1-(a) durante o rastreamento. No histograma correspondente — figura 8.2-(b) — observa-se uma considerável sobreposição das distribuições associadas à trajetória e ao fundo, devido à influência das condições de iluminação. Para a compensação destas condições, dispõe-se então de um padrão de iluminação, obtido previamente ao rastreamento pelo procedimento descrito na seção 5.3. A figura 8.3-(a) apresenta o padrão utilizado no pré-processamento das imagens durante todo o rastreamento da trajetória em questão.



(a)



(b)

Figura 8.2: (a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.1-(b)). (b) Histograma correspondente.

A divisão *pixel a pixel* da imagem da figura 8.1b pelo padrão de iluminação, segundo a técnica adotada, resulta na imagem da figura 8.3-(b). No histograma correspondente, apresentado na figura 8.4-(b), nota-se claramente um maior distanciamento das duas distribuições de luminâncias.



(a)

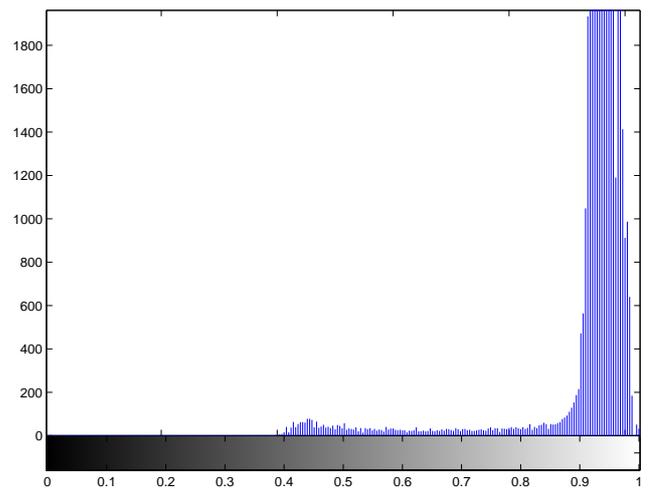


(b)

Figura 8.3: (a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura 8.2-(a). (b) Imagem obtida após a compensação.



(a)



(b)

Figura 8.4: (a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura 8.2-(a). (b) Histograma correspondente.

8.3 RESULTADOS DE BINARIZAÇÃO DAS IMAGENS PROCESSADAS

A etapa seguinte de processamento, ainda na fase dos algoritmos de visão denominada análise de imagens, diz respeito à binarização da imagem compensada, que permite a determinação de quais *pixels* representam pontos da trajetória de entrada. A determinação prévia de alguns parâmetros estatísticos se faz necessária ao cálculo do limiar de binarização apropriado.

No capítulo 4, foram apresentados os resultados dos cálculos automáticos dos parâmetros estatísticos de diversas imagens, inclusive algumas sintéticas, que permitiram comparar os valores fornecidos pelo algoritmo proposto com os utilizados em sua geração. Na /subsec:resultadosparametros, são apresentados os parâmetros obtidos para a imagem compensada da figura 8.4-(a). O resultado da binarização com base no limiar obtido destes parâmetros é então mostrado na seção 8.3.2.

8.3.1 Valores obtidos para os parâmetros estatísticos das imagens processadas

A tabela 8.1 apresenta os parâmetros obtidos para a imagem 8.4-a pelo algoritmo proposto no capítulo 4. O limiar de binarização correspondente é destacado no histograma da imagem, na figura 8.5.

Observe-se que este limiar leva à correta separação das distribuições associadas ao fundo e à trajetória. Valores de luminância intermediários às duas distribuições, que surgem devido à focalização imperfeita da câmera e causam uma pequena discrepância em relação ao modelo ideal de duas curvas gaussianas, são associados à distribuição da trajetória. Isto já era esperado, já que aquelas luminâncias ocorrem em poucos *pixels* da imagem, relativamente à quantidade de *pixels* do fundo. Assim, eles são atribuídos à distribuição da trajetória, também de ocorrência menor, levando a uma maior variância estimada desta distribuição mas ainda assim permitindo isolar corretamente os *pixels* do fundo.

8.3.2 Resultados do processo de limiarização

A figura 8.6 destaca a imagem binarizada com base no limiar destacado no histograma da figura 8.5. Da comparação com a imagem original, na figura 8.2-(a), percebe-se a classificação correta de todos os *pixels*.

Tabela 8.1: Parâmetros estatísticos referentes à imagem da figura 8.4-(a), utilizados no cálculo do limiar ótimo de binarização.

Média da distribuição associada à trajetória (μ_T^2)	0.63533
Variância da distribuição associada à trajetória (σ_T^2)	0.02366
Média da distribuição associada ao fundo (μ_F^2)	0.95066
Variância da distribuição associada ao fundo (σ_F^2)	0.00042376
Proporção de trajetória na imagem (p_T)	0.073403
<i>Limiar de binarização (L)</i>	0.88045

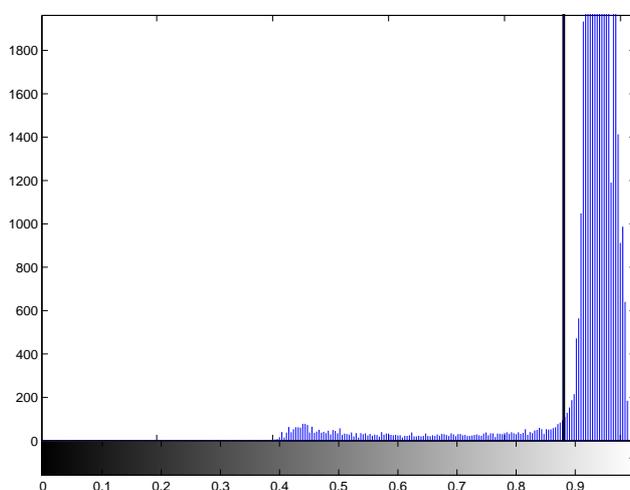


Figura 8.5: Histograma associado à imagem da figura 8.4-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela 8.1).

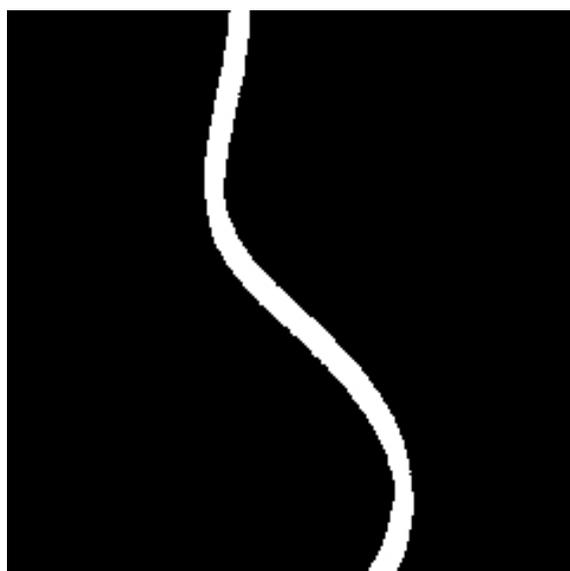


Figura 8.6: Imagem da figura 8.4-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura 8.5.

8.4 *PIXELS* EXTRAÍDOS DAS IMAGENS PELO ALGORITMO DE BUSCA CIRCULAR

O algoritmo de busca circular, aplicado na etapa seguinte, permite extrair da imagem binarizada *pixels* que representam posições a serem ocupadas, em seqüência, pelo TCP. A figura 8.7 destaca estes *pixels*, cujas coordenadas são apresentadas na tabela 8.2.

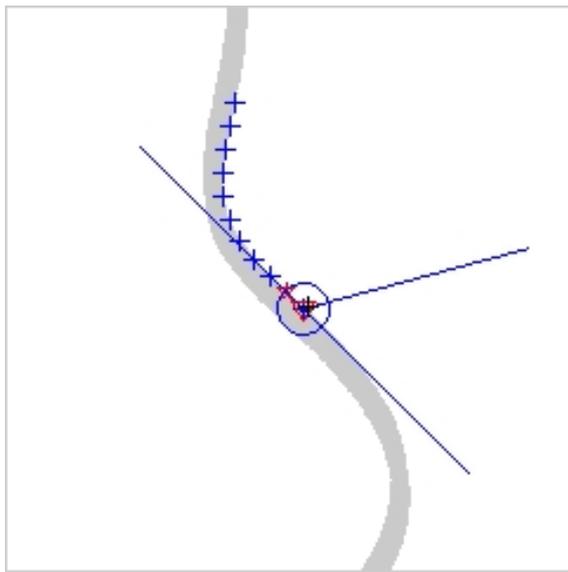


Figura 8.7: Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura 8.6 pela aplicação do algoritmo de busca circular.

Uma observação da figura 8.7 permite concluir que os *pixels* foram determinados corretamente, na seqüência determinada pelo lado externo da linha da trajetória, como proposto no capítulo 6. O ponto de partida da busca circular, no entanto, não coincide exatamente com o centro da imagem — ponto (128, 128). Como o ponto de partida corresponde ao último *pixel* extraído pela busca circular na imagem anterior, e como o TCP é mobilizado de forma que os pontos extraídos ocupem em seqüência o centro da imagem, a diferença observada evidencia um pequeno erro de posicionamento, que pode ser determinado a partir da equação (7.34):

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0,1610 & 0,0127 \\ 0,0164 & -0,2411 \end{bmatrix} \cdot \begin{bmatrix} 129 - 128 \\ 126 - 128 \end{bmatrix} = \begin{bmatrix} 0.1356 \\ 0.4985 \end{bmatrix}, \quad (8.1)$$

onde Δx e Δy são os erros nas coordenadas x e y do TCP, medidas em termos das palavras de 8 *bits* de comando do *plotter* (de 0 a 255).

Este erro é, portanto, bastante inferior à variação mínima das entradas do circuito de posicionamento, tendo em vista a resolução finita dos conversores D/A empregados, e corresponde também a deslocamentos do TCP inferiores aos valores mínimos permitidos pela resolução espacial do *plotter*. A seção 8.6 aborda detalhadamente a magnitude dos

Tabela 8.2: Coordenadas dos *pixels* extraídos da imagem da figura 8.6 pela aplicação do algoritmo de busca circular (figura 8.7).

<i>Pixel</i>	Coordenada <i>i</i>	Coordenada <i>j</i>
1	129	126
2	122	119
3	115	112
4	108	105
5	100	99
6	91	95
7	81	92
8	71	92
9	61	93
10	51	95
11	41	97

erros de posicionamento que podem ocorrer em virtude desta resolução espacial, que , mais do que a resolução dos conversores, é o principal fator determinante da precisão do sistema.

8.5 REFERÊNCIAS DE ENTRADA DO SISTEMA DE POSICIONAMENTO OBTIDAS

Uma vez determinada pela busca circular a seqüência de *pixels* na imagem binarizada, a etapa seguinte é obter as entradas do sistema de posicionamento, pela conversão das distâncias daqueles *pixels* ao centro da imagem em deslocamentos necessários do TCP.

Os deslocamentos obtidos dos *pixels* da figura 8.7 pela equação (7.34) são apresentados no gráfico da figura 8.8 e na tabela 8.3. As distâncias de cada ponto ao ponto de partida, calculadas ao longo da trajetória, são também apresentadas, na mesma tabela.

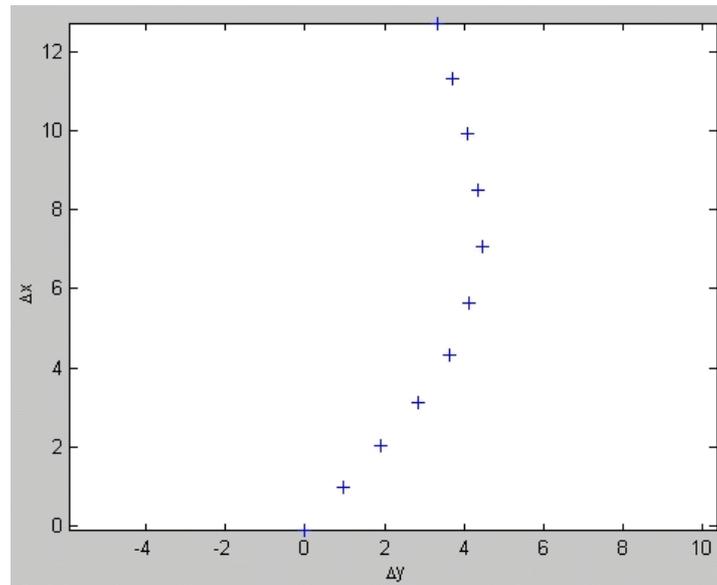


Figura 8.8: Entradas do módulo de controle da posição do TCP obtidas da figura 8.7, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela 8.2) em variações na posição do TCP (transformação dada pela equação (7.34)).

A interpolação dos deslocamentos Δx e Δy em função da distância percorrida d , como proposto na seção 7.3, conduziu então a dois polinômios de terceiro grau:

$$\Delta x(d) = -1,5136 \cdot 10^{-3} \cdot d^3 + 5,0610 \cdot 10^{-2} \cdot d^2 + 5,4632 \cdot 10^{-1} \cdot d - 1,4047 \cdot 10^{-1}. \quad (8.2)$$

$$\Delta y(d) = -5,7276 \cdot 10^{-4} \cdot d^3 + 9,8982 \cdot 10^{-2} \cdot d^2 - 1,8391 \cdot d + 1,2957. \quad (8.3)$$

Assim, a partir destes polinômios puderam ser calculados os novos deslocamentos necessários à obtenção de pontos uniformemente espaçados. Para tanto, os polinômios foram aplicados a distâncias uniformemente crescentes no intervalo definido na tabela 8.3. Os valores obtidos são apresentados no gráfico da figura 8.9 e na figura 8.9.

Tabela 8.3: Entradas do módulo de controle da posição do atuador obtidas da figura 8.7, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela 8.2) em variações na posição do atuador (transformação dada pela equação (7.34)).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	1.0799	-2.0712	1.9546
3	2.2953	-3.6438	3.3926
4	3.5108	-5.2164	4.8306
5	4.8746	-6.5316	6.2827
6	6.3741	-7.3482	7.7112
7	8.0218	-7.9074	9.2238
8	9.6315	-7.7433	10.6683
9	11.2285	-7.3382	12.1185
10	12.8129	-6.6920	13.5886
11	14.3973	-6.0458	15.0587

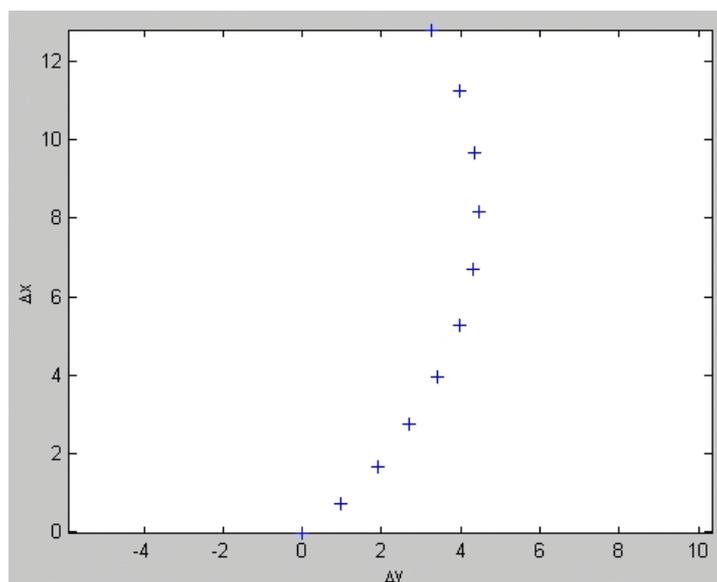


Figura 8.9: Entradas do módulo de controle da posição do TCP uniformemente espaçadas, obtidas pela aplicação dos polinômios (8.2) e (8.3) a distâncias d ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Tabela 8.4: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (8.2) e (8.3) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	0.7918	-1.2512	1.5059
3	1.9226	-3.3609	3.0117
4	3.2209	-5.0452	4.5176
5	4.6557	-6.3158	6.0235
6	6.1960	-7.1844	7.5294
7	7.8108	-7.6627	9.0352
8	9.4690	-7.7626	10.5411
9	11.1397	-7.4957	12.0470
10	12.7918	-6.8737	13.5529
11	14.3943	-5.9085	15.0587

O que se observa, na comparação dos dados das tabelas 8.3 e 8.4 é que a diferença entre os deslocamentos em cada caso poucas vezes se aproxima de uma unidade (menor diferença efetiva durante o rastreamento, devido à precisão finita dos conversores D/A, de 8 *bits*). Este fato, aliás, foi observado durante todo o rastreamento, para todas as trajetórias analisadas.

Isto ocorre porque as distâncias entre pontos consecutivos, para os dados da tabela 8.3, já são aproximadamente uniformes. Como cada *pixel* extraído na busca circular situa-se numa circunferência centrada no *pixel* anterior, as únicas variações possíveis entre as distâncias, em *pixels*, entre pontos consecutivos se dão devido à aproximação em espaço discreto da circunferência de busca, discutida na seção 6.2, ou a variações de seu raio; além disso, se a câmera apresenta resoluções distintas nos eixos i e j , uma mesma distância em *pixels* entre pontos consecutivos pode corresponder a diferentes deslocamentos do TCP, de acordo com a direção em que se dá aquela variação em *pixels*. No entanto, no sistema implementado a robustez dos algoritmos de visão propostos garantiu que não houvesse falhas nas trajetórias binarizadas, de forma que não se fez necessário aumentar o raio da circunferência durante a busca circular, e a câmera utilizada apresenta a mesma resolução em ambos os eixos. Conseqüentemente, as distâncias entre pontos consecutivos na tabela 8.3 apresentam poucas variações, em virtude apenas das aproximações em espaço discreto da circunferência de busca.

Conclui-se que as pequenas diferenças observadas entre os deslocamentos nas tabelas 8.3 e 8.4, que não chegam a influenciar o movimento do TCP devido à precisão finita dos conversores D/A e sobretudo à resolução espacial do *plotter*, não justificam a utilização do processo de interpolação descrito na seção 7.3, no caso do sistema implementado. Entretanto, sua implementação torna os algoritmos de visão propostos mais genéricos, aplicáveis, por exemplo, a sistemas de rastreamento que empregam câmeras com diferentes resoluções nos dois eixos [18] ou em que a aquisição de imagens se dá em condições mais desfavoráveis, levando a falhas durante a binarização.

8.6 TRAJETÓRIAS DO TCP OBTIDAS

As figuras 8.10, 8.11, 8.12 e 8.13 ilustram todas as referências de posição igualmente espaçadas calculadas pelo sistema de visão ao longo do percurso, para as entradas mostradas respectivamente nos itens (a), (b), (c) e (d) da figura 8.1. Conforme ressaltado na seção 8.1, o rastreamento foi executado até que o TCP percorresse o trajeto durante cerca de três voltas, no caso das trajetórias fechadas, ou todo o trajeto uma única vez, no caso da quarta trajetória. Esta escolha resultou num total de 84, 72, 60 e 114 imagens adquiridas durante os trajetos ilustrados, respectivamente.

Como no caso dos deslocamentos na figura 8.9, as coordenadas aqui representadas referem-se às palavras de 8 *bits* de comando do *plotter*, ou seja, estão escalonadas de 0

a 255, tanto para o canal x como para o y . As escalas dos dois eixos foram no entanto determinadas levando-se em conta os deslocamentos correspondentes em milímetros ($x \cdot D_x$ e $y \cdot D_y$, respectivamente²), de forma a manter a mesma proporção das trajetórias efetivas do TCP.

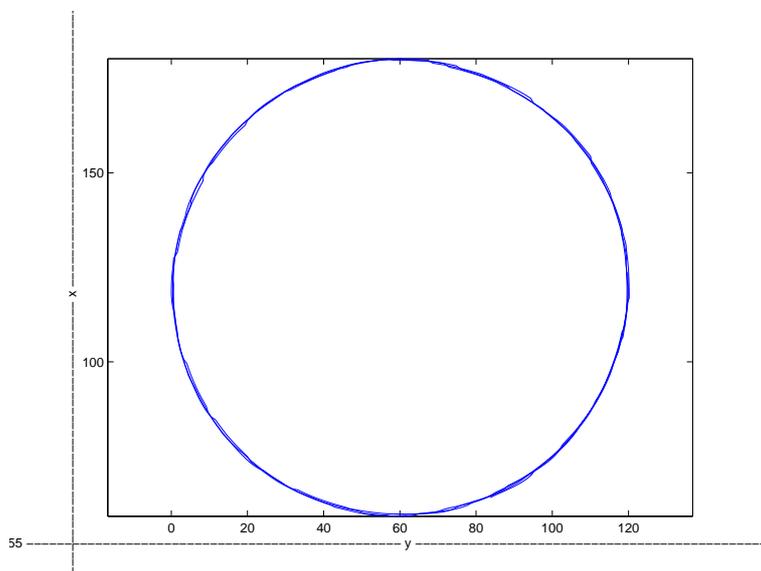


Figura 8.10: Referências de entrada do sistema de posicionamento do TCP obtidas para uma das trajetórias rastreadas (figura 8.1-(a)), a partir de 84 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.

²ver seção A.1.1.

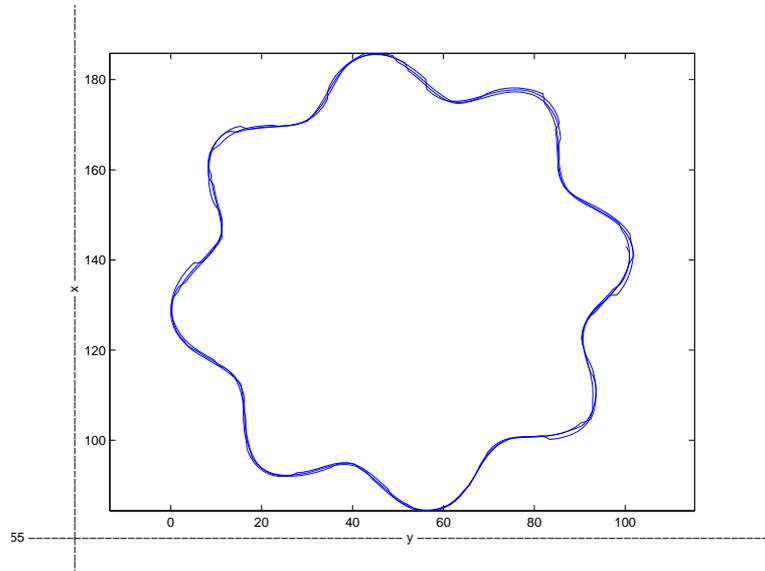


Figura 8.11: Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(b)), a partir de 72 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.

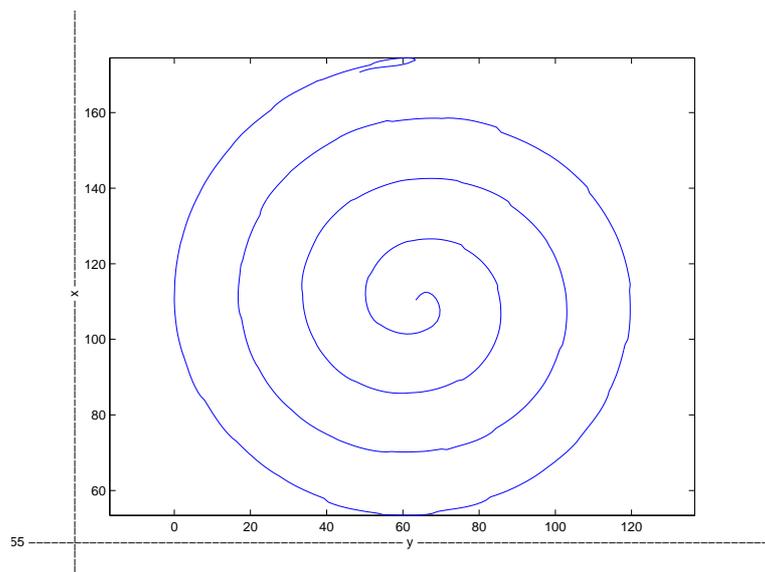


Figura 8.12: Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(c)), a partir de 60 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.

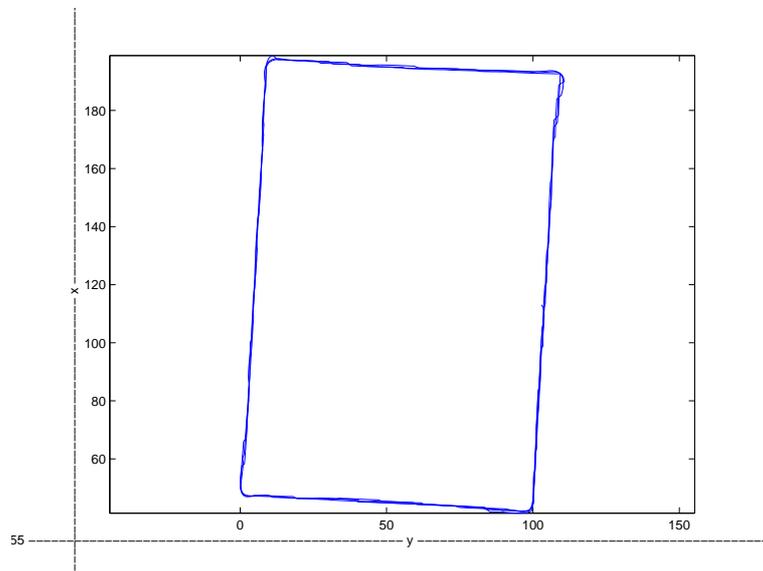
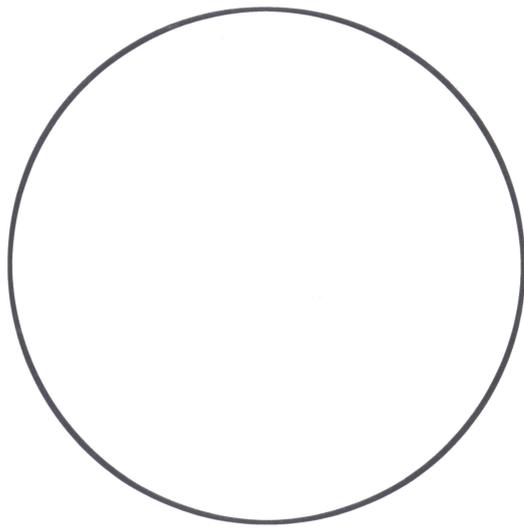
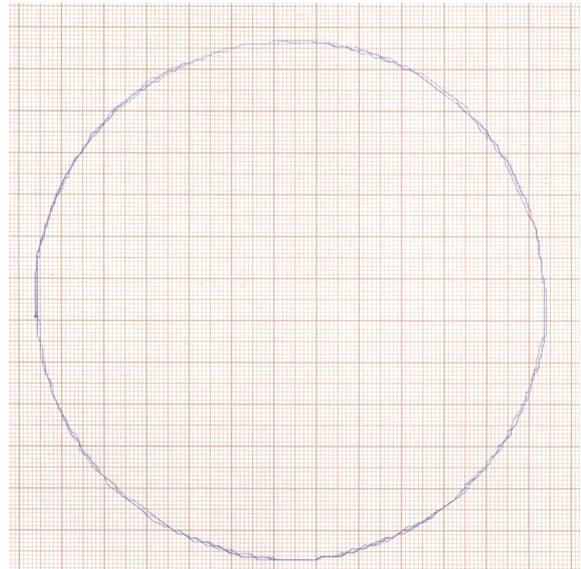


Figura 8.13: Referências de entrada do sistema de rastreamento obtidas para uma das trajetórias rastreadas (figura 8.1-(d)), a partir de 114 imagens adquiridas ao longo do trajeto e pela utilização dos métodos exemplificados para uma imagem nas seções 8.2 a 8.5.

As trajetórias efetivamente traçadas pelo TCP são apresentadas nas figuras 8.14-(b), 8.15-(b), 8.16-(b) e 8.17-(b), obtidas pela digitalização das curvas traçadas, em papel milimetrado, pela caneta fixa ao TCP. Nas figuras 8.18, 8.19, 8.20 e 8.21, por sua vez, estas curvas são sobrepostas às imagens digitalizadas das respectivas trajetórias de entrada, de forma a permitir comparar os percursos reais do TCP aos desejados.

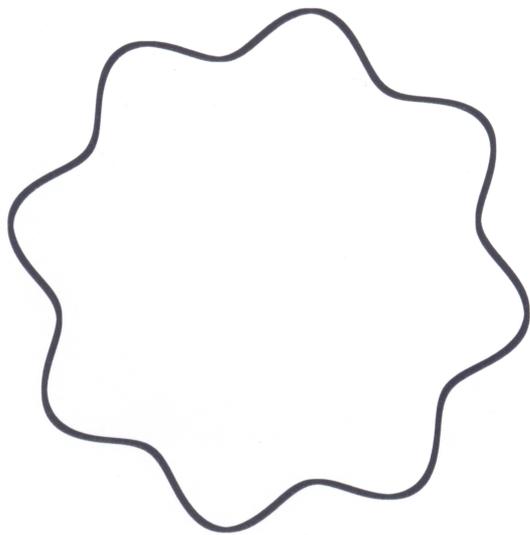


(a)

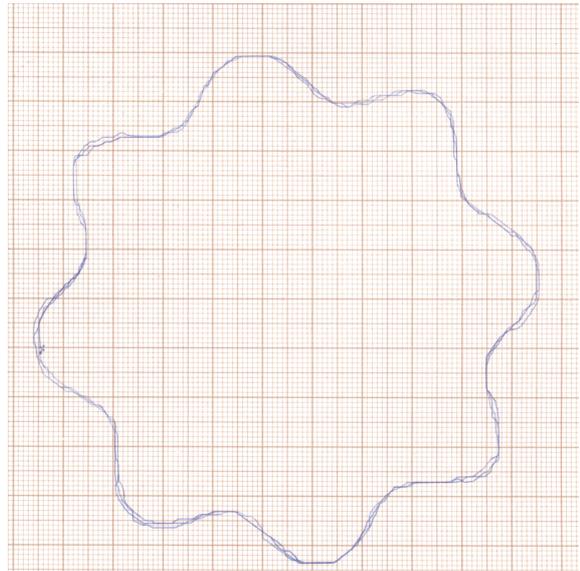


(b)

Figura 8.14: (a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.

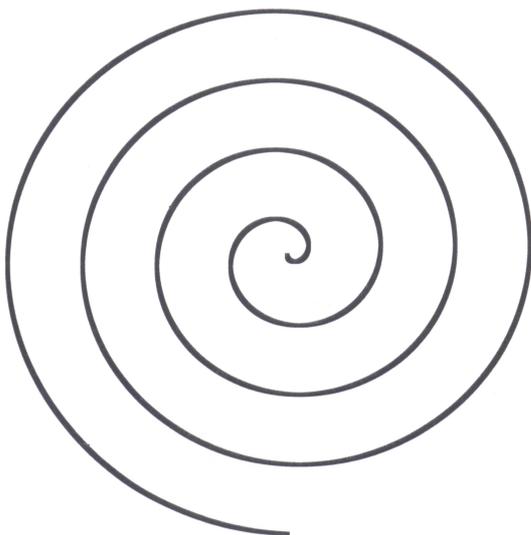


(a)

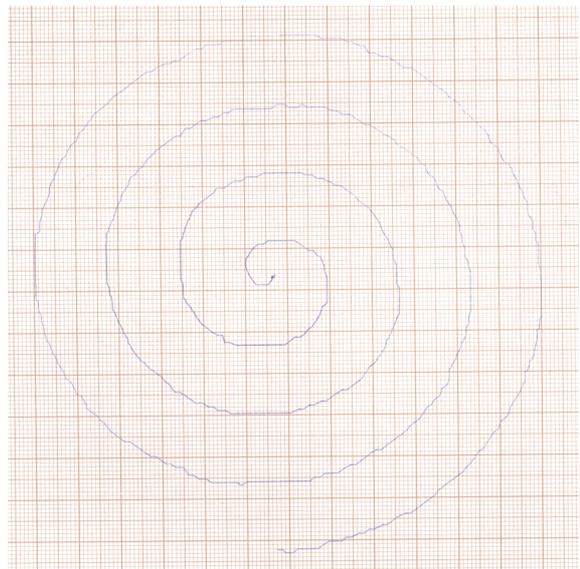


(b)

Figura 8.15: (a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.

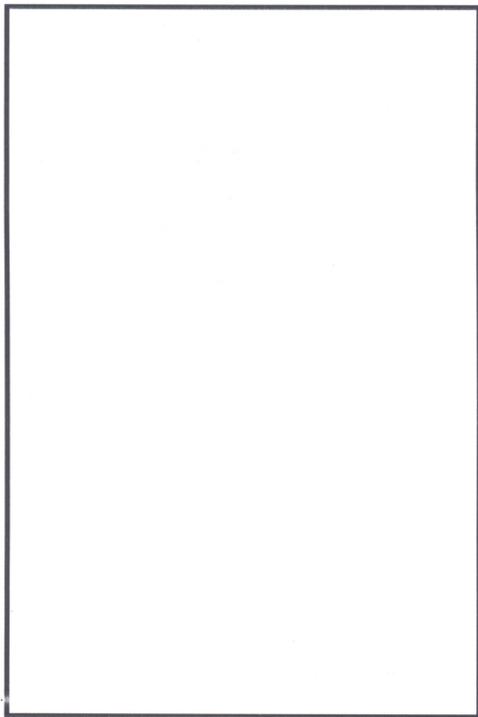


(a)

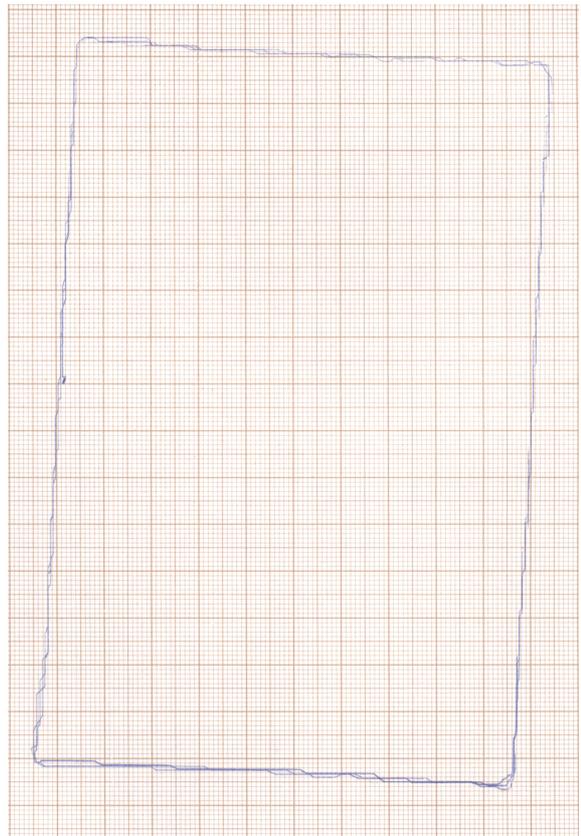


(b)

Figura 8.16: (a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.



(a)



(b)

Figura 8.17: (a) Imagem digitalizada de uma das trajetórias submetidas ao sistema de rastreamento implementado. (b) Trajetória do TCP durante o rastreamento (resultado obtido), registrada em papel milimetrado.

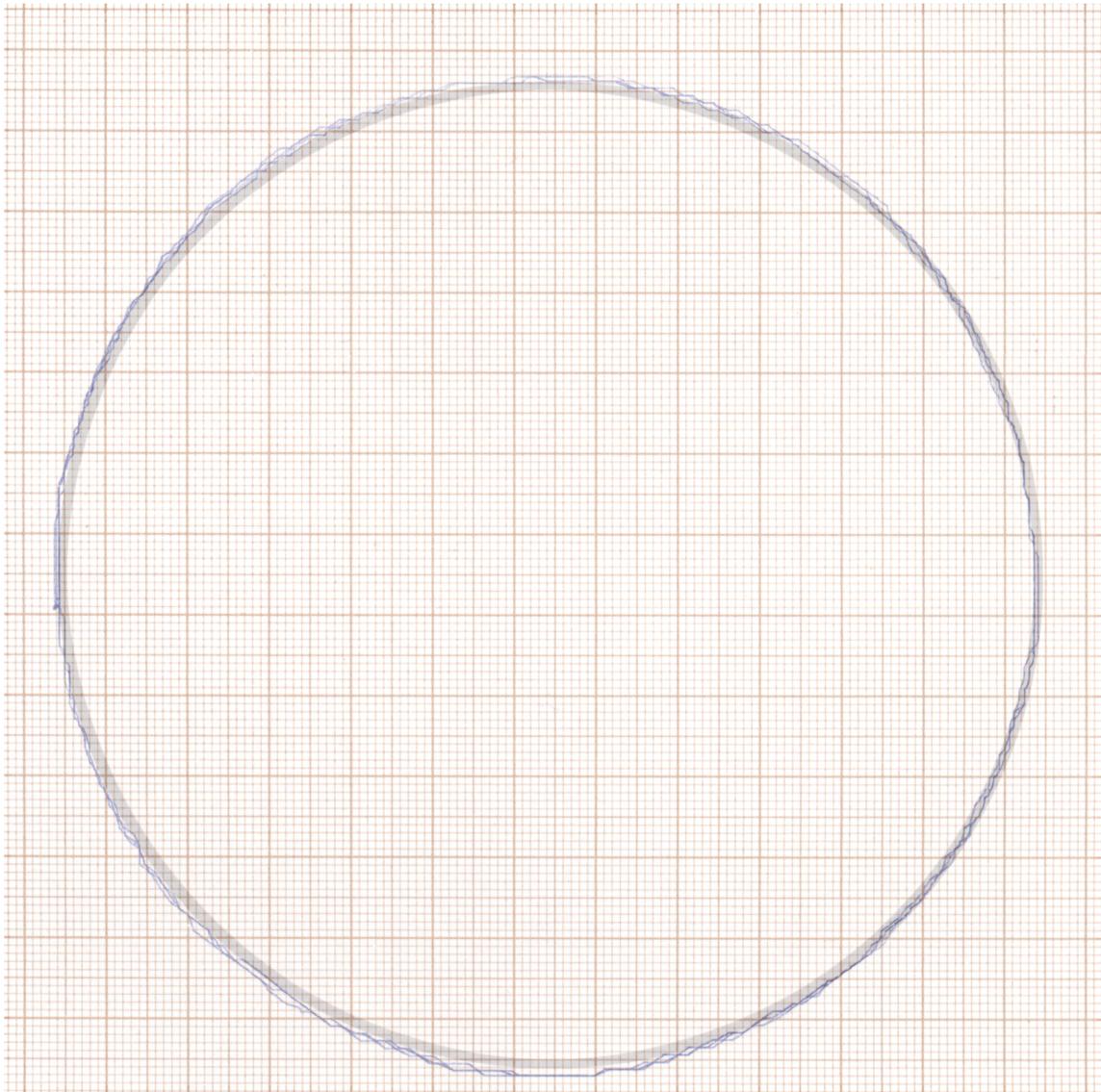


Figura 8.18: Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.14-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.14-(b)).

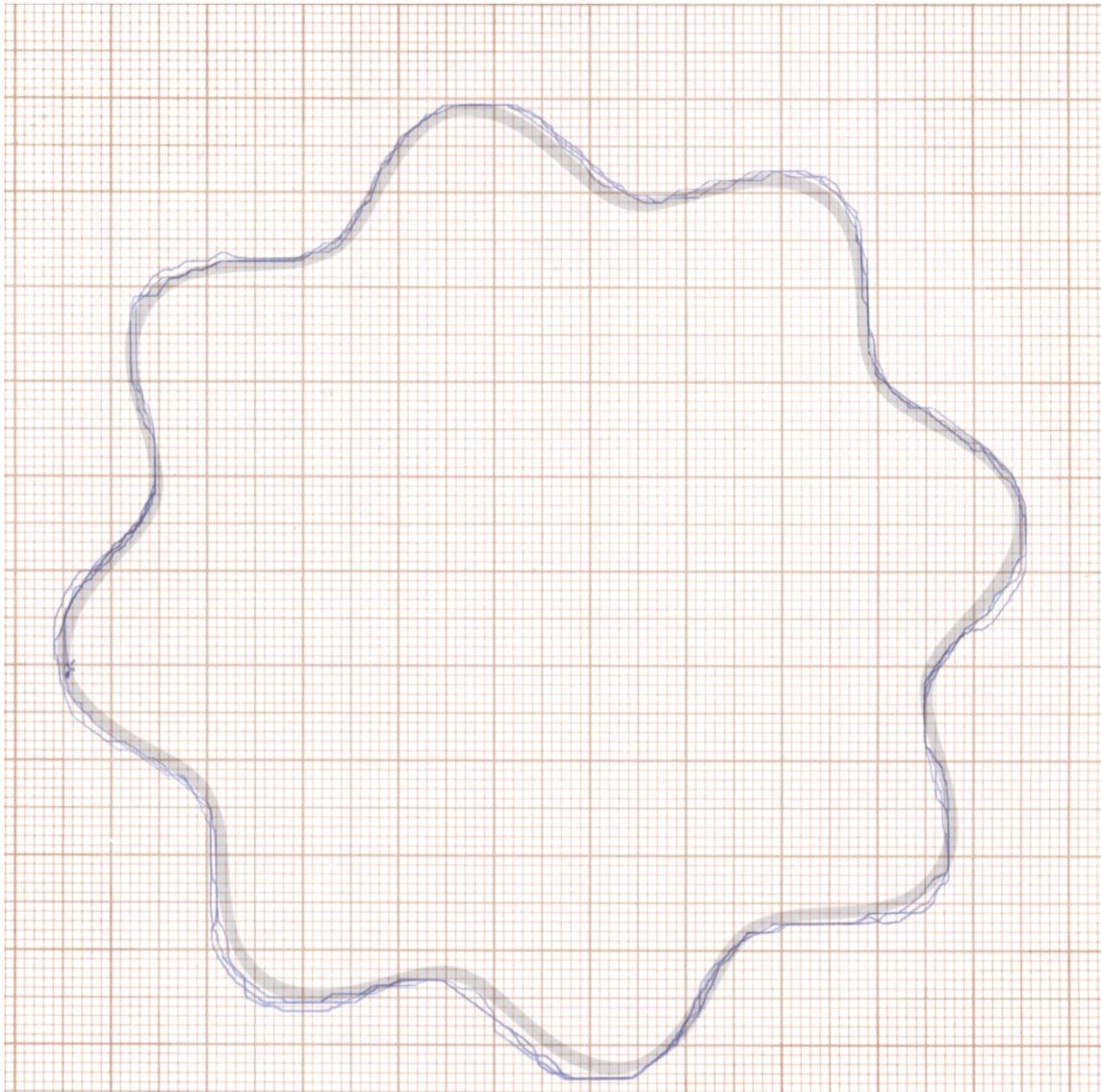


Figura 8.19: Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.15-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.15-(b)).

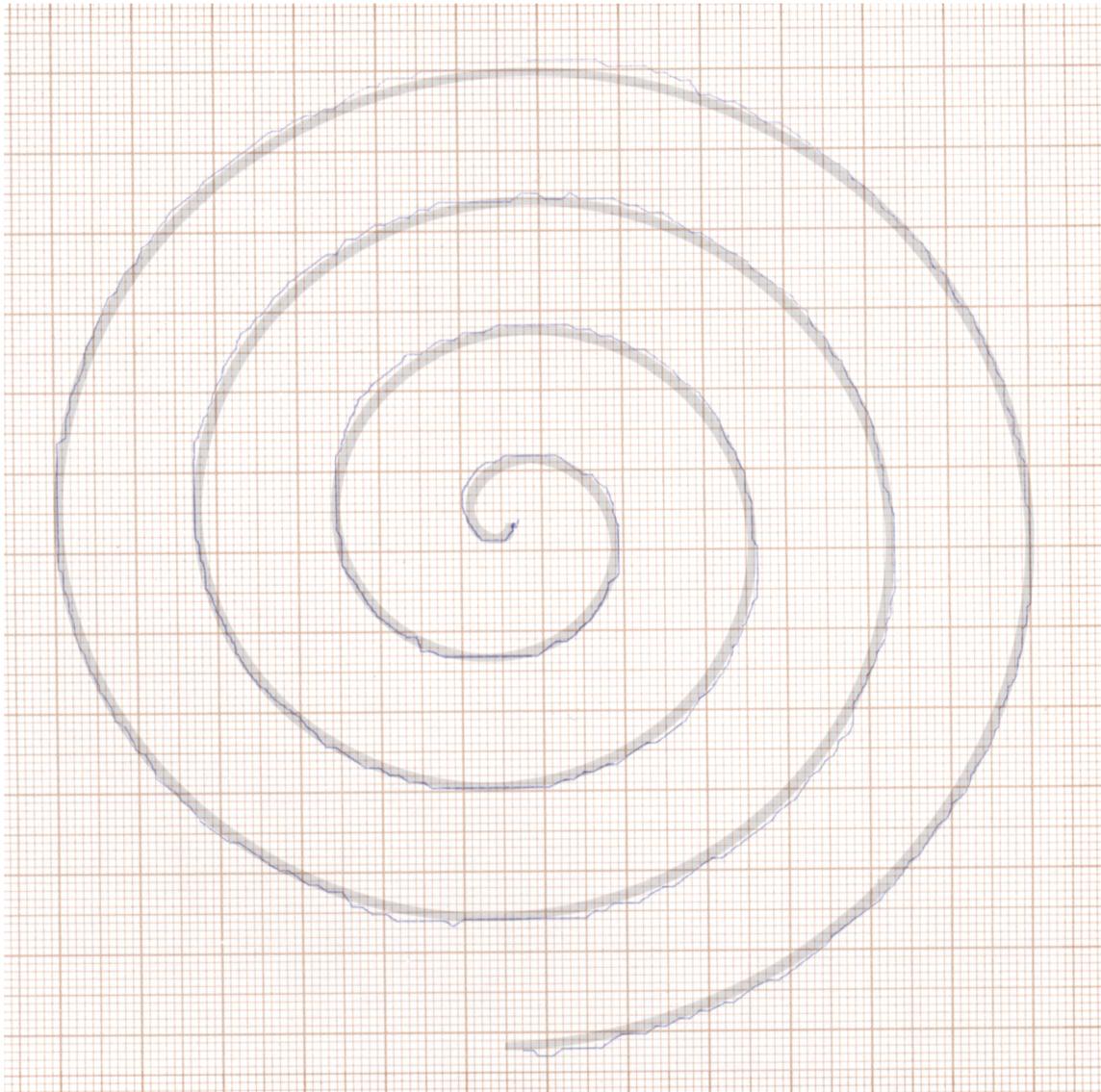


Figura 8.20: Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.16-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.16-(b)).

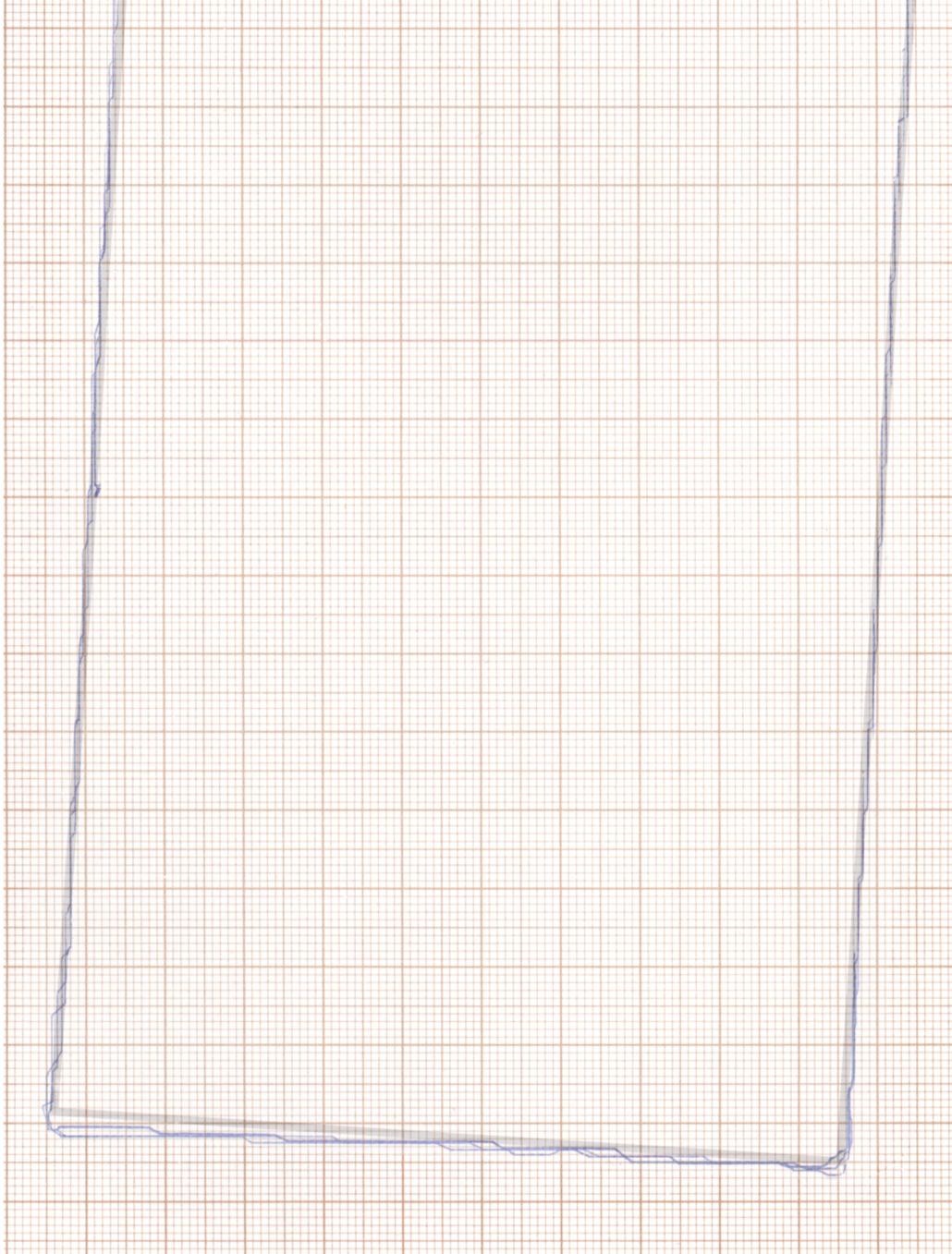


Figura 8.21: Sobreposição de uma das trajetórias submetidas ao sistema de rastreamento (figura 8.17-(a)) com a trajetória efetivamente seguida pelo TCP (figura 8.17-(b)).

Antes que sejam analisados os resultados desta sobreposição, no entanto, observe-se que uma avaliação do erro em milímetros entre as trajetórias efetivas do TCP e as trajetórias de entrada pode ser efetuada ponto a ponto no caso da circunferência e do retângulo, gerados a partir de expressões matemáticas analíticas. No entanto, como o sistema implementado não dispõe de um sensor específico de posição, o único registro da trajetória efetiva do TCP é a curva traçada pela caneta e mostrada nas figuras 8.14-(b), 8.15-(b), 8.16-(b) e 8.17-(b). Assim, a avaliação de erros ponto a ponto teve de ser feita a partir das referências de posição calculadas pelo sistema de visão e apresentadas nas figuras 8.10, 8.11, 8.12 e 8.13. Os erros assim obtidos, porém, são muito próximos aos erros observados nas trajetórias efetivas, conforme será discutido adiante.

Nesta avaliação dos erros associados a cada ponto calculado, deve-se, em uma abordagem genérica, determinar qual posição ao longo da linha contínua da trajetória de entrada

corresponde a cada ponto determinado. O erro, em magnitude, é então definido como a distância do ponto em questão ao seu correspondente na linha contínua.

O cálculo dos erros nas referências de posição calculadas pelo sistema implementado leva, no entanto, a um problema adicional. Como o plano da trajetória de entrada não coincide com o espaço de trabalho do *plotter*, e como ela pode ser situada livremente naquele plano (ver figura 2.2), os eixos x e y do *plotter* não coincidem, em geral, com os eixos adotados na definição analítica da curva a ser rastreada. Este problema se evidencia, aliás, nas figuras 8.14, 8.15, 8.16 e 8.17, em que se observa que a trajetória efetiva do TCP do *plotter* não mantém as mesmas relações com as margens do papel que a trajetória de entrada. É por isso que, nas figuras 8.18, 8.19, 8.20 e 8.21, as curvas de entrada foram rotacionadas e transladadas de forma a se sobrepor às traçadas pelo TCP.

Assim, o primeiro passo na avaliação dos erros consiste em determinar a posição da trajetória de entrada em relação ao sistema de coordenadas do *plotter*, a partir dos pontos calculados pelo sistema de visão. Esta determinação é feita de forma a levar a um melhor ajuste entre os pontos calculados e o modelo analítico da trajetória. O correspondente na curva de entrada de cada ponto calculado pode então ser determinado, bastando que se verifique qual o ponto que dele mais se aproxima, no modelo determinado (o algoritmo de busca circular já garante que os pontos extraídos estão na seqüência correta, definida pela trajetória).

No caso da circunferência, o erro de posição de um ponto calculado em relação à trajetória de entrada é a diferença entre sua distância ao centro da circunferência e o raio (no caso, $60,5\text{mm}$). O primeiro passo é, portanto, estimar as coordenadas x e y do centro da circunferência, a partir das médias das coordenadas de todos os pontos situados em um número inteiro de voltas (3) ao longo do trajeto. Durante o cálculo do centro, assim como, posteriormente, durante o cálculo das distâncias entre este ponto e as posições calculadas, deve-se levar em conta a relação entre a variação nas coordenadas x e y definidas em termos das palavras de comando de 8 *bits* e a variação correspondente na posição do TCP, em milímetros (equações (A.1) e (A.2)).

Já para o retângulo, o primeiro passo na avaliação dos erros é calcular a equação das quatro retas que definem os lados da trajetória calculada pelo sistema de visão, o que equivale a determinar a posição do modelo adotado (retângulo) no sistema de coordenadas do *plotter*. Para tanto, os pontos calculados são inicialmente agrupados de acordo com o lado do retângulo a que pertencem; as equações de cada lado são então obtidas interpolando-se os pontos correspondentes por uma função polinomial de primeiro grau³.

Em seguida, determinam-se os vértices do retângulo, a partir das intersecções duas a duas das retas calculadas. Desta forma, o erro, em módulo, entre cada ponto calculado e

³Neste processo de interpolação, considera-se como variável independente aquela dentre x e y que apresenta valores distribuídos ao longo de um maior intervalo. Isto leva a um melhor condicionamento numérico durante a aplicação do método dos mínimos quadrados no cálculo do polinômio.

a trajetória de entrada pode ser estimado como a distância mínima entre aquele ponto e uma das quatro semi-retas calculadas, dentro dos limites estabelecidos pelos vértices.

As figuras 8.22 e 8.23 apresentam os erros assim avaliados para os 840 e os 1140 pontos nas trajetórias traçadas pelo *plotter* respectivamente para a circunferência e o retângulo de entrada. Observa-se, nestas figuras, que os erros ficam restritos a uma faixa de valores (de cerca de $-1.5mm$ a $1.5mm$ no caso da circunferência e de $-2.5mm$ a $2.5mm$ no caso do retângulo), ou seja, não há um efeito acumulativo do erro ao longo do rastreamento. Os valores médios dos erros foram de $0,0046mm$ e $0,0846mm$ e os desvios-padrões de $0,6521mm$ e $0,8685mm$, respectivamente para a circunferência e o retângulo; os erros raiz média quadrática foram de $0,6517mm$ e $0,8722mm$.

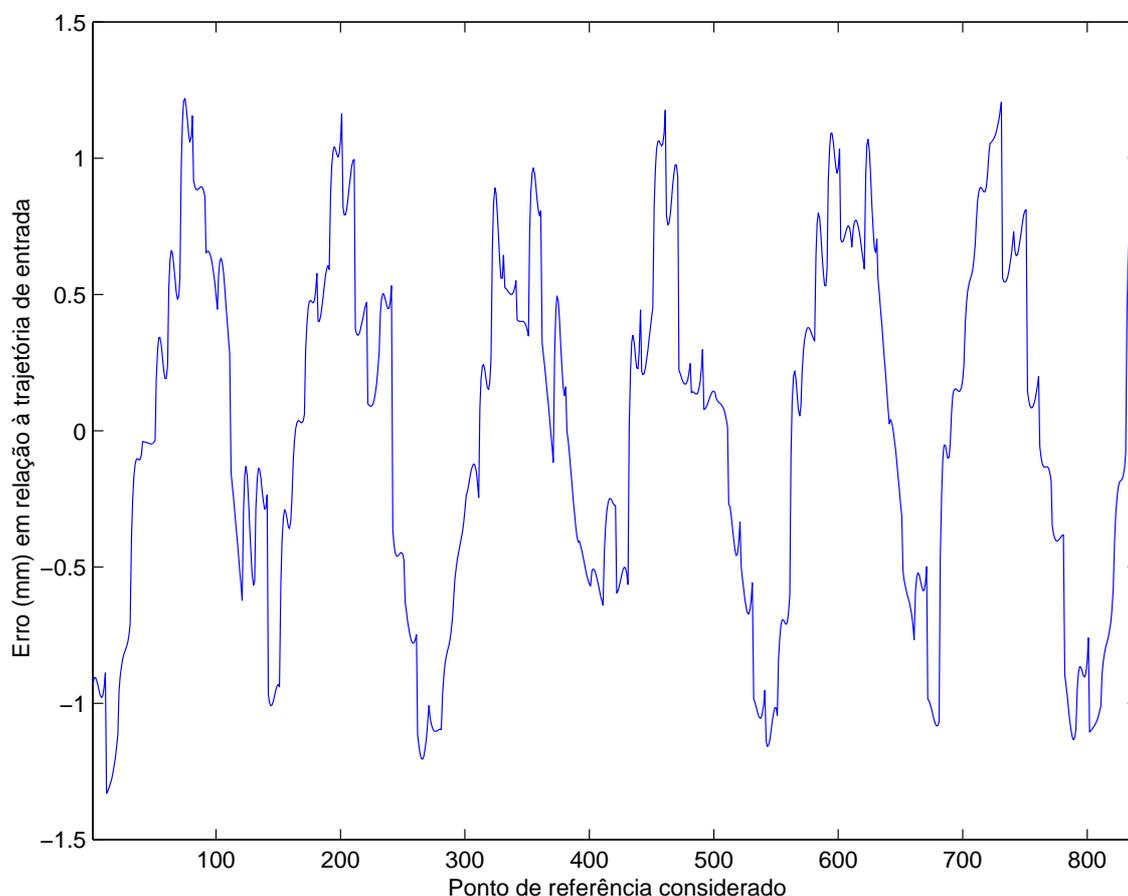


Figura 8.22: Erros avaliados entre as posições de referência calculadas pelo sistema de visão e a trajetória da figura 8.14-(a), durante o rastreamento. O valor médio destes erros é de $0,0046mm$, e o desvio-padrão, de $0,6521mm$. O erro raiz média quadrática é de $0,6517mm$.

Estes fatos também são verificados a partir da sobreposição das trajetórias efetivas do TCP com as trajetórias de entrada, nas figuras 8.18 e 8.21. Considerando-se a diferença entre a curva traçada pelo TCP e a de entrada, mais grossa, constata-se, com auxílio da escala milimetrada, um erro máximo não superior a cerca de $1.5mm$ no caso da circunferência e a $2,5mm$ no caso do retângulo (os erros negativos ocorrem quando o TCP se

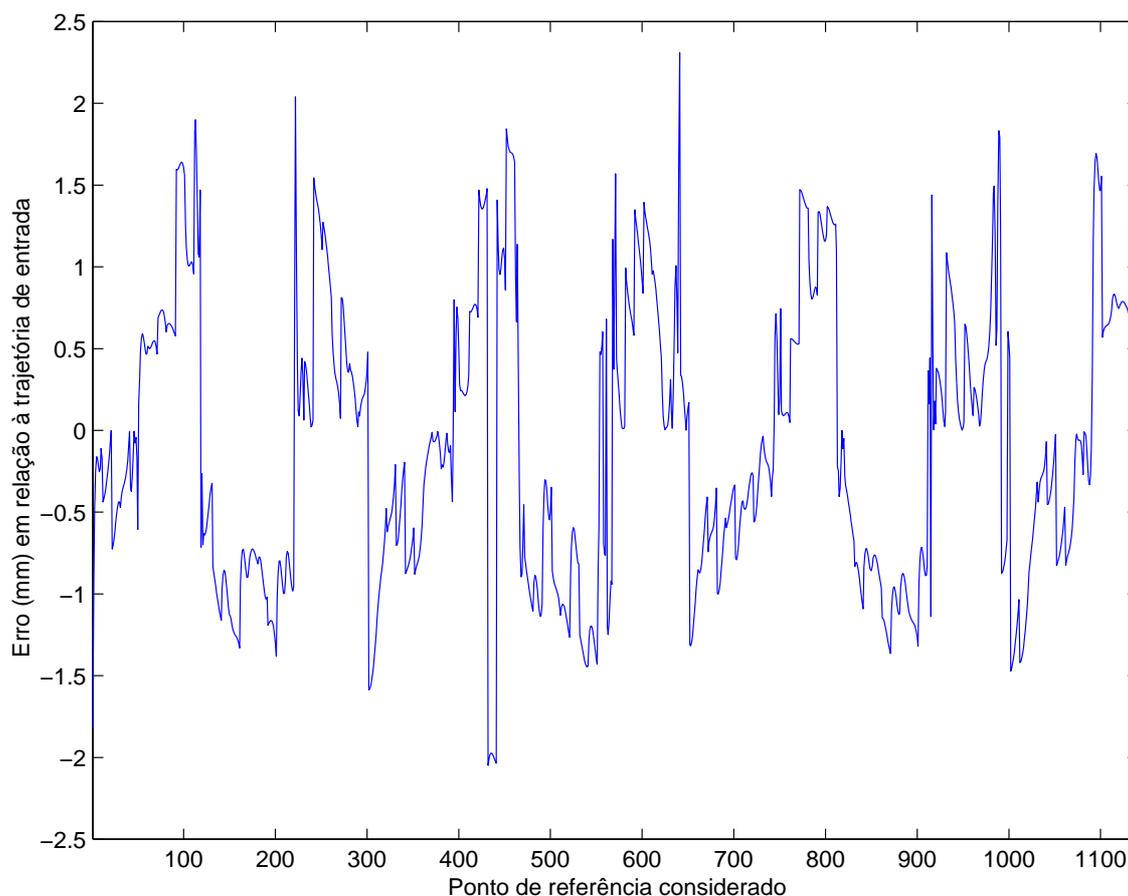


Figura 8.23: Erros avaliados entre as posições de referência calculadas pelo sistema de visão e a trajetória da figura 8.17-(a), durante o rastreamento. O valor médio destes erros é de $0,0846mm$, e o desvio-padrão, de $0,8685mm$. O erro raiz média quadrática é de $0,8722mm$.

afasta da trajetória de entrada em direção ao centro, aproximando-se do lado interno da linha, já que o rastreamento é feito pelo lado interno). Observa-se ainda que os resultados se repetem nas voltas sucessivas realizadas pelo TCP, mostrando que o erro não tende a aumentar ao longo do trajeto.

Resultados semelhantes são observados no caso da segunda e da terceira trajetórias, nas figuras 8.19 e 8.20. Na curva fechada da figura 8.19, o mesmo trajeto é seguido pelo TCP nas sucessivas voltas, e o erro fica assim restrito a uma faixa de valores não superiores a cerca de $2.5mm$. No caso da figura 8.21, o TCP acompanha a trajetória de entrada a partir do centro e em direção aos limites do espaço de trabalho, mas o erro não se amplia à medida que aumenta o raio da trajetória e seu valor máximo fica em torno de $1.5mm$.

A natureza dos erros medidos pode ser analisada tendo em vista a precisão do *plotter*. Conforme observado na página 125 do apêndice A, uma variação mínima da referência de entrada corresponde a um deslocamento $D_x = 0,8953mm$ no caso do eixo x e a $D_y = 0,5977mm$ no caso do eixo y , nas regiões em que o *plotter* permite uma melhor resolução espacial; nas regiões mais afastadas do centro, uma variação apenas do *bit* me-

nos significativo não faz deslocar o TCP, ou seja, o *plotter* passa a ser comandado com uma resolução de apenas 7 *bits*, sendo os deslocamentos mínimos nos eixos x e y respectivamente $2D_x = 1,7906mm$ e $2D_y = 1.1954mm$. Observe-se, além disso, que se a trajetória ideal a ser seguida apresenta um ponto exatamente a meio caminho entre duas posições possíveis do TCP, o erro observado, de metade do deslocamento mínimo, não pode ser corrigido, ainda que o sistema de visão permita detectar a diferença entre a imagem daquele ponto-alvo e o *pixel* de referência⁴. Assim, haverá erros da ordem de $\sqrt{D_x^2 + D_y^2} = 1.0765mm$ simplesmente pelo fato de não poder ser deslocado o TCP às posições ideais, mesmo sendo detectada a diferença entre estas posições e suas posições efetivas; além disso, mesmo erros superiores podem ser obtidos, devido a não-linearidades do *plotter* ou à calibração imperfeita do sistema (os erros máximos observados nas figuras 8.19 e 8.21 sugerem uma não-linearidade do *plotter*, já que se repetem nas 3 voltas na mesma região de seu espaço de trabalho mas não em outras regiões idênticas de cada trajetória).

No entanto, a detecção dos erros pelo sistema de visão e a possibilidade de corrigi-los quando ultrapassam o deslocamento mínimo do TCP explicam por que eles ficam situados entre um valor mínimo negativo e um valor máximo positivo (ver figuras 8.22 e 8.23). Quando a magnitude do erro ultrapassa o deslocamento mínimo do TCP, este pode ser mobilizado no sentido de reduzi-la; nos gráficos apresentados, observa-se, de fato, que quando o erro atinge seu valor máximo ou mínimo, ele é imediatamente reduzido em módulo. Neste momento, volta a se situar numa faixa de valores em que já não se garante mais sua correção, até que ele ultrapasse novamente o deslocamento mínimo do TCP e volte a ser imediatamente reduzido. Assim, o erro oscila em uma faixa de valores bem conhecida e não apresenta um efeito acumulativo durante o rastreamento.

Estas observações permitem concluir que o sistema implementado permitiu o rastreamento de trajetórias a partir apenas da informação extraída pelos algoritmos de visão das imagens adquiridas durante a movimentação do TCP. A precisão alcançada foi bastante aceitável diante da resolução do *plotter* empregado, com erros de posição que atingiram valores máximos da ordem de 2,5mm mas que, em geral, foram bem menores (nos dados ilustrados nas figuras 8.22 e 8.23, os erros raiz média quadrática são de 0,6517mm e 0,8722mm e os desvios-padrões de 0,6521mm e 0,8685mm, respectivamente). No estabelecimento desta precisão, os algoritmos de visão não foram os fatores limitantes, já que deslocamentos bastante inferiores — de menos de 0,15mm — são detectáveis por aqueles algoritmos.

⁴Com efeito, de acordo com a matriz de calibração obtida — equação (7.33) — erros de apenas cerca de 0,1441mm acarretam a diferença de um *pixel* na imagem adquirida, conforme mostrado na seção 7.1.2.

Capítulo 9

CONCLUSÕES

Neste trabalho, foi implementado um sistema de visão computacional para rastreamento de trajetórias por manipuladores robóticos. Ênfase foi dada aos algoritmos de visão propostos para a solução do problema, e não ao controle da dinâmica de manipuladores. Assim, as referências de entrada do sistema de posicionamento são determinadas exclusivamente a partir de imagens adquiridas da trajetória a ser rastreada, por uma câmera que se movimenta no espaço de trabalho fixa ao TCP.

Em uma etapa de pré-processamento, cada imagem de entrada é inicialmente compensada quanto à influência de condições não-ideais de iluminação. Uma contribuição deste trabalho é a proposta de um algoritmo para o cálculo automático de alguns parâmetros da imagem compensada, que permitem descrevê-la em termos de um modelo estatístico para uma imagem de um objeto sobre um fundo homogêneo quanto à reflectância. A partir destes parâmetros é então possível calcular um limiar apropriado à binarização da imagem, e conseqüentemente determinar quais *pixels* representam pontos da trajetória.

Em testes preliminares de desempenho, o algoritmo proposto para o cálculo automático dos parâmetros estatísticos foi aplicado na análise de diversas imagens de trajetórias, reais e sintéticas. Nestes casos, permitiu o cálculo com erros inferiores a 3% em relação aos parâmetros utilizados na geração das imagens; em todos os casos, permitiu determinar um limiar que separa corretamente as duas distribuições gaussianas que representam medidas da reflectância da trajetória e do fundo.

A etapa seguinte no rastreamento da trajetória diz respeito à definição de uma relação de ordem entre os *pixels* da imagem binarizada, de acordo com a seqüência em que os pontos correspondentes no espaço de trabalho do manipulador devem ser percorridos e por meio de um algoritmo designado busca circular. A seqüência de *pixels* então obtida permite determinar os deslocamentos necessários do TCP para que ele ocupe sucessivamente os pontos especificados, a partir de uma relação linear estabelecida durante a calibração do sistema óptico. Finalmente, as coordenadas dos pontos-alvos são ainda ajustadas em um processo de interpolação, cujo objetivo é garantir uma distância constante entre pontos consecutivos na nova seqüência gerada e, conseqüentemente, uma velocidade uniforme do

TCP ao ser enviado a intervalos regulares de tempo às posições especificadas. Só depois de percorridas estas posições é que uma nova imagem da trajetória é adquirida, dando início a um novo ciclo de processamento.

Para a avaliação de desempenho dos algoritmos implementados, em situações práticas de rastreamento, utilizou-se um *plotter* com dois graus de liberdade guiado por visão computacional. O sistema completo foi avaliado para diversas trajetórias de entrada, sendo então analisados os resultados de todos os processos intermediários do rastreamento e o desempenho global do sistema, em termos dos erros medidos entre as trajetórias efetivas do TCP e as trajetórias de entrada.

Os resultados observados mostraram que o sistema implementado permite o rastreamento de trajetórias a partir apenas da informação extraída pelos algoritmos de visão das imagens adquiridas durante a movimentação do TCP, como fora objetivado. A precisão alcançada foi bastante aceitável diante da resolução do *plotter* empregado, com erros de posição que atingiram valores máximos da ordem de $3mm$ mas que, em geral, foram bem menores, de acordo com a análise estatística dos erros obtidos para algumas trajetórias analíticas de entrada. No estabelecimento desta precisão, os algoritmos de visão não foram os fatores limitantes, já que deslocamentos bastante inferiores — de menos de $0,15mm$ — são detectáveis por aqueles algoritmos para o sistema óptico empregado, como se infere de sua matriz de calibração obtida.

Uma sugestão para pesquisas futuras e aprimoramento do sistema implementado é a utilização de sensores de posição do TCP, de forma que se possa efetuar seu controle em malha fechada mesmo enquanto ele é enviado às várias posições calculadas a partir de uma imagem, e não apenas durante a aquisição de imagens. É importante, no entanto, que se utilize um manipulador com precisão superior à do *plotter* aqui empregado, já que, neste caso, a precisão do sistema global já é bastante aceitável frente à resolução espacial do manipulador. Diversas técnicas de controle poderiam então ser empregadas com o intuito de minimizar o sinal de erro medido durante o rastreamento; o uso de imagens para geração das referências de entrada sugere, por exemplo, o emprego de técnicas de controle preditivo [3][29]: de cada imagem são extraídas as posições desejadas em vários instantes futuros, que podem então ser utilizados na definição do sinal atuante no instante atual para que se minimize o erro global do sistema em todos os instantes (com esta estratégia, é possível, por exemplo, eliminar o erro de arraste para uma rampa de entrada, observado no controle clássico).

Outra possibilidade, já que cada imagem adquirida permite avaliar o erro de posição atual e, assim, fechar a malha de controle, é reduzir ao máximo o número de posições que o TCP tem de percorrer antes que seja adquirida uma nova imagem, ou seja, aproximar ao máximo a taxa de aquisição de imagens da taxa de atualização das referências de entrada do sistema de posicionamento. Como existe um limite mínimo para esta taxa, de forma que não se varie consideravelmente a velocidade do TCP, trata-se de otimizar o tempo

total de execução dos algoritmos de visão aplicados a cada imagem e de maximizar, assim, a taxa efetiva de aquisição. Nesta pesquisa, conforme ressaltado, foi propositadamente adotado um número grande de posições de referência extraídas de cada imagem, para que se pudesse avaliar o desempenho de diferentes algoritmos de visão, e, assim, não houve preocupação em otimizar o tempo de execução. Mas algumas observações podem ser feitas com o intuito de reduzir o custo computacional no sistema proposto.

A mais evidente, no caso bem específico da câmera e dos algoritmos utilizados, é eliminar o processo de interpolação das posições de referência na fase final do ciclo de controle. Conforme analisado na seção 8.5, a igual resolução da câmera nos dois eixos do plano de imagem e a robustez dos algoritmos de visão, que garantiram a binarização sem falhas das imagens de entrada, levaram a uma distância praticamente constante entre posições de referência consecutivas, mesmo sem a aplicação da interpolação teoricamente necessária para se garantir esta condição.

Também durante o pré-processamento e a binarização das imagens pode haver uma redução do esforço computacional. Os parâmetros estatísticos das imagens utilizados na determinação do limiar de binarização não são avaliados para cada imagem adquirida, conforme já destacado, o que já representa um ganho em termos de processamento. Ainda assim, todas as imagens adquiridas são pré-processadas e binarizadas por completo, embora isto não seja de fato necessário. Durante a aplicação do algoritmo de busca circular, o que se observa é que cada *pixel* extraído da imagem é determinado a partir do *pixel* anterior, pela intersecção de uma circunferência centrada neste *pixel* com a imagem binarizada. Assim, para cada *pixel* a ser extraído apenas uma pequena região em torno do *pixel* anterior precisa ser binarizada; aplicando-se a compensação da não-uniformidade da iluminação, bem como o processo de limiarização propriamente dito, apenas a esta região, reduz-se significativamente a quantidade de operações envolvidas, embora a implementação fique um pouco mais complexa devido à necessidade de se definir a área a ser processada para cada *pixel* a ser extraído. Deve-se observar ainda que durante a determinação dos parâmetros estatísticos da imagem segundo o algoritmo proposto, no entanto, toda a imagem deve ser processada, conforme descrito no capítulo 4.

Finalmente, cabe mencionar mais uma vez uma importante restrição do sistema implementado, que corresponde a um importante problema em rastreamento de trajetórias e que justifica uma pesquisa específica para sua solução. Uma vez iniciado o rastreamento, ele segue de forma automática com base nos algoritmos de visão implementados; entretanto, não é abordado o problema da determinação do ponto de partida da trajetória, de forma que um operador humano deve informar este ponto, bem como a posição inicial do TCP no espaço de trabalho, já que não se dispõe de um sensor de posição. Esta condição é um importante empecilho ao total automatismo em tarefas de rastreamento baseadas em visão; um aprimoramento do sistema proposto seria portanto o desenvolvimento de um algoritmo que permitisse identificar, em uma varredura do espaço de trabalho, o ponto

inicial da trajetória; naturalmente, este problema é trivial no caso de trajetórias fechadas, já que o primeiro ponto encontrado na trajetória pode ser adotado como o inicial, mas para as demais trajetórias deve-se determiná-lo com base nas imagens adquiridas durante a varredura.

Referências Bibliográficas

- [1] Apostol, T. M. *Cálculo*, volume 2. Editora Reverté Ltda., Loreto, 13–15, Local B, 08029 Barcelona, 1994.
- [2] Apostol, T. M. *Cálculo*, volume 1. Editora Reverté Ltda., Loreto, 13–15, Local B, 08029 Barcelona, 1994.
- [3] Bauchspiess, A. *Prädiktive sensorgesteuerte Bahnführung von Handhabungssystemen*. Tese de Doutorado, Der Technischen Fakultät der Universität Erlangen-Nürnberg, Maio de 1995.
- [4] Bauchspiess, A., Alfaro, S. C. A., Trier, W., e Drews, P. Sistema de visão computacional inteligente para robôs de soldagem. *3^o Simpósio Brasileiro de Automação Inteligente*, pages 64–69, Setembro de 1997.
- [5] D’azzo, J. J. e Houpis, C. H. *Análise e Projeto de Sistemas de Controle Lineares*. Editora Guanabara, Travessa do Ouvidor, 11, Rio de Janeiro, RJ — CEP 20040, 1988.
- [6] Faugeras, O. *Three-Dimensional Computer Vision — A Geometric Viewpoint*. The MIT Press, Massachusetts, 1996.
- [7] Fujisawa, S., Yamamoto, T., e Suita, Y. Development of path tracking control for omni-directional mobile robot using visual servo system. *IECON’01: The 27th Annual Conference of the IEEE Industrial Electronics Society*, pages 2166–2170, 2001.
- [8] Gonzalez, R. C. e Woods, R. E. *Processamento de Imagens Digitais*. Editora Edgard Blücher LTDA, Rua Pedroso Alvarenga, 1245 – cj. 22, São Paulo – SP, Brasil, 2000.
- [9] Grewe, L. L. e Kak, A. C. Stereo vision. *Robot Vision Laboratory, School of Electrical Engineering, Purdue University, Indiana*, 1994.
- [10] Hager, G. D., Hutchinson, S., e Corke, P. Tutorial tt3: A tutorial on visual servo control. *IEEE Conference on Robotics and Automation*, 1994.
- [11] Helstrom, C. W. *Probability and Stochastic Processes for Engineers*. Macmillan Publishing Company, 866 Third Avenue, New York, New York 10022, 1984.
- [12] Horn, B. K. P. Determining lightness from an image. *Computer Graphics and Image Processing*, 3:277–299, Dezembro de 1974.
- [13] Horn, B. K. P. *Robot Vision*. The MIT Electrical Engineering and Computer Science Series. The MIT Press, 1998.
- [14] Land, E. H. The retinex. *American Scientist*, 52(2):247–264, Junho de 1964.
- [15] Land, E. Experiments in color vision. *Scientific American*, 200(5):84–99, Maio de 1959.
- [16] Marks, R. L., Lee, M. J., e Rock, S. M. Using visual sensing for control of an underwater robotic vehicle. (rlmarks@sun-valley.Stanford.Edu):213–224.

- [17] Mendonça, A. e Zelenovsky, R. *PC e Periféricos — Um Guia Completo de Programação*. Editora Ciência Moderna Ltda., 1996.
- [18] Miosso, C. J. e Bauchspiess, A. 3d photogrammetry of polyhedral structures using fuzzy border detection. *Intelligent Assembly and Disassembly — IAD2001, IFAC*, Novembro de 2001.
- [19] Miosso, C. J., Bauchspiess, A., e Moreira, P. L. Determinação automática dos parâmetros estatísticos de imagens de 2 regiões de reflectâncias constantes para cálculo do limiar Ótimo de binarização. *VSBAI — V Simpósio Brasileiro de Automação Inteligente*, 2001.
- [20] Morais, J. A. T. Controle de manipuladores utilizando visão. Dissertação de Mestrado, Universidade Federal do Rio de Janeiro, COPPE, Dezembro de 1998.
- [21] National Semiconductor, www.national.com. *DAC0800 / DAC0802 8-Bit Digital-to-Analog Converters*, Junho de 1999.
- [22] Nayar, S. K. e Bolle, R. M. Computing reflectance ratios from an image. *Pattern Recognition — <http://citeseer.nj.nec.com/nayar93computing.html>*, 26(10):1529–1542, 1993.
- [23] Ohyama, W., Wakabayashi, T., Kimura, F., Tsuruoka, S., e Sekioka, K. Automatic left ventricular endocardium detection in echocardiograms based on ternary thresholding method. *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR'00)*, 2000.
- [24] Oppenheim, A. V. e Schafer, R. W. *Discrete-Time Signal Processing*. Prentice Hall Signal Processing Series. Prentice Hall, Englewood Cliffs, New Jersey 07632, 1989.
- [25] Papoulis, A. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, 1965.
- [26] Parker, J. R. Gray level thresholding in badly illuminated images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):813–819, Agosto de 1991.
- [27] Philips Electronic Components and Materials Division. *TTL Logic Series*, Agosto de 1984.
- [28] Quénot, G. M., Coutelle, C., Sérot, J., e Zavidovique, B. Implementing image processing applications on a real-time architecture. *Computer Architectures for Machine Perception — New Orleans, Louisiana, USA*, Dezembro de 1993.
- [29] Soeterboek, R. *Predictive Control — A Unified Approach*. Prentice Hall International Series in Systems and Control Engineering. Prentice Hall International (UK) Limited, Campus 400, Maylands Avenue, Hemel Hempstead, Herts HP2 7EZ, 1992.
- [30] Strang, G. *Linear Algebra and Its Applications*. Harcourt Brace Jovanovich, Inc., Orlando, Florida 32887, 1988.
- [31] Strang, G. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Box 812060, Wellesley MA 02181 USA, 1993.

Apêndice A

APARATO EXPERIMENTAL: SISTEMA DE CONTROLE DE POSIÇÃO E DE AQUISIÇÃO DE IMAGENS

A.1 INTERFACE ENTRE A PORTA PARALELA DO PC E O *PLOTTER*

A.1.1 Circuito de Interface

O acionamento do *plotter* a partir da porta paralela do PC exige que se possa alterar isoladamente a tensão elétrica em cada um de seus canais x e y de entrada, fazendo-se uso dos 8 *bits* de saída de dados nela disponíveis (pinos 2 a 9 [17]). A divisão destes *bits* entre os dois canais, de forma que a cada canal esteja associado sempre um mesmo grupo específico de pinos, é inviável, já que 4 *bits* representam uma resolução demasiado baixa para o posicionamento do *plotter* (apenas 16 posições possíveis em cada direção).

É preciso, portanto, que os 8 *bits* sejam sempre utilizados para especificar cada nova coordenada x e y , o que permite uma resolução de 256 posições em cada direção. Esta abordagem cria três dificuldades concernentes à conexão entre a porta paralela e o *plotter*, devendo ser utilizado um circuito específico de interface que permita:

1. selecionar o canal do *plotter* a ser acionado (um único por vez);
2. manter o valor de entrada de um canal mesmo que o outro seja acionado, quando uma nova palavra de 8 *bits* será escrita na porta paralela;
3. converter a palavra de 8 *bits* associada a cada canal do *plotter* em uma tensão analógica, cujos valores mínimo e máximo estão associados às posições extremas do TCP ao longo do eixo correspondente.

A figura A.1 descreve resumidamente, em um diagrama de blocos, o circuito implementado segundo estas diretrizes, conforme explicado a seguir.

A segunda condição claramente implica a necessidade de se utilizarem 2 memórias de 8 *bits*, que devem manter as 2 palavras digitais que especificam as posições x e y do TCP do *plotter*. Foram utilizados com este fim dois registradores 74374, conforme mostra a figura A.1.

Porta Paralela do PC

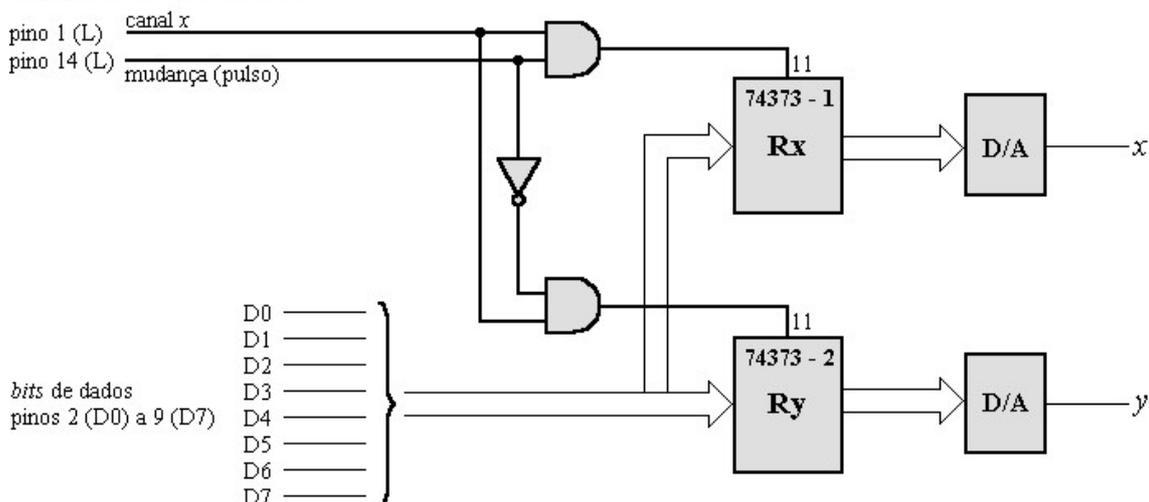


Figura A.1: Diagrama de blocos que descreve o circuito de interface entre a porta paralela do PC e o *plotter*. O atributo L indica que o pino correspondente atua com lógica invertida, ou seja, a escrita dos valores 0 e 1 leva a saída a 5V e a 0V, respectivamente.

As saídas destes dois registradores ficam permanentemente ligadas à entrada de dois conversores digital-analógico (DAC0800). Objetiva-se que aquelas saídas só se alterem quando deva ser mobilizado o TCP do *plotter*, ficando em geral armazenadas as posições atuais em palavras de 8 *bits* e em níveis analógicos, graças à conexão permanente aos conversores.

A alteração dos valores armazenados pelos 2 registradores se dá pela aplicação de uma borda de subida em uma entrada específica (gatilho), no caso dos registradores 74373 associada ao pino 11 [27]. Neste instante, são armazenados os 8 *bits* que definem a nova posição em uma das direções (x ou y). Para geração deste pulso, foi utilizado um dos pinos de controle (número 14) da porta paralela. Por se tratar de um terminal a coletor aberto, este pino pode ser utilizado diretamente como uma saída [17].

Um segundo *bit* de controle, associado ao pino 1, também a coletor aberto, foi ainda utilizado para selecionar o canal a ser alterado e, portanto, o registrador que deverá armazenar a palavra de 8 *bits* transmitida pelos pinos 2 a 9. Arbitrou-se que a seleção do canal x é feita aplicando-se o nível lógico alto no pino 1. Assim, este foi ligado juntamente com o pino 14 à entrada de uma porta E, cuja saída se associou ao pino 11 do registrador RX (canal x). Desta forma, a borda de subida gerada no pino 14 da porta paralela só é de fato transmitida a este registrador se o nível lógico em 1 for alto, garantindo que seu

conteúdo não seja alterado durante o acionamento do canal y . Por outro lado, à entrada 11 do registrador RY (canal y) ligou-se outra porta E, que recebe como entradas o *bit* 1 invertido e o *bit* 14 da porta paralela, de sorte que a seleção do canal y é feita pela aplicação do nível lógico baixo ao pino 1. Observe-se que um único canal do *plotter* será selecionado por vez, o que satisfaz a primeira condição exposta.

Deve-se observar que os pinos 1 e 14 da porta paralela atuam, no entanto, com lógica invertida, ou seja, a escrita do valor 0 em um deles resulta numa tensão de saída de $5V$, enquanto que o valor 1 o conduz ao potencial do terra [17]. Assim, a seleção do canal x é feita pela escrita do valor 0 no pino 1, e não do valor 1, como ocorreria se fosse utilizada lógica positiva. Da mesma forma, a borda de subida que comanda a transição de estado dos registradores é gerada escrevendo-se 1 e depois 0 no pino 14.

Resta apenas, agora, satisfazer a terceira condição. O TCP do *plotter* percorre uma área útil de $280mm \times 220mm$, e sua posição em cada eixo varia a uma taxa de $20mm/V$ em função da diferença de potencial elétrico aplicada no canal correspondente. Desta forma, para que seja utilizada toda a área útil do *plotter* e ao mesmo tempo se evite a saturação do dispositivo, como ocorreria se fosse aplicada em um dos canais uma tensão superior à necessária para deslocar o TCP ao ponto extremo, é necessário que os conversores digital-analógico associados aos canais x e y sejam configurados para fornecer as faixas de tensão de $-7V$ a $7V$ e de $-5,5V$ a $5,5V$, respectivamente. Estes devem ser os valores de saída dos conversores na situação em que os oito *bits* de entrada estão em nível lógico alto.

A figura A.2 ilustra a configuração do conversor DAC0800 para uma faixa de operação de $-12V$ a $12V$ ($24V_{pp}$) ([21]). Uma faixa distinta pode ser obtida pela modificação das resistências entre os pinos 2 e 4 e a fonte de $12V$. Este procedimento, cabe observar, leva a uma mudança nas correntes de saída I_s e \bar{I}_s , de forma que a relação entre a tensão máxima de saída e aquelas resistências não é uma simples proporção direta. Os valores apropriados tiveram de ser determinados, pois, empiricamente.

A figura A.3 mostra a configuração utilizada dos conversores associados aos canais x e y . No primeiro caso, a substituição dos resistores de $10k\Omega$ pelos de $2,7k\Omega$ permitiu que a saída operasse na faixa de $-5,63V$ a $5,83V$. No segundo, foram utilizados resistores de $1,7k\Omega$ e a saída foi feita variar de $-3,83V$ a $3,82V$. Observe-se que boa parte da área útil do *plotter* ($229,2mm \times 153,0mm$) ficou desta forma acessível ao TCP, o que permite uma boa resolução espacial sem que haja saturação do dispositivo.

Tendo em vista a distância máxima percorrida pelo TCP do *plotter* em cada eixo, podem ser determinados os deslocamentos D_x e D_y causados por uma variação unitária, em palavras de comando de 8 *bits*, nas coordenadas x e y , respectivamente. Como há 256 posições possíveis em cada eixo, obtêm-se:

$$D_x = \frac{229,2mm}{256} = 0,8953mm \text{ e} \quad (A.1)$$

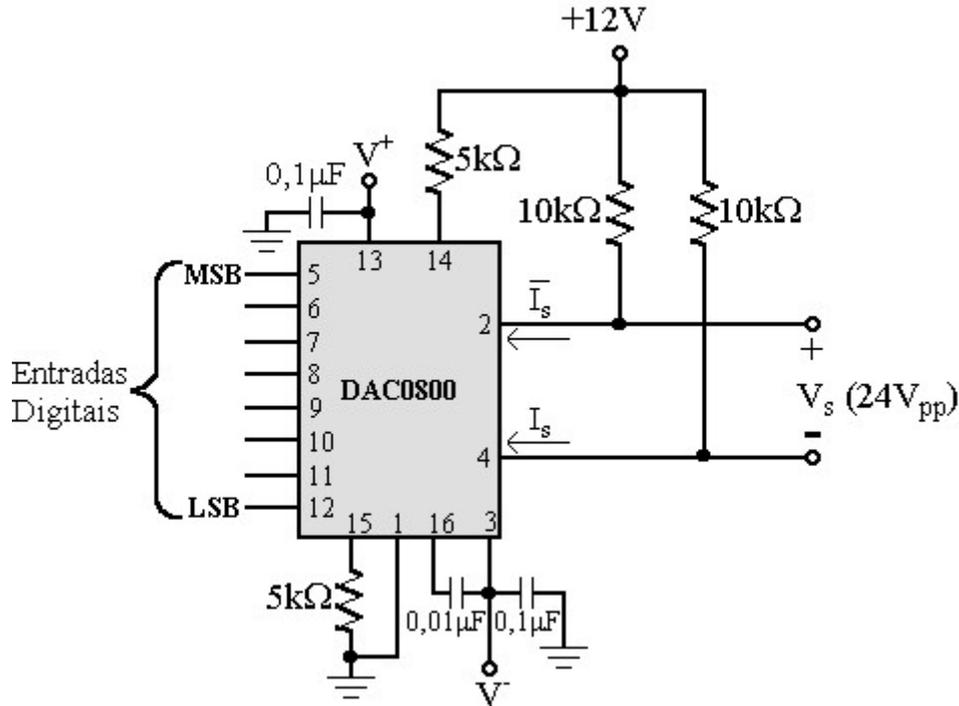


Figura A.2: Configuração do conversor DAC0800 para a faixa de operação de $-12V$ a $12V$ ($24V_{pp}$). MSB e LSB referem-se aos *bits* mais e menos significativos da palavra de entrada, respectivamente.

$$D_y = \frac{153,0mm}{256} = 0,5977mm. \quad (A.2)$$

A figura A.4 ilustra o circuito completo de interface entre o *plotter* e a porta paralela utilizado.

A.1.2 Rotinas de Interface

Para se alterar a posição do TCP do *plotter* com uso do circuito mostrado na figura A.4, não é bastante que se escreva a nova posição nos 8 pinos de dados da porta paralela. Importa ainda estabelecer no pino 1 o potencial elétrico que seleciona a coordenada a ser alterada, bem como gerar no 14 o pulso que comanda a transição de estado do registrador correspondente.

Os pinos 1 e 14 fazem parte de uma porta bidirecional associada ao endereço $37Ah$, conforme a tabela A.1. Os oito *bits* de dados, os pinos 2 a 9, se fazem acessíveis, por sua vez, em $378h$.

Uma rotina em *Assembly* bastante simples permite escrever nestes endereços de E/S os níveis lógicos apropriados que, em seqüência, ajustam a coordenada desejada. Assim,

Tabela A.1: Associação entre os pinos da porta paralela do PC e os endereços pelos quais eles são acessados.

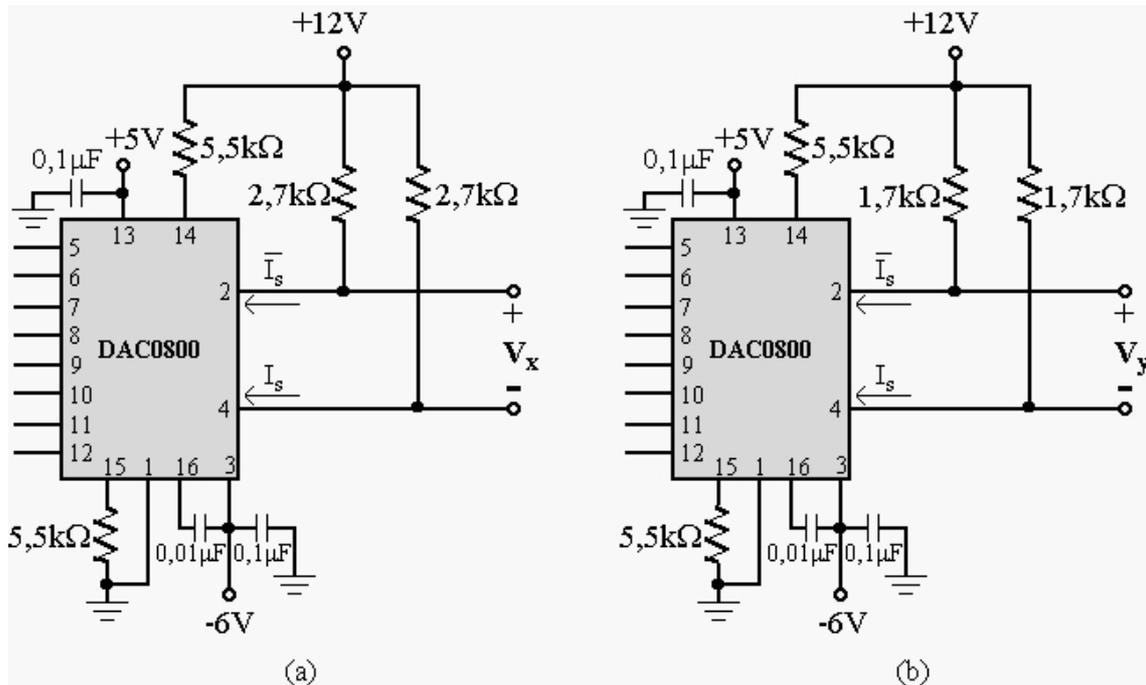


Figura A.3: Configuração utilizada dos conversores associados aos canais x (a) e y (b) do *plotter*, para operação nas faixas de $-5,63V$ a $5,63V$ e $-3,83V$ a $3,82V$, respectivamente.

para alterar a coordenada x , por exemplo, é necessário colocar o pino 1 em um potencial de $5V$. Só depois de se escrever a nova posição nos pinos 2 a 9 é que deve então ser gerado o pulso de subida no pino 14, garantindo a alteração da coordenada x sem que o estado do registrador RY seja alterado. Desta forma, a rotina em questão deve, em seqüência:

1. Escrever 0 no *bit* menos significativo no endereço $37Ah$, o que permite a alteração do conteúdo apenas do registrador RX;
2. Escrever 1 no *bit* imediatamente mais significativo no endereço $37Ah$, para que posteriormente possa ser gerado o pulso de subida que comandará a mudança de estado do registrador RX;
3. Escrever o valor da nova coordenada x no endereço $378h$;
4. Escrever 0 no segundo *bit* menos significativo no endereço $37Ah$, o que gera o pulso de subida que atualiza a coordenada armazenada no registrador RX.

Estas operações de escrita são realizadas com o uso de instruções *out*. Como os endereços utilizados são especificados com mais de 8 *bits*, não é possível empregar, no entanto, o endereçamento direto, mas sim o indireto. O endereço em questão é inicialmente transferido ao registrador dx , com uma instrução do tipo `mov dx,37Ah` ou `mov dx,378h`. A palavra p de saída é em seguida passada ao registrador al com uso da instrução `mov al,p`, para só então se realizar a operação de escrita propriamente dita, com `out dx,al`.

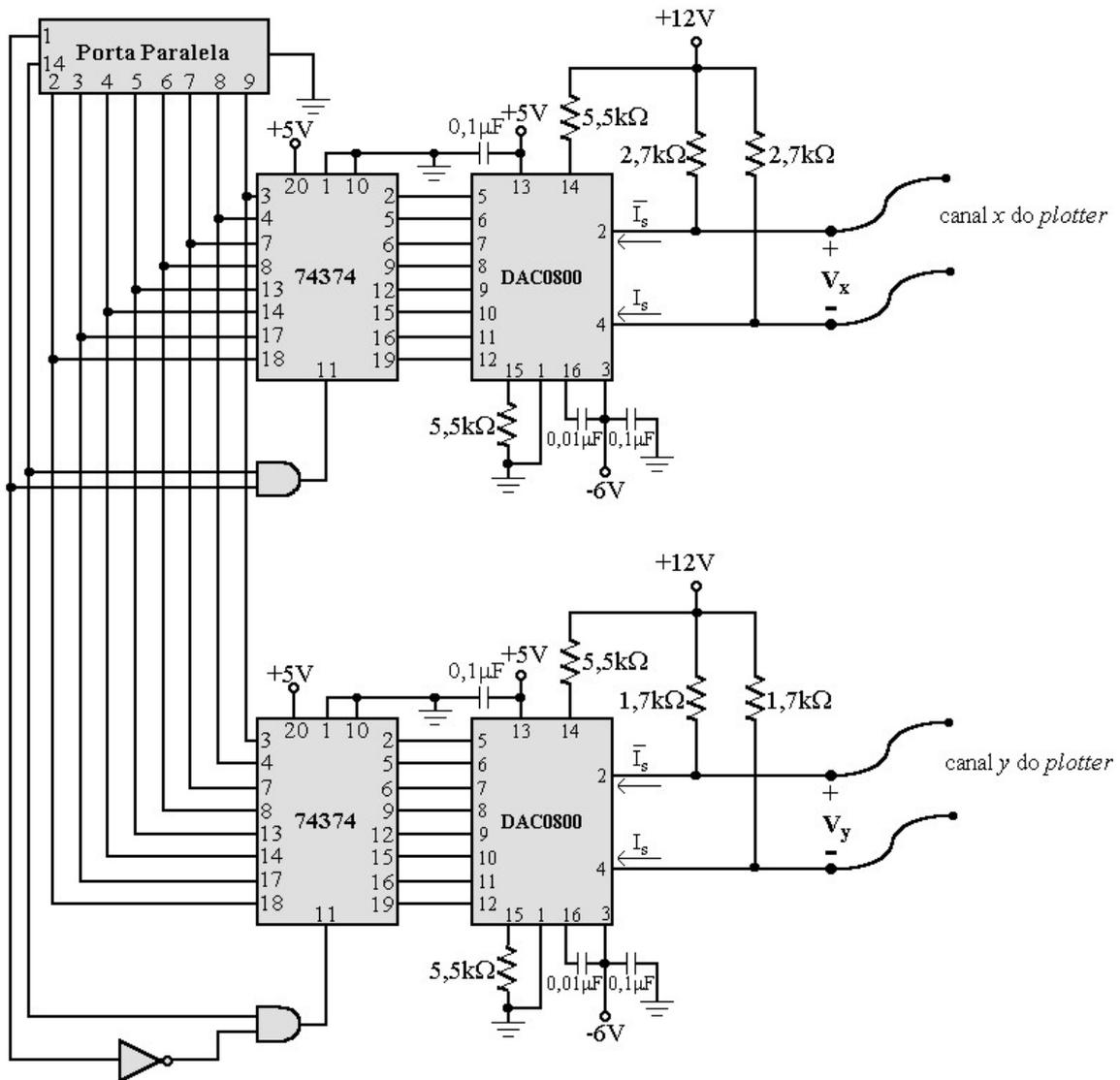


Figura A.4: Circuito completo de interface entre a porta paralela e o *plotter*.

Outro ponto a ressaltar é que se deve garantir que os níveis lógicos alto e baixo na borda de subida gerada tenham duração suficiente para que se opere a mudança de estado do registrador. Isto não ocorre se todas as instruções de escrita associadas às operações 1, 2, 3 e 4 forem executadas em seguida, sem pausas.

Para garantir a duração mínima do pulso, foi utilizada, por conseguinte, uma estrutura de repetição sem operações, imediatamente antes e após a escrita do valor 0 (nível lógico alto) no pino 14, garantindo a duração mínima dos níveis lógicos baixo e alto, respectivamente.

Com o objetivo de facilitar o interfaceamento com outros programas, em especial com o MatLab, como será descrito adiante, as intruções em *Assembly* foram incorporadas a uma função em linguagem C. A rotina de alteração da coordenada x ficou estruturada conforme descrito a seguir.

Como somente a coordenada x deve ser alterada por esta rotina, o pino 1 deve permanecer sempre em um potencial de $5V$, enquanto que o pino 14 deverá estar ora em $5V$, ora no potencial de terra. Duas variáveis de 8 *bits* (tipo `unsigned char`) foram criadas para descrever as duas posições possíveis:

```
unsigned char pino14baixo_pino1alto=2; /* 0010 em base 2 */
unsigned char pino14alto_pino1alto=0; /* 0000b em base 2 */

/* As saídas 1 e 14 são invertidas */
```

Em seguida, são realizadas as operações de escrita 1 e 2. Como os pinos 1 e 14 fazem parte de uma palavra associada a um único endereço, apenas uma instrução `out` é realizada. O pino 1 é colocado em $5V$ e o pino 14, no potencial de terra, para que posteriormente possa ser gerado o pulso de subida:

```
asm mov al,pino14baixo_pino1alto
asm mov dx,37Ah /* endereço das saídas de controle, incluindo os pinos
                1 e 14 */
asm out dx,al
```

A nova coordenada x é então escrita nos 8 pinos de dados (operação 3):

```
asm mov dx,378h /* endereço das 8 saídas de dados (pinos 2 a 9) */
asm mov al,coordx
asm out dx,al
espera(); /* estrutura de repetição sem operação, cujo tempo total de exe-
          cução foi ajustado para garantir que os níveis lógicos associados
          ao pino 14 tenham duração suficiente para a transferência da nova
          coordenada ao registrador apropriado */
```

Finalmente, atribui-se o nível lógico alto ao pino 14, o que gera o pulso de subida que comanda a transferência da palavra de 8 *bits* para o registrador `RX`:

```
asm mov dx,37Ah
asm mov al,pino14alto_pino1alto
asm out dx,al
espera();
```

A função `ajusta_coordenada_x`, como foi designada, é apresentada no apêndice B, à página 135. A rotina de alteração da coordenada y , `ajusta_coordenada_y`, é análoga, diferindo apenas pelo fato de que o pino 1 é mantido no potencial de terra, e é apresentada

na página 138.

A.2 ROTINAS DE AQUISIÇÃO DAS IMAGENS DE TRAJETÓRIAS

A.2.1 Descrição das Rotinas de Aquisição de Imagens Implementadas

Na implementação das rotinas de aquisição das imagens das trajetórias, fez-se uso de funções específicas da API¹ do Windows, presentes na biblioteca *Video for Windows*. Estas funções permitem a criação de uma janela e sua associação a um *driver* de vídeo, bem como a aquisição e o armazenamento em arquivos de imagens isoladas.

Duas rotinas principais foram implementadas. A primeira, `imcap.c`, em linguagem C, realiza a aquisição de imagens sucessivas à taxa máxima permitida pela câmera, pelo microcomputador e pelas funções da API utilizados, bem como o armazenamento destas imagens em um mesmo arquivo. Isto permite que o MatLab tenha acesso sempre à imagem mais recente, por meio da leitura do arquivo no momento em que ela se fizer necessária utilizando-se a segunda rotina — `imcap.m`. Esta rotina foi desenvolvida de forma a garantir que, durante uma leitura, tenha sido concluída a escrita do arquivo por `imcap.c`, ou seja, a leitura não deve ocorrer durante uma operação de escrita.

Na rotina `imcap.c`, o primeiro passo realizado é a criação de uma janela a ser posteriormente associada ao *driver* de aquisição de vídeo. Para tanto, empregou-se a função `capCreateCaptureWindow`:

```
hwnd=capCreateCaptureWindow(  
    title, // nome  
    WS_CLIPSIBLINGS | WS_CLIPCHILDREN, // estilo  
    x, y, width, height, // posição e tamanho  
    NULL, 0);
```

A variável de saída `hwnd`, do tipo `HWND`, é um índice (*handle*) para a janela recém-criada. Os parâmetros `title`, `x`, `y`, `width` e `height` são previamente definidos ao longo do programa e especificam uma série de propriedades da janela criada:

- `title`, um ponteiro para uma cadeia de caracteres, define o título presente na barra superior da janela (adotou-se “Sistema de Seguimento de Trajetórias Baseado em Visão Computacional”);

¹*Application Programming Interface* — Interface de Programação de Aplicativos

- `x` e `y`, do tipo `int`, definem a posição da janela (adotaram-se as coordenadas $x = 0$ e $y = 0$);
- `width` e `height`, do tipo `int`, especificam respectivamente a largura e a altura da janela, em número de *pixels* (adotaram-se os valores `width=640` e `height=480`, correspondentes às maiores dimensões de imagens possibilitadas pela câmera utilizada).

A janela criada deve então ser exibida, antes que se possa efetuar sua associação a um *driver* de aquisição de vídeo. Utilizou-se então a função `ShowWindow`, que recebe como entrada o índice `hwnd`:

```
ShowWindow(hwnd,SW_SHOW);
```

A conexão ao *driver* de aquisição da câmera foi então realizada pela função `capDriverConnect`:

```
conecta_driver=capDriverConnect(hwnd,i);
```

A variável de saída `conecta_driver`, do tipo `int`, apenas indica se ocorreu ou não a conexão, e assume os valores 1 ou 0, respectivamente. Em situações normais, com a câmera e seus programas corretamente instalados, será obtido portanto sempre o valor `conecta_driver=1`.

Finalmente, foi implementado então um ciclo de repetições em que uma imagem é adquirida e armazenada em arquivo *bitmap* independente de dispositivo (DIB — *Device Independent Bitmap*). O ciclo é executado enquanto o usuário não pressionar uma tecla com o programa de aquisição em primeiro plano dentre as tarefas executadas pelo sistema:

```
while(!kbhit())
{
    capGrabFrame(hwnd); // Captura de uma imagem.
    capFileSaveDIB(hwnd,arquivo_destino); // Armazenamento da imagem.
}
```

As funções `capGrabFrame` e `capFileSaveDIB` executam respectivamente a aquisição de uma imagem e sua gravação no arquivo especificado pelo ponteiro `arquivo_destino`. Como não é executada função alguma entre dois ciclos consecutivos de aquisição e escrita, a estrutura implementada leva à taxa máxima de aquisição permitida pela câmera, pelo microcomputador e pelas funções da API utilizados.

Encerrado o ciclo por comando do usuário, basta então que seja desfeita a conexão entre a janela `hwnd` e o *driver* de aquisição. Para tanto, utilizou-se a função `capDriverDisconnect`:

```
capDriverDisconnect(hwnd);
```

Já a rotina `imcap.m`, executada pelo MatLab, tem por objetivo realizar a leitura da última imagem adquirida. O problema central relacionado consiste em garantir que sua escrita no arquivo DIB tenha sido concluída, ou seja, a leitura não deve ocorrer durante a gravação da imagem.

A solução a este problema é simples, tendo-se em vista que, no MatLab, uma operação de leitura de imagem sem sucesso (por exemplo, de um arquivo aberto para escrita) não altera a variável de saída. Assim, definindo-se previamente uma matriz `I` vazia, pode-se garantir que a última imagem adquirida por `imcap.c` tenha sido armazenada no arquivo DIB antes da leitura, repetindo-se a chamada à função `I=imread(arquivo_captura)` até que as dimensões de `I` não sejam mais nulas. O ciclo de leitura ficou portanto assim esquematizado:

```
I=[];  
while(size(I,1)==0)  
    I=imread(arquivo_captura);  
end
```

Deve-se observar, no entanto, que, mesmo que a imagem adquirida não seja colorida, mas em níveis de cinza, `imread` pode retornar três planos (iguais) à variável `I`, como em uma imagem RGB. Neste caso, extrai-se o primeiro plano:

```
if tipo_imagem=='cinza'  
    I=I(:,:,1);  
end;
```

A seção B.2 do apêndice B apresenta as listagens completas das rotinas `imcap.c` e `imcap.m`.

A.2.2 Determinação da Taxa de Aquisição de Imagens Obtida

Dois fatores influenciam a taxa máxima de aquisição de imagens que pode ser obtida com uso das rotinas descritas na seção A.2.1. De um lado, ela é limitada pelo número de chamadas da função `imcap.m` que o MatLab permite realizar por unidade de tempo, para a configuração do microcomputador utilizado. De outro lado, há as restrições impostas pela câmera, pela taxa de transferência de dados pela porta USB e pelas características das bibliotecas de funções da API do Windows utilizadas. Na determinação da taxa de aquisição resultante da integração dos dois processos correspondentes, um associado à

rotina `imcap.c`, outro à `imcap.m`, deve-se considerar a frequência com que a operação de leitura da imagem ocorrerá antes de concluída sua escrita no arquivo DIB, caso em que ela terá de ser repetida até o fim do armazenamento e o sucesso da operação.

A taxa de aquisição resultante foi determinada por meio da realização de um número específico N de chamadas da função `imcap.m` pelo MatLab, em um ciclo de repetições, com a execução em paralelo de `imcap.c`. Com o auxílio das funções `tic` (*time of input command* — comando de tempo de entrada) e `toc` (*time of output command* — comando de tempo de saída), foi avaliado o tempo t de execução deste ciclo, em milissegundos. A razão entre o número total de imagens lidas, N , e o tempo total de leitura em segundos, $t/1000$, forneceu então a taxa de aquisição, em número de quadros por segundo.

A seqüência de comandos para cálculo da taxa de aquisição ficou, portanto, assim organizada:

```
tic; %zera o cronômetro, para medição do tempo total de leitura, t.
for k=1:N
    I=imcap(arquivo_captura);
end
t=toc; %tempo total de leitura, em milissegundos.
taxa=N*1000/t; %taxa de aquisição, em quadros por segundo.
```

Este procedimento forneceu uma taxa de aquisição máxima da ordem de 3 quadros por segundo, na resolução máxima permitida pela câmera.


```

unsigned char pino14baixo_pino1alto=2; /* 0010 em base 2 */
unsigned char pino14alto_pino1alto=0; /* 0000b em base 2 */
                                     /* As saídas 1 e 14 são invertidas */

/* Inicialmente, seleciona-se a coordenada x (pino 1 em nível alto) e atribui-se o nível lógico baixo ao pino 14, para que posteriormente possa ser gerado o pulso de subida. */

asm mov al,pino14baixo_pino1alto
asm mov dx,37Ah /* endereço das saídas de controle, incluindo os pinos
                1 e 14 */
asm out dx,al

/* Em seguida, escreve-se a nova coordenada x nos 8 pinos de dados (2 a 9). */

asm mov dx,378h /* endereço das 8 saídas de dados (pinos 2 a 9) */
asm mov al,coordx
asm out dx,al
espera(); /* estrutura de repetição sem operação, cujo tempo total de execução foi ajustado para garantir que os níveis lógicos associados ao pino 14 tenham duração suficiente para a transferência da nova coordenada ao registrador apropriado */

/* Finalmente, atribui-se o nível lógico alto ao pino 14, o que gera o pulso de subida que comanda a transferência da palavra de 8 bits para o registrador X. */

asm mov dx,37Ah
asm mov al,pino14alto_pino1alto
asm out dx,al
espera();
}

```

Rotina em C para Compilação em uma DLL Acessível ao MatLab — pos_x.c

```

/*****
* Programa para posicionamento da ponta do "plotter" em relação ao eixo *
* x a partir do MatLab. *
* *
* Utiliza-se a rotina ajusta_coordenada_x em "ajusta_x.c". *
* *

```

```

*   Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser *
* ajustada (x para nível lógico alto, y para nível lógico baixo), enquanto *
* que os pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 *
* bits. O pino 14, finalmente, é utilizado para a borda de subida que *
* comanda a transferência da palavra de 8 bits para o registrador apropriado *
* (X ou Y).
*
*
*                               Última atualização: 07/01/2001. *
*****/

```

```

#pragma inline /* Uso de instruções em Assembler em "ajusta_x.c" */
#include "espera.c" /* Rotina utilizada em "ajusta_x.c" */
#include "ajusta_x.c"
#include <mex.h> /* Para execução a partir do MatLab. */

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
/* nlhs: número de parâmetros de saída da função MatLab;
   plhs: ponteiro para cada matriz de saída da função;
   nrhs: número de parâmetros de entrada da função MatLab;
   prhs: ponteiro para cada matriz de entrada da função;
*/

{
  int m,n,coordx; /* m,n: dimensões da matriz de entrada (devem ser ambas 1,
                  /* já que esta corresponde à nova coordenada x - coordx) */
  double x; /* recebe a nova coordenada x, sendo verificado se foi fornecido
             um valor inteiro, conforme esperado */

  m=mxGetM(prhs[0]); /* número de linhas da matriz de entrada */
  n=mxGetN(prhs[0]); /* número de colunas da matriz de entrada */

  if(m*n!=1) /* verificação de que a entrada fornecida é um escalar */
    mexErrMsgTxt("coordx deve ser um escalar.");
  else
  {
    x=*mxGetPr(prhs[0]); /* leitura do valor fornecido */
    coordx=x;
    if((coordx!=x)||((coordx<0)||((coordx>255))) /* verifica se foi fornecido
                                                um inteiro entre 0 e 255*/
        mexErrMsgTxt("coordx deve ser um inteiro entre 0 e 255.");
    else
      ajusta_coordenada_x(coordx);
  }
}

```

Rotina em MatLab — pos_x.m

```
function pos_x(coordx);

% pos_x(coordx);
%
% Posiciona a ponta do "plotter" na posição (coordx,coordy), onde coordy é a
% coordenada y atual. Em outras palavras, somente a coordenada x é alterada.
%
% coordx deve ser um inteiro entre 0 e 255.
%
% Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser ajus-
% tada (x para nível lógico alto, y para nível lógico baixo), enquanto que os
% pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 bits. O
% pino 14, finalmente, é utilizado para a borda de subida que comanda a trans-
% ferência da palavra de 8 bits para o registrador apropriado (X ou Y).

% Função implementada na DLL pos_x.dll.
% Última atualização: 06/10/2001.
```

B.1.2 Rotina de Ajuste da Coordenada *y* do *Plotter*

Rotina em C com Acesso a *Hardware* por Instruções em *Assembly* da INTEL
— ajusta_y.c

```
/*
 * Rotina para posicionamento da ponta do "plotter" em relação ao eixo y.
 *
 * Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser
 * ajustada (x para nível lógico alto, y para nível lógico baixo), enquanto
 * que os pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8
 * bits. O pino 14, finalmente, é utilizado para a borda de subida que
 * comanda a transferência da palavra de 8 bits para o registrador apropriado
 * (X ou Y).
 *
 * Observação: no preâmbulo, antes da inclusão de "ajusta_x.c", deve ser
 * incluído também "espera.c". Deve ainda ser utilizada a diretiva
 * #pragma inline, devido à utilização de instruções em Assembler.
 *
 * Última atualização: 07/01/2001.
 */
void ajusta_coordenada_y(unsigned char coordy)
{
```

```

/* Somente a coordenada y deve ser alterada; assim, o pino 1 apresenta sempre
o nível lógico baixo, enquanto que em 14 deve ser gerado um pulso de subida.*/

unsigned char pino14baixo_pino1baixo=3; /* 0011 em base 2*/
unsigned char pino14alto_pino1baixo=1; /* 0001 em base 2*/
/* As saídas 1 e 14 são invertidas */

/* Inicialmente, seleciona-se a coordenada y (pino 1 em nível baixo) e atribui-se
o nível lógico baixo ao pino 14, para que posteriormente possa ser gerado o pulso de subida. */

asm mov al,pino14baixo_pino1baixo
asm mov dx,37Ah /* endereço das saídas de controle, incluindo os pinos
1 e 14 */
asm out dx,al

/* Em seguida, escreve-se a nova coordenada y nos 8 pinos de dados (2 a 9). */

asm mov dx,378h /* endereço das 8 saídas de dados (pinos 2 a 9) */
asm mov al,coordy
asm out dx,al
espera(); /* estrutura de repetição sem operação, cujo tempo total de execução
foi ajustado para garantir que os níveis lógicos associados
ao pino 14 tenham duração suficiente para a transferência da nova
coordenada ao registrador apropriado */

/* Finalmente, atribui-se o nível lógico alto ao pino 14, o que gera o pulso
de subida que comanda a transferência da palavra de 8 bits para o registrador
Y. */

asm mov dx,37Ah
asm mov al,pino14alto_pino1baixo
asm out dx,al
espera();
}

```

Rotina em C para Compilação em uma DLL Acessível ao MatLab — pos_y.c

```

/*****
* Programa para posicionamento da ponta do "plotter" em relação ao eixo *
* y a partir do MatLab. *
* *

```

```

*   Utiliza-se a rotina ajusta_coordenada_y em "ajusta_y.c".           *
*                                                                 *
*   Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser *
* ajustada (x para nível lógico alto, y para nível lógico baixo), enquanto *
* que os pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 *
* bits. O pino 14, finalmente, é utilizado para a borda de subida que *
* comanda a transferência da palavra de 8 bits para o registrador apropriado *
* (X ou Y).                                                         *
*                                                                 *
*                                                                 *
*                               Última atualização: 07/01/2001. *
*****
#pragma inline /* Uso de instruções em Assembler em "ajusta_y.c" */
#include "espera.c" /* Rotina utilizada em "ajusta_y.c" */
#include "ajusta_y.c"
#include <mex.h> /* Para execução a partir do MatLab. */

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
/* nlhs: número de parâmetros de saída da função MatLab;
   plhs: ponteiro para cada matriz de saída da função;
   nrhs: número de parâmetros de entrada da função MatLab;
   prhs: ponteiro para cada matriz de entrada da função;
*/

{
  int m,n,coordy; /* m,n: dimensões da matriz de entrada (devem ser ambas 1,
                  /* já que esta corresponde à nova coordenada y - coordy) */
  double y; /* recebe a nova coordenada y, sendo verificado se foi fornecido
             um valor inteiro, conforme esperado */

  m=mxGetM(prhs[0]); /* número de linhas da matriz de entrada */
  n=mxGetN(prhs[0]); /* número de colunas da matriz de entrada */

  if(m*n!=1) /* verificação de que a entrada fornecida é um escalar */
    mexErrMsgTxt("coordy deve ser um escalar.");
  else
  {
    y=*mxGetPr(prhs[0]); /* leitura do valor fornecido */
    coordy=y;
    if((coordy!=y)||((coordy<0)||((coordy>255))) /* verifica se foi fornecido
                                                um inteiro entre 0 e 255*/
        mexErrMsgTxt("coordy deve ser um inteiro entre 0 e 255.");
    else
      ajusta_coordenada_y(coordy);
  }
}

```

```
}
```

Rotina em MatLab — pos_y.m

```
function pos_y(coordy);

% pos_y(coordy);
%
% Posiciona a ponta do "plotter" na posição (coordx,coordy), onde coordx é a
% coordenada x atual. Em outras palavras, somente a coordenada y é alterada.
%
% coordy deve ser um inteiro entre 0 e 255.
%
% Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser ajus-
% tada (x para nível lógico alto, y para nível lógico baixo), enquanto que os
% pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 bits. O
% pino 14, finalmente, é utilizado para a borda de subida que comanda a trans-
% ferência da palavra de 8 bits para o registrador apropriado (X ou Y).

% Função implementada na DLL pos_y.dll.
% Última atualização: 06/10/2001.
```

B.1.3 Rotina de Ajuste Simultâneo das Coordenadas x e y do *Plotter*

Rotina em C para Compilação em uma DLL Acessível ao MatLab — pos.c

```
/*
*****
* Programa para posicionamento da ponta do "plotter" em relação aos eixos *
* x e y a partir do MatLab. *
* *
* Utilizam-se as rotinas ajusta_coordenada_x e ajusta_coordenada_y, res- *
* pectivamente em "ajusta_x.c" e "ajusta_y.c". *
* *
* Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser *
* ajustada (x para nível lógico alto, y para nível lógico baixo), enquanto *
* que os pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 *
* bits. O pino 14, finalmente, é utilizado para a borda de subida que *
* comanda a transferência da palavra de 8 bits para o registrador apropriado *
* (X ou Y). *
* *
* Última atualização: 07/01/2001. *
*****
*/

#pragma inline /* Instruções em Assembler em "ajusta_x.c" e "ajusta_y.c" */
```

```

#include "espera.c" /* Rotina utilizada em "ajusta_x.c" e "ajusta_y.c" */
#include "ajusta_x.c"
#include "ajusta_y.c"
#include <mex.h> /* Para execução a partir do MatLab. */

void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
/* nlhs: número de parâmetros de saída da função MatLab;
   plhs: ponteiro para cada matriz de saída da função;
   nrhs: número de parâmetros de entrada da função MatLab;
   prhs: ponteiro para cada matriz de entrada da função;
*/

{
    int mx,nx,my,ny,coordx,coordy; /* mx,nx: dimensões da primeira matriz de
                                   entrada (devem ser ambas 1, já que esta cor-
                                   responde à nova coordenada x - coordx) */
                                   /* my,ny: dimensões da segunda matriz de
                                   entrada (devem ser ambas 1, já que esta cor-
                                   responde à nova coordenada y - coordy) */

    double x,y; /*recebem as novas coordenadas x e y, sendo verificado se foram
                 fornecidos valores inteiros, conforme esperado */

    mx=mxGetM(prhs[0]); /* número de linhas da primeira matriz de entrada */
    nx=mxGetN(prhs[0]); /* número de colunas da primeira matriz de entrada */
    my=mxGetM(prhs[1]); /* número de linhas da segunda matriz de entrada */
    ny=mxGetN(prhs[1]); /* número de colunas da segunda matriz de entrada */

    if(mx*nx*my*ny!=1) /* verifica se as entradas fornecidas são escalares */
        mexErrMsgTxt("coordx e coordy devem ser escalares.");
    else
    {
        x=*mxGetPr(prhs[0]); /* leitura do primeiro valor fornecido */
        coordx=x;
        y=*mxGetPr(prhs[1]); /* leitura do segundo valor fornecido */
        coordy=y;
        if((coordx!=x)|| (coordx<0)|| (coordx>255)|| (coordy!=y)|| (coordy<0)||
           (coordy>255))
            /* verifica se foram fornecidos dois inteiros entre 0 e 255*/
            mexErrMsgTxt("coordx e coordy devem ser inteiros entre 0 e 255.");
        else
        {
            ajusta_coordenada_x(coordx);
            ajusta_coordenada_y(coordy);
        }
    }
}

```

```
}  
}
```

Rotina em MatLab — pos.m

```
function pos(coordx,coordy);  
  
% pos(coordx,coordy);  
%  
% Posiciona a ponta do "plotter" na posição (coordx,coordy).  
%  
% coordx e coordy devem ser inteiros entre 0 e 255.  
%  
% Admite-se que o pino 1 da porta paralela seleciona a coordenada a ser ajustada (x para nível lógico alto, y para nível lógico baixo), enquanto que os pinos 2 a 9, de dados, comunicam a nova posição, com resolução de 8 bits. O pino 14, finalmente, é utilizado para a borda de subida que comanda a transferência da palavra de 8 bits para o registrador apropriado (X ou Y).  
  
% Função implementada na DLL pos.dll.  
% Última atualização: 06/10/2001.
```

B.2 ROTINAS DE AQUISIÇÃO DE IMAGENS

B.2.1 Rotina de Captura de Imagens Sucessivas em um Mesmo Arquivo Utilizando a API do Windows — imcap.c

```
/*  
* Rotina para aquisição de imagens utilizando a API do Windows. *  
* *  
* As imagens são capturadas em forma seqüencial e armazenadas sempre num *  
* mesmo arquivo "bitmap", especificado pela variável arquivo_destino. Isto *  
* permite que um programa externo --- o imcap.m, implementado em MatLab --- *  
* tenha acesso à imagem mais recentemente capturada, pela leitura do arquivo *  
* no momento em que ela se fizer necessária. *  
* *  
* As rotinas da API do Windows utilizadas se encontram na biblioteca *  
* vfw.h ("Video for Windows"). A inclusão da biblioteca windows.h também se *  
* faz necessária à utilização destas funções. *  
* *  
* Última atualização: 10/10/2001. *  
*/  
  
#include <stdio.h>
```

```

#include <windows.h>
#include <vfw.h>

void main(void)
{
    /* Título da janela de captura de imagens a ser criada. */
    char title[70]; // Título com 67 caracteres.
    strcpy(title, "Sistema de Seguimento de Trajetórias");
    strcat(title, " Baseado em Visão Computacional");

    /* Posição e dimensões da janela de captura a ser criada. */
    int x = 0;
    int y = 0;
    int width = 640;
    int height = 480;

    /* "Handle" para a janela de captura a ser criada. */
    HWND hwnd;

    /* Variável que indica se houve ou não conexão da janela criada a um
       "driver" de captura de vídeo. */
    int conecta_driver;

    /* Parâmetro necessário à associação da janela criada ao "driver" de cap-
       tura de imagens (valor padrão: i=0). */
    WPARAM i=0;

    /* Arquivo em que são armazenadas as imagens capturadas. */
    char *arquivo_destino="captura.bmp";
    CAPDRIVERCAPS CapDriverCaps; // variável que, quando da utilização da fun-
    // ção capDriverGetCaps, recebe uma série de
    // informações sobre o "driver" de captura de
    // vídeo conectado a uma janela. Entre estas
    // informações está o fato de apresentar ou
    // não o "driver" uma caixa de diálogo para
    // configuração dos parâmetros de aquisição.

    /* Criação de uma janela com os parâmetros especificados para a captura de
       imagens. */
    hwnd = capCreateCaptureWindow(
        title, // nome
        WS_CLIPSIBLINGS | WS_CLIPCHILDREN, // estilo
        x, y, width, height, // posição e tamanho
        NULL, 0);
}

```

```

/* Se o "handle" retornado pela função capCreateCaptureWindow é NULL, não
pôde ser criada a janela. */
if(hwnd==NULL)
    MessageBox(NULL,
        "Não foi possível criar uma janela para monitoração.",
        "Error", MB_OK);
else
{
    /* Exibição a janela criada. */
    ShowWindow(hwnd, SW_SHOW);

    /* Tentativa de se estabelecer uma conexão entre a janela exibida e o
"driver" de captura. */
    conecta_driver = capDriverConnect(hwnd, i);

    /* Se o valor retornado pela função capDriverConnect for 0, não houve
conexão da janela exibida a um "driver" de captura. */
    if(!conecta_driver)
        MessageBox(NULL,
            "Não houve conexão a um driver de captura de vídeo.",
            "Error", MB_OK);
    else
    {
        /* Informações sobre o "driver" de aquisição de vídeo. Entre elas está
o fato de apresentar ou não o "driver" uma caixa de diálogo para
configuração dos parâmetros de aquisição. */
        capDriverGetCaps(hwnd, &CapDriverCaps, sizeof (CAPDRIVERCAPS));

        /* Caixa de diálogo para configuração da fonte de vídeo (incluindo
resolução das imagens, se o ajuste de brilho e outros parâmetros é
automático ou não, etc). */
        if (CapDriverCaps.fHasDlgVideoSource)
            capDlgVideoSource(hwnd);

        /* Uma nova imagem é capturada e armazenada em arquivo_destino en-
quanto não for pressionada uma tecla pelo usuário, com a ênfase do
sistema voltada para a janela do programa. */
        while(!kbhit())
        {
            capGrabFrame(hwnd); // Captura de uma imagem.
            capFileSaveDIB(hwnd, arquivo_destino); // Armazenamento da imagem.
        }

        /* Após pressionada uma tecla pelo usuário, é feita a desconexão da
janela criada ao "driver" de captura. */
    }
}

```

```

        capDriverDisconnect(hwnd);
    }
}
return(hwnd);
}

```

B.2.2 Rotina de Leitura pelo MatLab das imagens capturadas — imcap.m

```

function I=imcap(arquivo_captura,tipo_imagem);

% I=imcap(arquivo_captura,tipo_imagem);
%
% Faz a leitura da última imagem capturada em arquivo_imagem; um ciclo de ten-
% tativas de leitura garante que o programa externo de captura (imcap.c) pos-
% sa concluir a escrita do arquivo.
%
% A entrada tipo_imagem especifica se a imagem a ser lida é colorida ('cores')
% ou em níveis de cinza ('cinza'). O valor padrão é 'cinza'.
%
% Última modificação: 07/11/2001.

% Se a segunda entrada não for fornecida, é lida uma imagem em níveis de
% cinza.

if nargin<2
    tipo_imagem='cinza';
end

% Ciclo de tentativas de leitura da imagem armazenada em arquivo_captura; en-
% quanto uma nova imagem não é lida, é realizada uma nova tentativa de lei-
% tura. Isto garante que o programa externo de captura (imcap.c) conclua a
% escrita do arquivo, caso isto não se tenha dado antes da primeira tentativa.
I=[];
while (size(I,1)==0)
    [I,map]=imread(arquivo_captura);
end

% Mesmo que a imagem seja em níveis de cinza, imread pode retornar três planos
% (iguais) de imagem. Neste caso, extrai-se o primeiro plano. Caso a imagem
% esteja em formato indexado, ela é convertida para o formato "gray".
if tipo_imagem=='cinza'
    if(size(I,3)==3)
        I=I(:,:,1);
    end
end

```

```
    else
        I=ind2gray(I,map);
    end
end;

I=double(I)/255;
```


Apêndice C

INTERPOLAÇÃO DAS COORDENADAS DA TRAJETÓRIA POR UMA FUNÇÃO POLINOMIAL DA DISTÂNCIA PERCORRIDA

Na segunda etapa da análise de cenas realizada, descrita na seção 7.3, é feita uma interpolação das coordenadas dos pontos extraídos pelo algoritmo de busca circular, em função da distância s_k ao início da trajetória e utilizando como função interpoladora um polinômio de 3º grau. Neste apêndice, é apresentada a técnica utilizada nesta interpolação; o procedimento utilizado é bastante genérico no que diz respeito ao grau do polinômio interpolador.

Considere-se o problema de se calcular os coeficientes do polinômio p de grau m que melhor interpola os N pontos (x_k, y_k) , $1 \leq k \leq N$. Se

$$p(x) = a_m x^m + a_{m-1} x^{m-1} + \dots + a_0 \quad (\text{C.1})$$

é a expressão matemática que define o polinômio, em função da abscissa x , então, supondo o caso ideal em que

$$p(x_k) = y_k \quad \forall 1 \leq k \leq N, \quad (\text{C.2})$$

ou seja, em que o polinômio interpola com exatidão todos os pontos (x_k, y_k) :

$$\begin{cases} a_m x_1^m + a_{m-1} x_1^{m-1} + \dots + a_0 = y_1 \\ a_m x_2^m + a_{m-1} x_2^{m-1} + \dots + a_0 = y_2 \\ \vdots \\ a_m x_N^m + a_{m-1} x_N^{m-1} + \dots + a_0 = y_N \end{cases} \quad (\text{C.3})$$

Em formalismo matricial:

$$\mathbf{X} \cdot \mathbf{a} = \mathbf{y}, \quad (\text{C.4})$$

com:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^m \end{bmatrix}, \quad (\text{C.5})$$

$$\mathbf{a} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix} \text{ e} \quad (\text{C.6})$$

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{bmatrix}. \quad (\text{C.7})$$

A equação (C.4) corresponde a um sistema linear cujas incógnitas são os $m + 1$ coeficientes do polinômio que se deseja obter, dados por $a_0, a_1, a_2, \dots, a_m$. Se $N > m + 1$, ou seja, se há mais pontos a serem interpolados do que coeficientes a calcular, o sistema apresenta mais equações do que incógnitas. Neste caso, pode-se utilizar a solução pelo método dos mínimos quadrados, que permite minimizar o erro quadrático $e = \|\mathbf{X} \cdot \mathbf{a} - \mathbf{y}\|^2$ [31][30] e assim obter os coeficientes do polinômio que melhor interpola os N pontos (x_k, y_k) :

$$\mathbf{a} = (\mathbf{X}^T \cdot \mathbf{X})^{-1} \cdot \mathbf{X}^T \cdot \mathbf{y}. \quad (\text{C.8})$$

Apêndice D

RESULTADOS INTERMEDIÁRIOS OBTIDOS DURANTE O RASTREAMENTO

D.1 PRIMEIRA TRAJETÓRIA

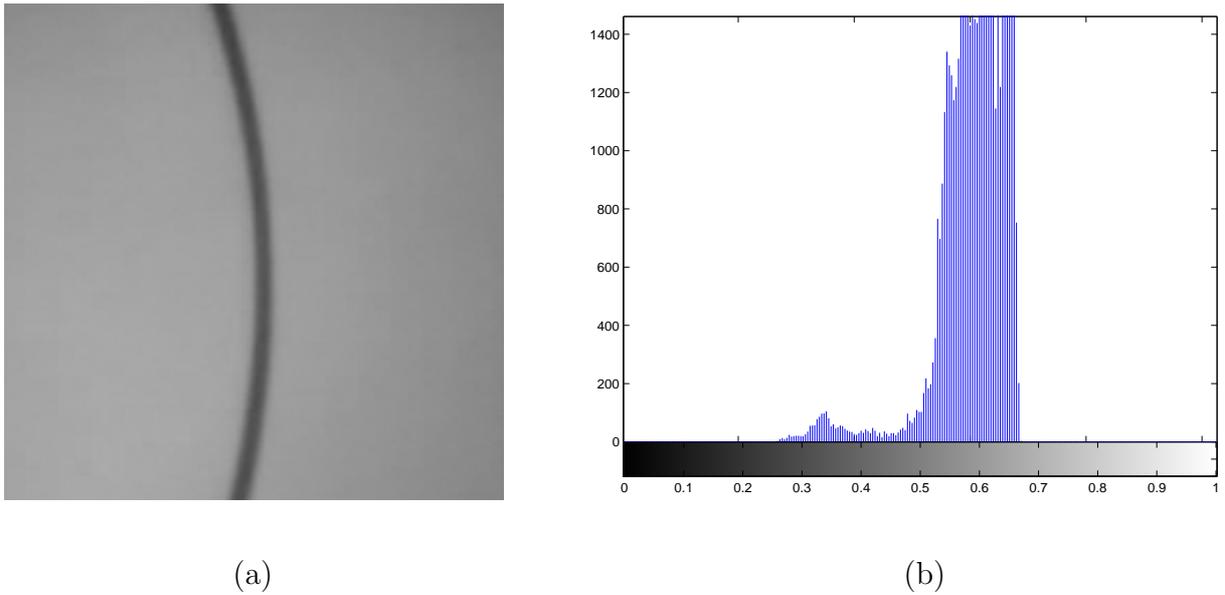


Figura D.1: (a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.14-(a)). (b) Histograma correspondente.

Polinômios interpoladores:

$$\Delta x(d) = -2,6106 \cdot 10^{-5} \cdot d^3 + 9,1656 \cdot 10^{-5} \cdot d^2 + 1,1169 \cdot d - 5,6971 \cdot 10^{-3}. \quad (\text{D.1})$$

$$\Delta y(s) = 2,8373 \cdot 10^{-4} \cdot d^3 - 1,9928 \cdot 10^{-2} \cdot d^2 + 1,4225 \cdot 10^{-1} \cdot d + 1,6702 \cdot 10^{-2}. \quad (\text{D.2})$$

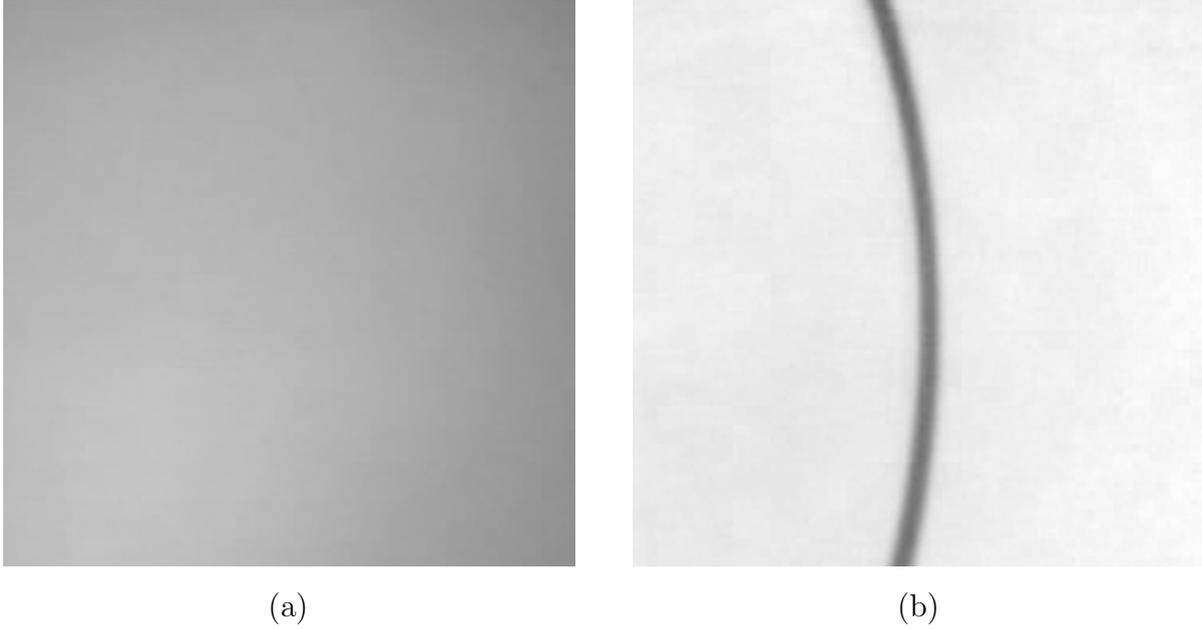
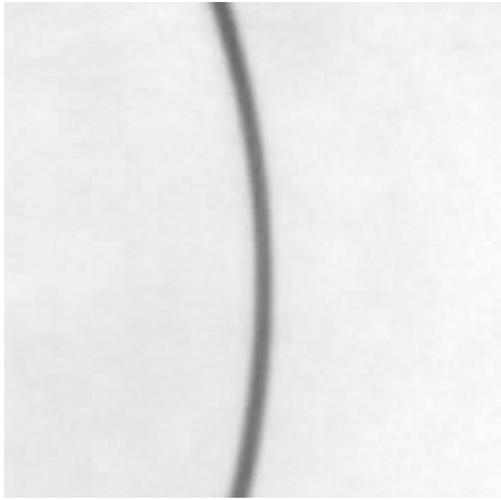


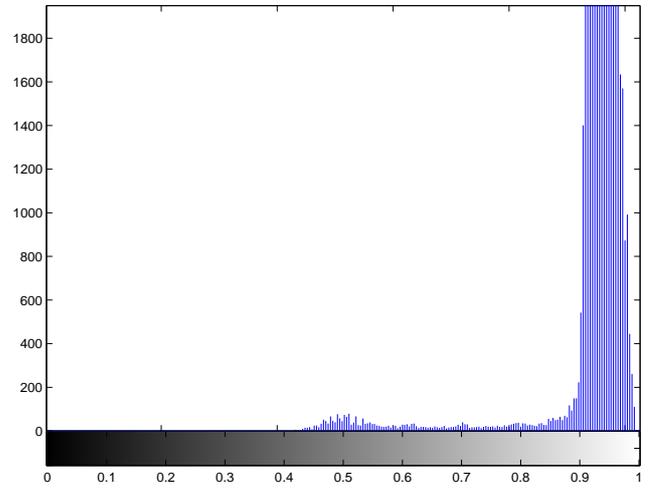
Figura D.2: (a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.1-(a). (b) Imagem obtida após a compensação.

Tabela D.1: Parâmetros estatísticos referentes à imagem da figura D.3-(a), utilizados no cálculo do limiar ótimo de binarização.

Média da distribuição associada à trajetória (μ_T^2)	0.63756
Variância da distribuição associada à trajetória (σ_T^2)	0.018666
Média da distribuição associada ao fundo (μ_F^2)	0.93886
Variância da distribuição associada ao fundo (σ_F^2)	0.00043546
Proporção de trajetória na imagem (p_T)	0.056528
<i>Limiar de binarização (L)</i>	0.86602



(a)



(b)

Figura D.3: (a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.1-(a). (b) Histograma correspondente.

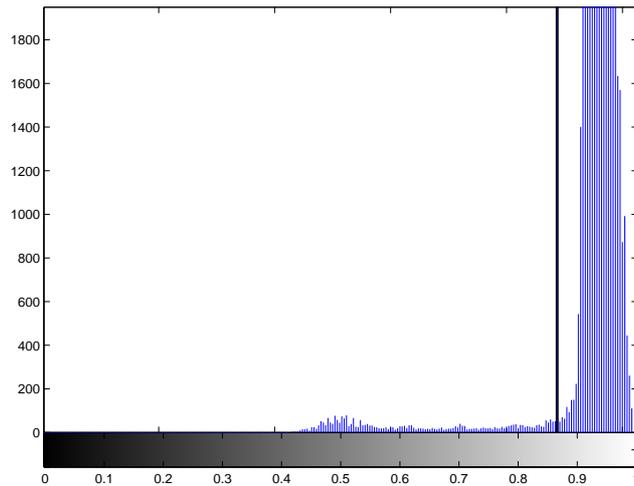


Figura D.4: Histograma associado à imagem da figura D.3-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.1).

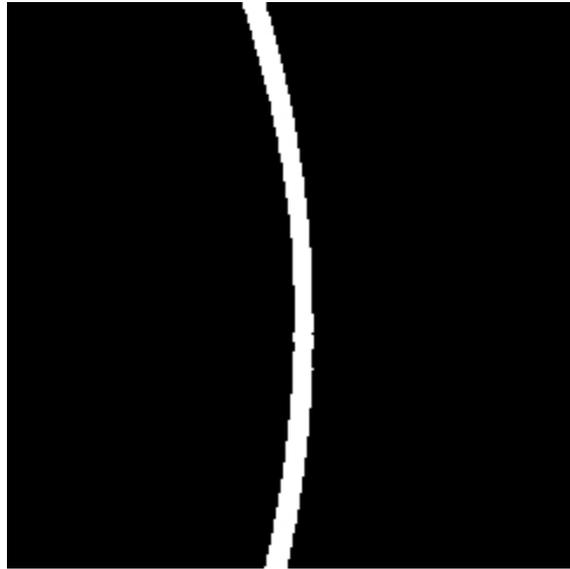


Figura D.5: Imagem da figura D.3-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.4.

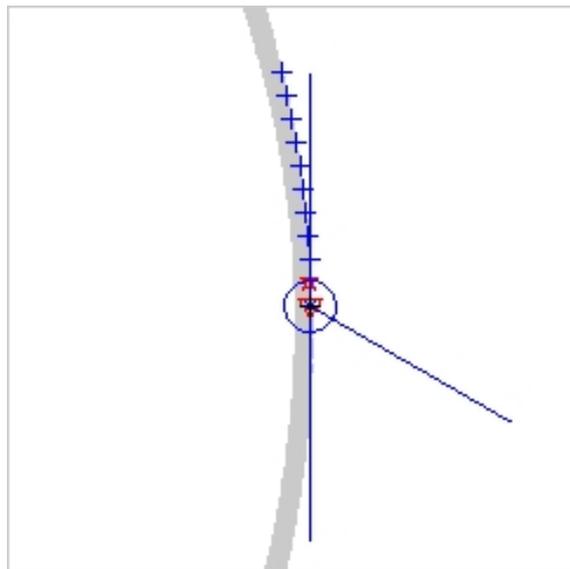


Figura D.6: Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.5 pela aplicação do algoritmo de busca circular.

Tabela D.2: Coordenadas dos *pixels* extraídos da imagem da figura D.5 pela aplicação do algoritmo de busca circular (figura D.6).

<i>Pixel</i>	Coordenada <i>i</i>	Coordenada <i>j</i>
1	128	128
2	118	128
3	108	128
4	98	127
5	88	126
6	78	125
7	68	124
8	58	122
9	48	120
10	38	118
11	28	116

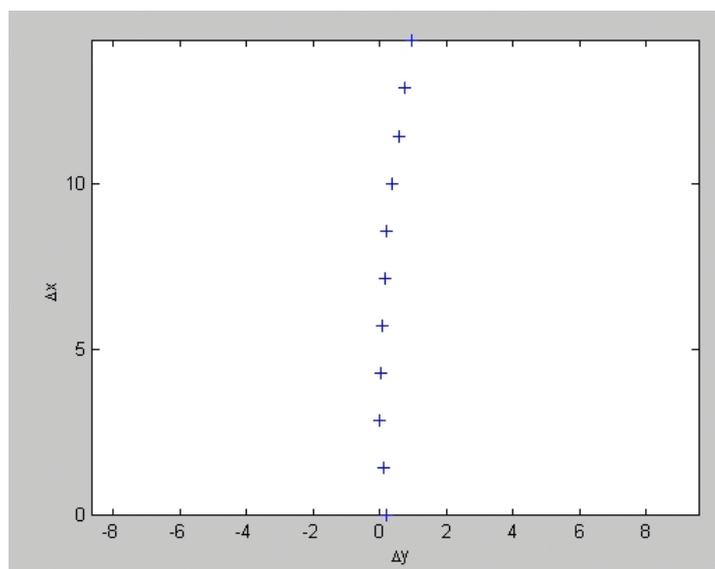


Figura D.7: Entradas do módulo de controle da posição do atuador obtidas da figura D.6, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.2) em variações na posição do atuador (transformação dada pela equação (7.34)).

Tabela D.3: Entradas do módulo de controle da posição do atuador obtidas da figura D.6, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.2) em variações na posição do atuador (transformação dada pela equação (7.34)).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	1.6097	0.1641	1.4445
3	3.2194	0.3281	2.8890
4	4.8418	0.2511	4.3423
5	6.4642	0.1741	5.7955
6	8.0866	0.0971	7.2488
7	9.709	0.0201	8.7020
8	11.344	-0.298	10.1782
9	12.9791	-0.6161	11.6543
10	14.6141	-0.9342	13.13051
11	16.2492	-1.2522	14.6067

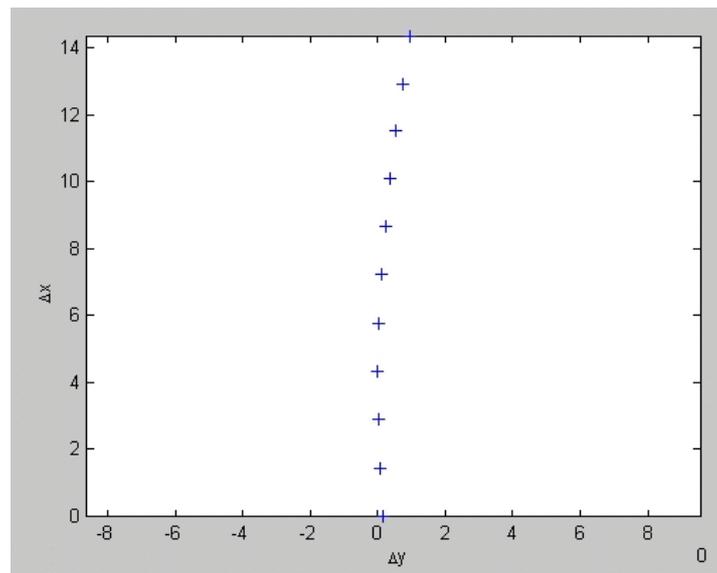


Figura D.8: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.1) e (D.2) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Tabela D.4: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.1) e (D.2) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	1.6259	0.1828	1.4607
3	3.2573	0.2693	2.9213
4	4.8882	0.2812	4.3820
5	6.5180	0.2241	5.8427
6	8.1462	0.1032	7.3033
7	9.7724	-0.0763	8.7640
8	11.3960	-0.3089	10.2247
9	13.0166	-0.5895	11.6853
10	14.6337	-0.9126	13.1460
11	16.2468	-1.2730	14.6067

D.2 TERCEIRA TRAJETÓRIA

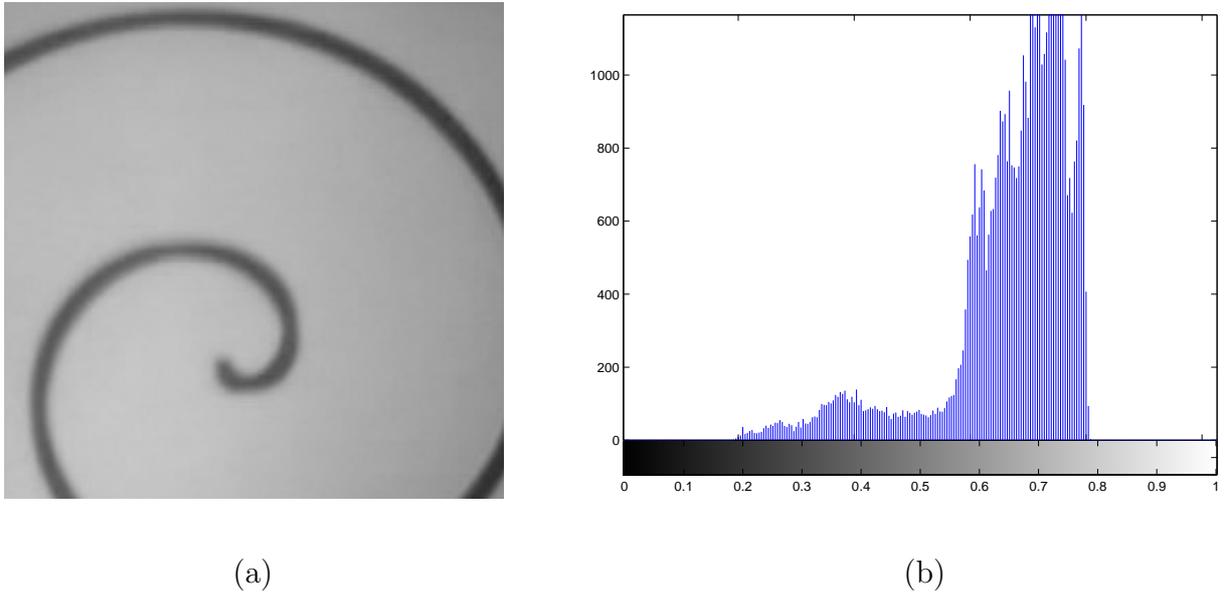


Figura D.9: (a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.16-(a)). (b) Histograma correspondente.

Tabela D.5: Parâmetros estatísticos referentes à imagem da figura D.11-(a), utilizados no cálculo do limiar ótimo de binarização.

Média da distribuição associada à trajetória (μ_T^2)	0.6604
Variância da distribuição associada à trajetória (σ_T^2)	0.030053
Média da distribuição associada ao fundo (μ_F^2)	0.95995
Variância da distribuição associada ao fundo (σ_F^2)	0.00019787
Proporção de trajetória na imagem (p_T)	0.17168
<i>Limiar de binarização</i> (L)	0.91476

Polinômios interpoladores:

$$\Delta x(d) = -2,7051 \cdot 10^{-5} \cdot d^3 - 4,7172 \cdot 10^{-2} \cdot d^2 + 6,9518 \cdot 10^{-1} \cdot d - 3,0483 \cdot 10^{-1}. \quad (D.3)$$

$$\Delta y(d) = 2,3003 \cdot 10^{-3} \cdot d^3 - 5,0670 \cdot 10^{-2} \cdot d^2 - 1,3038 \cdot d + 6,2185 \cdot 10^{-1}. \quad (D.4)$$

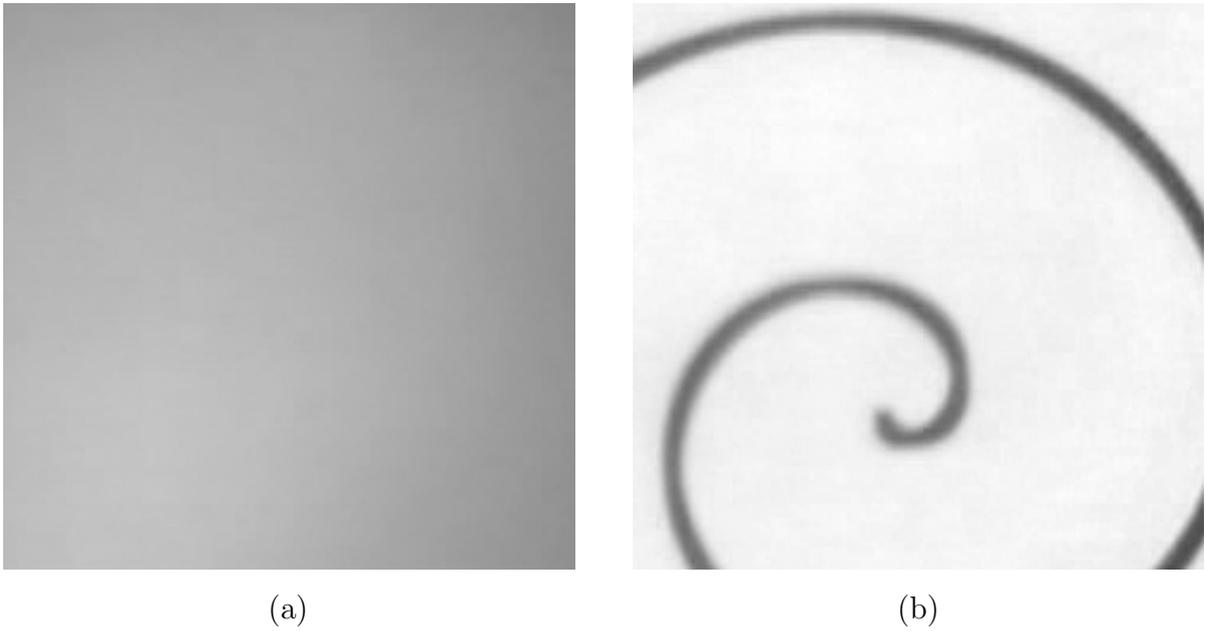


Figura D.10: (a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.9-(a). (b) Imagem obtida após a compensação.

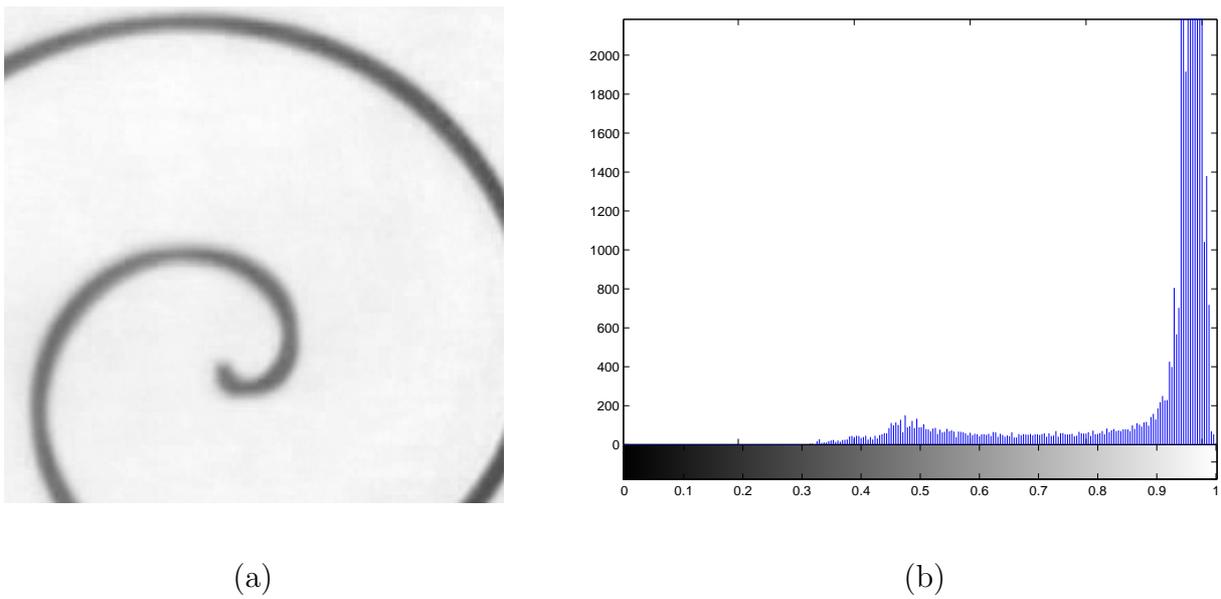


Figura D.11: (a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.9-(a). (b) Histograma correspondente.

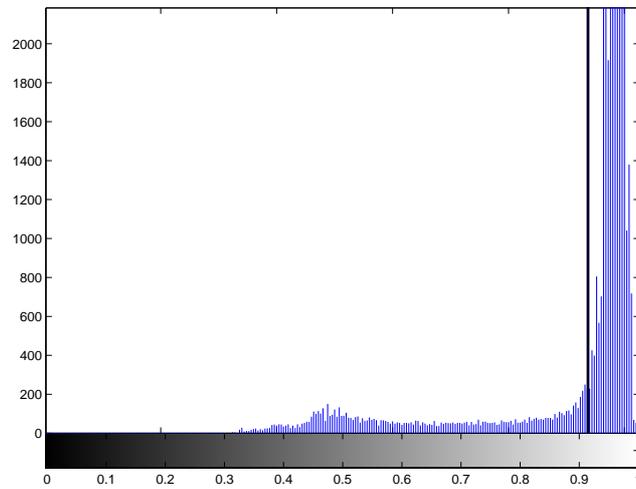


Figura D.12: Histograma associado à imagem da figura D.11-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.5).



Figura D.13: Imagem da figura D.11-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.12.

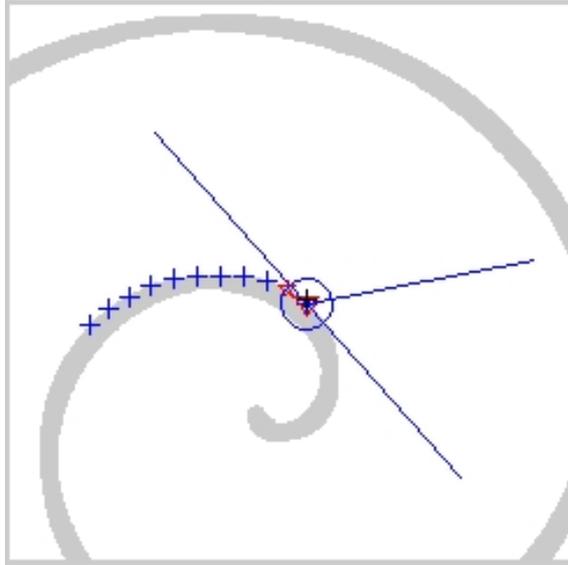


Figura D.14: Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.13 pela aplicação do algoritmo de busca circular.

Tabela D.6: Coordenadas dos *pixels* extraídos da imagem da figura D.13 pela aplicação do algoritmo de busca circular (figura D.14).

<i>Pixel</i>	Coordenada <i>i</i>	Coordenada <i>j</i>
1	130	128
2	124	120
3	120	111
4	118	101
5	118	91
6	118	81
7	119	71
8	122	61
9	127	52
10	132	43
11	139	35

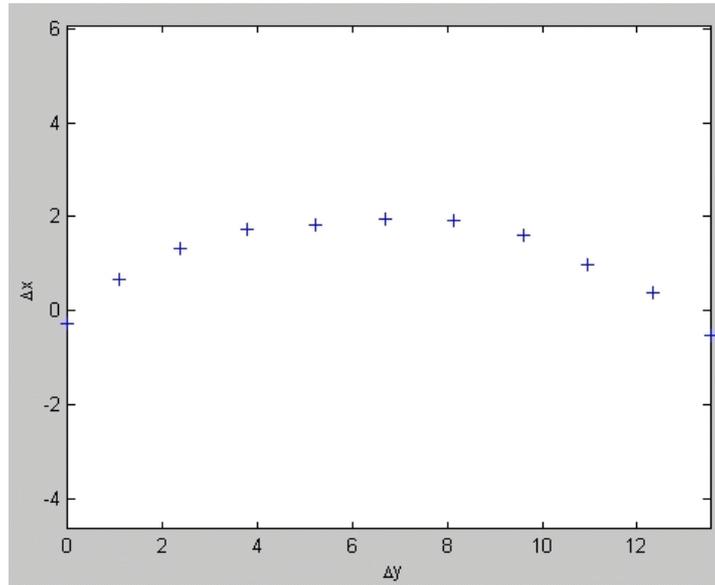


Figura D.15: Entradas do módulo de controle da posição do atuador obtidas da figura D.14, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.6) em variações na posição do atuador (transformação dada pela equação (7.34)).

Tabela D.7: Entradas do módulo de controle da posição do atuador obtidas da figura D.14, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.6) em variações na posição do atuador (transformação dada pela equação (7.34)).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	0.7452	-1.8629	1.7760
3	1.5032	-3.9669	3.2049
4	1.9518	-6.3448	4.6819
5	2.0785	-8.7555	6.1272
6	2.2052	-11.1661	7.5725
7	2.1709	-13.5932	9.0235
8	1.8147	-16.0531	10.5280
9	1.1239	-18.3047	12.0091
10	0.4331	-20.5564	13.4902
11	-0.5924	-22.5998	15.0181

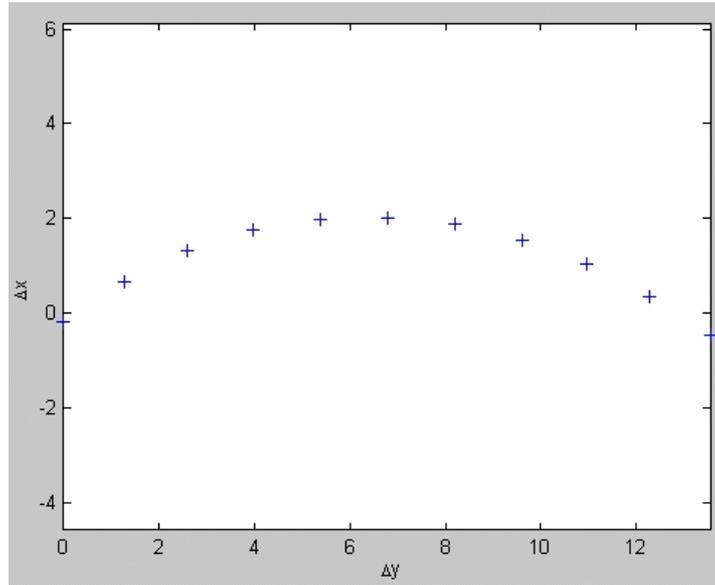


Figura D.16: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.3) e (D.4) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Tabela D.8: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.3) e (D.4) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	0.6327	-1.4426	1.5018
3	1.3569	-3.6889	3.0036
4	1.8673	-6.0703	4.5054
5	2.1631	-8.5400	6.0072
6	2.2441	-11.0512	7.5091
7	2.1094	-13.5573	9.0109
8	1.7587	-16.0114	10.5127
9	1.1914	-18.3669	12.0145
10	0.4068	-20.5769	13.5163
11	-0.5955	-22.5947	15.0181

D.3 QUARTA TRAJETÓRIA

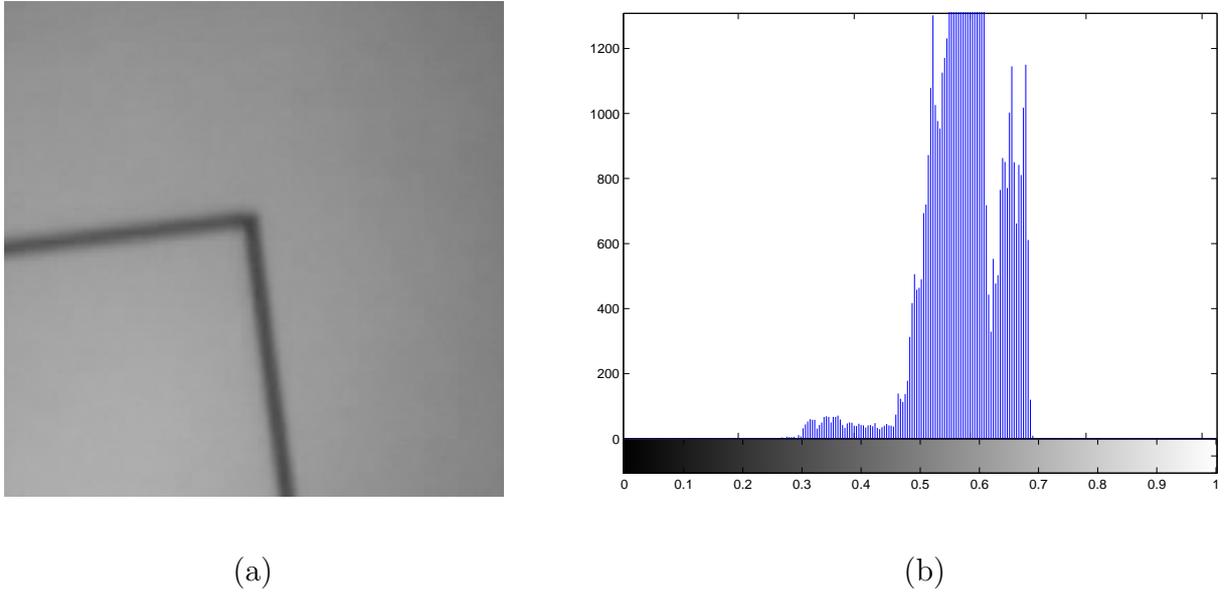


Figura D.17: (a) Exemplo de imagem adquirida, a partir de uma posição arbitrária, de uma das trajetórias rastreadas (figura 8.17-(a)). (b) Histograma correspondente.

Tabela D.9: Parâmetros estatísticos referentes à imagem da figura D.19-(a), utilizados no cálculo do limiar ótimo de binarização.

Média da distribuição associada à trajetória (μ_T^2)	0.65727
Variância da distribuição associada à trajetória (σ_T^2)	0.016076
Média da distribuição associada ao fundo (μ_F^2)	0.93668
Variância da distribuição associada ao fundo (σ_F^2)	0.00050449
Proporção de trajetória na imagem (p_T)	0.057535
<i>Limiar de binarização (L)</i>	0.86013

Polinômios interpoladores:

$$\Delta x(d) = 5,7845 \cdot 10^{-3} \cdot d^3 - 1,7726 \cdot 10^{-1} \cdot d^2 + 1,6495 \cdot d - 7,2917 \cdot 10^{-1}. \quad (\text{D.5})$$

$$\Delta y(d) = 6,7866 \cdot 10^{-3} \cdot d^3 - 2,0244 \cdot 10^{-1} \cdot d^2 + 1,9720 \cdot 10^{-1} \cdot d - 8,6969 \cdot 10^{-2}. \quad (\text{D.6})$$

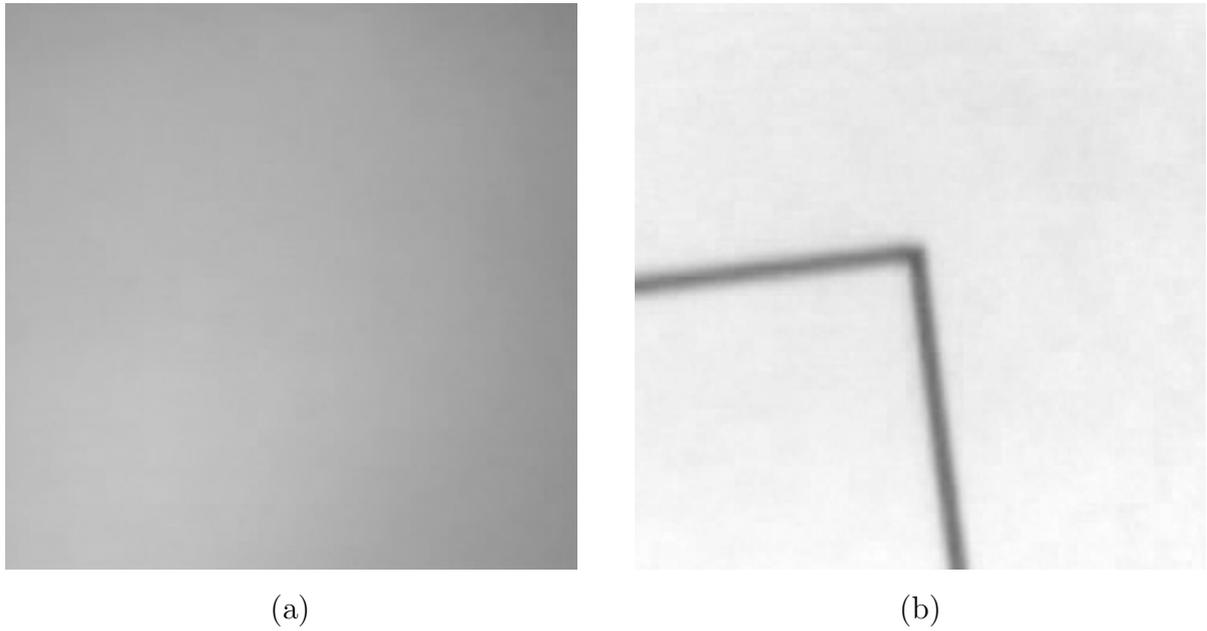


Figura D.18: (a) Padrão de iluminação utilizado durante a compensação da não-uniformidade da iluminação na geração da imagem da figura D.17-(a). (b) Imagem obtida após a compensação.

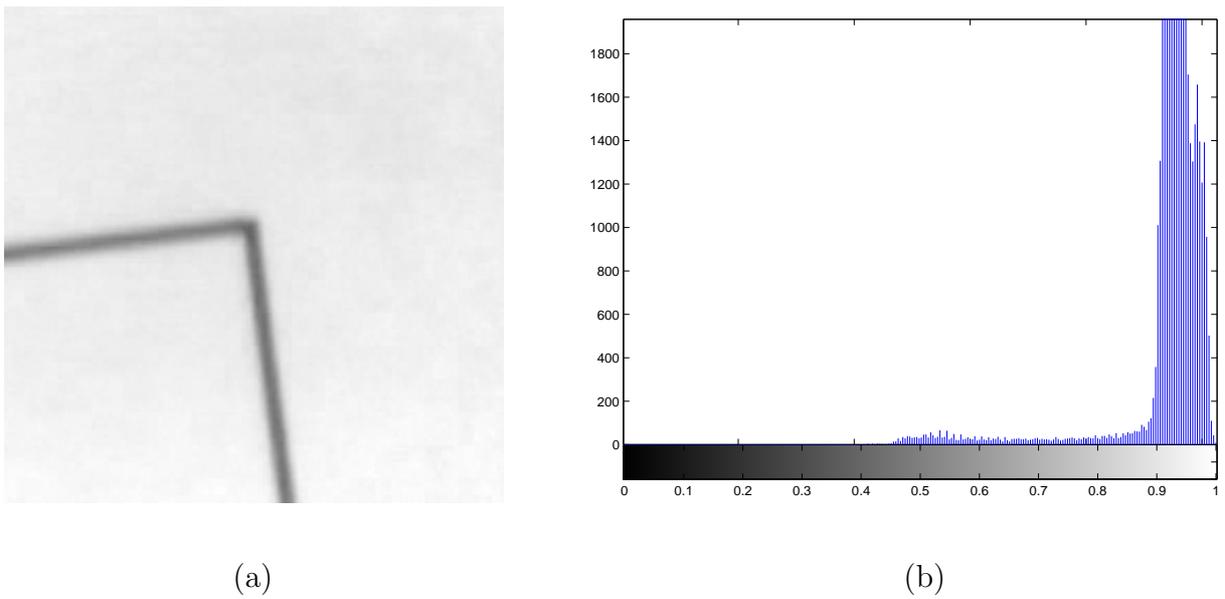


Figura D.19: (a) Imagem obtida pela compensação da não-uniformidade da iluminação na geração da imagem da figura D.17-(a). (b) Histograma correspondente.

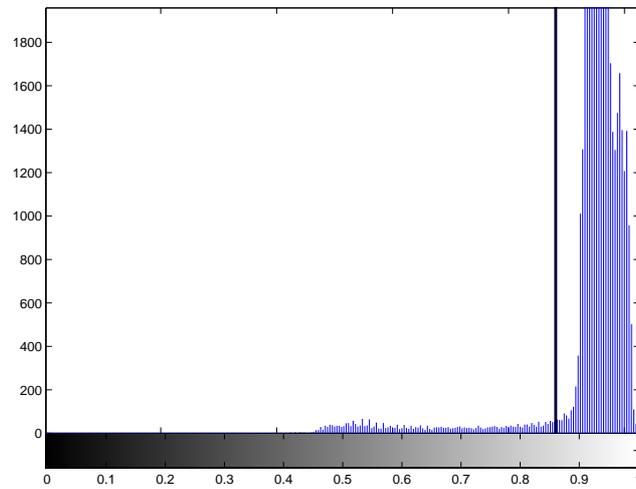


Figura D.20: Histograma associado à imagem da figura D.19-(a) (compensada quanto à iluminação), com destaque para o limiar de binarização calculado (tabela D.9).

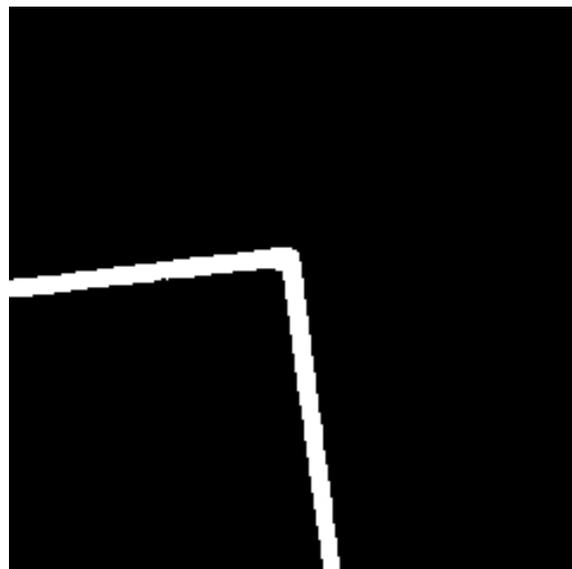


Figura D.21: Imagem da figura D.19-(a) (compensada quanto à iluminação) após o processo de binarização com base no limiar destacado na figura D.20.

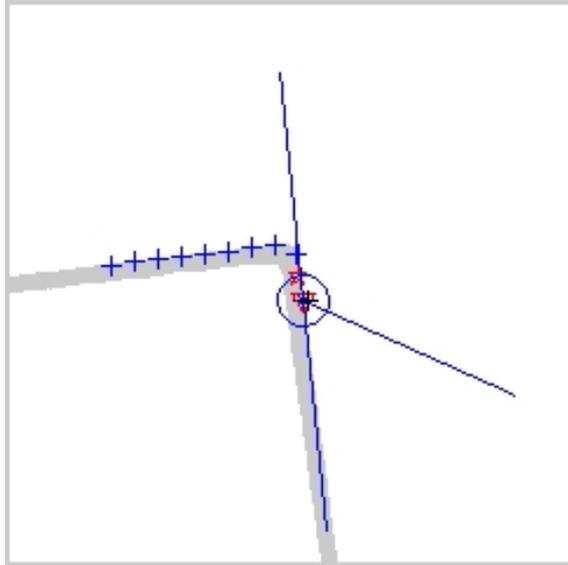


Figura D.22: Pontos de referência de entrada do sistema de rastreamento extraídos a partir da imagem da figura D.21 pela aplicação do algoritmo de busca circular.

Tabela D.10: Coordenadas dos *pixels* extraídos da imagem da figura D.21 pela aplicação do algoritmo de busca circular (figura D.22).

<i>Pixel</i>	Coordenada <i>i</i>	Coordenada <i>j</i>
1	128	126
2	118	124
3	108	123
4	104	114
5	105	104
6	107	94
7	108	84
8	109	74
9	110	64
10	111	54
11	113	44

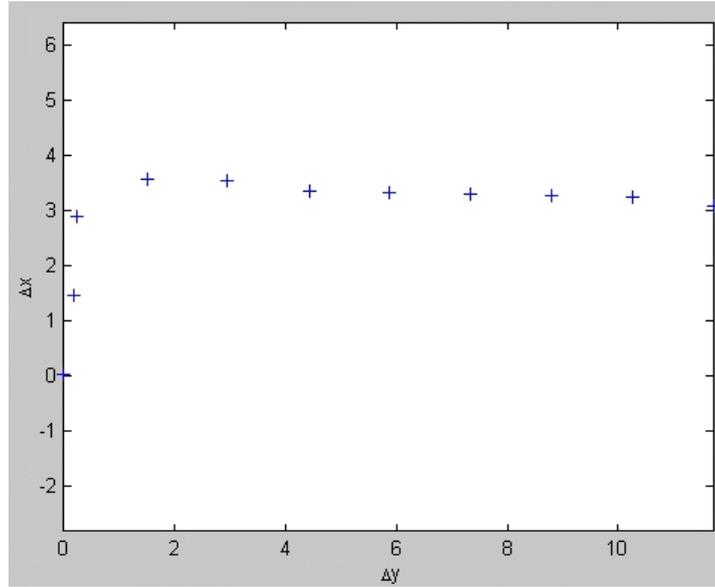


Figura D.23: Entradas do módulo de controle da posição do atuador obtidas da figura D.22, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.10) em variações na posição do atuador (transformação dada pela equação (7.34)).

Tabela D.11: Entradas do módulo de controle da posição do atuador obtidas da figura D.22, pela conversão das distâncias do centro da imagem aos *pixels* extraídos (tabela D.10) em variações na posição do atuador (transformação dada pela equação (7.34)).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	1.6604	-0.8002	1.9590
3	3.2828	-0.8772	3.4122
4	4.0407	-2.9812	4.8412
5	4.0064	-5.4083	6.2922
6	3.8112	-7.8518	7.7630
7	3.7769	-10.2789	9.2140
8	3.7426	-12.7060	10.6650
9	3.7084	-15.1330	12.1160
10	3.6741	-17.5601	13.5670
11	3.4788	-20.0036	15.0379

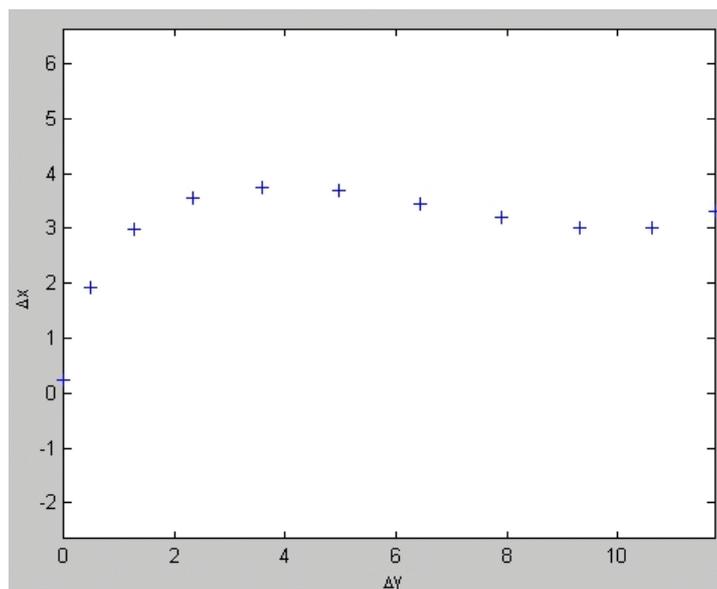


Figura D.24: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.5) e (D.6) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Tabela D.12: Entradas do módulo de controle da posição do atuador uniformemente espaçadas, obtidas pela aplicação dos polinômios (D.5) e (D.6) a distâncias s ao ponto de referência uniformemente crescentes (distância entre pontos consecutivos constante).

Ponto	Variações na coordenada x (Δx)	Variações na coordenada y (Δy)	Distância ao ponto de referência, medida ao longo da trajetória (mm)
2	1.3702	-0.2251	1.5038
3	2.7859	-1.1404	3.0076
4	3.6360	-2.6943	4.5114
5	4.0385	-4.7483	6.0152
6	4.1113	-7.1640	7.5190
7	3.9727	-9.8029	9.0227
8	3.7404	-12.5265	10.5265
9	3.5327	-15.1964	12.0303
10	3.4675	-17.6741	13.5341
11	3.6628	-19.8211	15.0379