

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**RASTREAMENTO EM VÍDEO DAS
CARACTERÍSTICAS DA PUPILA**

Cláriton Rodrigues Bernadelli

Uberlândia, MG

2011

RASTREAMENTO EM VÍDEO DAS CARACTERÍSTICAS DA PUPILA

Cláriton Rodrigues Bernadelli

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciências - Área de concentração: Processamento Digital de Sinais.

Prof. Dr. Antônio C. P. Veiga
Orientador

Prof. Dr. Alexandre Cardoso
Coordenador do curso de Pós-Graduação

Uberlândia, MG

2011

RASTREAMENTO EM VÍDEO DAS CARACTERÍSTICAS DA PUPILA

Cláriton Rodrigues Bernadelli

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de Mestre em Ciências - Área de concentração: Processamento Digital de Sinais.

Aprovada em 19 de Dezembro de 2011.

Banca examinadora:

Antônio. C. P. Veiga, Dr - Orientador (UFU)

Henrique Takachi Moriya, Dr (USP)

Gilberto Arantes Carrijo, PhD (UFU)

Edna Lúcia Flôres, Dra (UFU)

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG - Brasil

B517r Bernadelli, Cláriton Rodrigues, 1979-
2011 Rastreamento em vídeo das características da pupila / Cláriton
Rodrigues Bernadelli. - 2011.

131 p. : il.

Orientador: Antônio C. P. Veiga.

Dissertação (mestrado) ó Universidade Federal de Uberlândia,
Programa de Pós-Graduação em Engenharia Elétrica.
Inclui bibliografia.

1. Engenharia elétrica - Teses. 2. Pupila (Olhos) - Teses. I. Veiga,
Antônio Cláudio Paschoarelli, 1963- II. Universidade Federal de
Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III.
Título.

CDU: 621.3

*Aos meus pais,
Pela vida e pelo exemplo de vida.*

Agradecimentos

Aos meus amados pais, Henrique e Eunice, pelo apoio, amor e respeito que sempre dedicaram a nossa família. Meus grandes exemplos!

Aos meus irmãos, Sérgio e Patrícia, melhores amigos que alguém poderia ter.

À minha querida Dri, pelo amor, amizade e companheirismo.

Ao meu orientador Prof. Dr. Paschoarelli.

Aos meus professores, especialmente ao Prof. PhD Gilberto Carrijo e Prof. Dra. Edna Flôres.

A todos os meus colegas de pós-graduação. Ao Tiago Henrique e ao professor Carlos Augusto do Laboratório de Automação, Servomecanismos e Controle (LASEC), pela grande ajuda na construção do protótipo e ao físico e futuro engenheiro eletricista Alessandro Gontijo, pela imprescindível participação.

A todos os colegas e amigos da Algar Segurança, pelos momentos de descontração, pelo incentivo e apoio. Em especial a Sueli, Carlos Coelho, Vera, David, João Marcos e ao grande parceiro de viola caipira, Paulo.

Aos meus alunos, pelo aprendizado e inspiração.

À CAPES pelo suporte financeiro a este trabalho.

A Deus, por fazer tudo isso possível.

“O essencial é invisível aos olhos”

Antoine de Saint - Exupèry



Resumo

Bernadelli, C. R. & Veiga, A. C. P. Rastreamento em vídeo das características da pupila.

Processamento de Imagens - Vídeo Tracking, FEELT-UFU, Uberlândia, 2011, 131p.

O estudo dos movimentos da pupila tem interesse clínico relevante, pois essa atua como um indicador objetivo da sensibilidade da retina à luz e, por consequência, do nervo óptico. A sua oscilação permanente é resultado do equilíbrio entre os fluxos opostos do sistema nervoso simpático e parassimpático. O estado da pupila de um indivíduo permite o diagnóstico de várias doenças, entre as quais distúrbios do sono (narcolepsia), esquizofrenia, Alzheimer, entre outras. Este trabalho apresenta o protótipo de um equipamento estável capaz de rastrear, em vídeo, a área da pupila e detectar sua variação em uma escala temporal devido à influência da iluminação. O algoritmo de rastreamento implementado em um programa de interface simples baseia-se no método de detecção de contorno, onde a pupila é aproximada por circunferências *frame a frame*, utilizando a transformada circular de Hough. Todas as variáveis observadas tiveram o comportamento esperado quando comparadas com seus principais modelos teóricos. O tamanho inicial da pupila em relação à íris, antes do estímulo luminoso, ficou entre 24 e 39%, sendo os maiores valores referentes a iluminação em torno do vermelho (550 – 650 nm) e do verde (450 – 600 nm). A mesma relação para a situação de máxima constrição ficou entre 17 e 26%. O tempo médio de latência variou de 388 a 110 ms, enquanto o tempo médio de latência para a máxima constrição variou de 2 s a 500 ms. A velocidade de constrição variou de 0,75 a 3,6 mm/s enquanto a velocidade de dilatação variou de 0,25 a 1,47 mm/s. Ambas as velocidades alcançaram os maiores valores nas intensidades luminosas intermediárias, referentes as iluminações em torno do vermelho (550 – 650 nm) e do verde (450 – 600 nm), comprovando a relação com a máxima amplitude.

Palavras-chave

Matching, Pupila, Transformada de Hough, Vídeo Tracking.

Abstract

Bernadelli, C. R. & Veiga, A. C. P. Video tracking of pupil features.
Image Processing - Video Tracking, FEELT-UFU, Uberlândia, 2011, 131p.

The study of the pupil's movements has significant clinical interest because it acts as an objective indicator of retinal sensitivity to light and, consequently, of the optic nerve. Its permanent oscillation is the result of balance between opposing flows of the sympathetic and parasympathetic nervous system. The pupil's state of an individual allows the diagnosis of various diseases, including sleep disorders (narcolepsy), schizophrenia, Alzheimer's, among others. This project presents a stable prototype device capable of tracking in video, the pupil area and detects its variation on a time scale due to the lighting influence. The implemented tracking algorithm in a simple interface program is based on the detection method of contour, where the pupil is approximated by circles frame by frame, using the Hough transform for circles. All observed variables had the expected behavior when compared with their main theoretical models. The initial size of the pupil in relation to the iris, before the light stimulus, was between 24 and 39%, and the highest values for lighting around the red (550 – 650 nm) and green (450 – 600 nm). The same relationship for the situation of maximum constriction was between 17 and 26%. The average latency time ranged from 388 to 110 ms, while the average latency to maximum constriction ranged from 2 s to 500 ms. The speed of constriction varied from 0.75 to 3.6 mm/s while the speed of expansion ranged from 0.25 to 1.47 mm/s. Both rates reached the highest values in the intermediate light intensities referring to the lights around the red (550 – 650 nm) and green (450 – 600 nm), confirming the relationship with the maximum amplitude.

Keywords

Hough Transform, Matching, Pupil, Video Tracking.

Sumário

Sumário	x
Lista de Figuras	xiii
Lista de Tabelas	xv
Lista de Códigos	xvi
Lista de Abreviações	xvii
1 INTRODUÇÃO	1
1.1 Introdução	1
1.2 Objetivos deste Trabalho	3
1.3 Estado da Arte	4
1.4 Estrutura deste Trabalho	5
1.5 Considerações Finais deste Capítulo	6
2 MOVIMENTOS PUPILARES	7
2.1 Introdução	7
2.2 Midríase: dilatação pupilar	10
2.3 Miose: contração pupilar	11
2.4 Pupilometria	13
2.5 Considerações Finais deste Capítulo	15

3	VIDEO TRACKING	16
3.1	Introdução	16
3.2	Rastreamento de Objetos Rígidos	17
3.2.1	Rastreamento de objetos rígidos 2D	17
3.2.2	Tratamento de Oclusões	22
3.2.3	Rastreamento de Objetos Rígidos 3D	23
3.3	Rastreamento de Objetos Articulados	24
3.3.1	Rastreamento de Objetos Articulados 3D	24
3.4	Considerações Finais	25
4	EXTRAÇÃO DE CARACTERÍSTICA EM IMAGENS DIGITAIS POR MEIO DA CORRESPONDÊNCIA DE FORMA	27
4.1	Introdução	27
4.2	Limiarização (<i>Thresholding</i>) e <i>Subtraction</i>	28
4.3	Modelo de Correspondência <i>Template Matching</i>	29
4.4	<i>Template Matching</i> Utilizando Transformada de Fourier (TF)	33
4.5	Transformada de Hough (TH)	35
4.5.1	TH para linhas	36
4.5.2	TH para Círculos	40
4.5.3	TH para Elipses	43
4.5.4	Decomposição do Espaço de Parâmetros	47
4.5.5	Redução do Espaço de Parâmetros para Linhas	48
4.5.6	Redução do Espaço de Parâmetros para Círculos	49
4.5.7	Redução do Espaço de Parâmetros para Elipses	55
4.6	Outras Extensões para a TH	60
4.7	Considerações Finais deste Capítulo	60
5	METODOLOGIA, AQUISIÇÃO E PROCESSAMENTO DO VIDEO STREAM	62
5.1	Introdução	62
5.2	Padrões Pupilares para o Propósito de Pesquisa e Prática Clínica	62
5.3	Modelos para Simulação de Dinâmica Pupilar	63
5.3.1	Modelo de Moon e Spencer	63
5.3.2	Modelo de Groot e Gebhard	64
5.3.3	Modelo de Pokorny e Smith	65
5.3.4	Modelo de Ellis	65
5.3.5	Modelo de Link e Stark	67
5.4	Sincronizando os <i>Frames</i> do <i>Video Stream</i> com a Fonte de Iluminação	67
5.4.1	Controle do Brilho	71
5.5	Processamento do <i>Video Stream</i>	72
5.5.1	Pré processamento	72

5.6	Considerações Finais deste Capítulo	74
6	APLICAÇÃO DO SISTEMA PROPOSTO	75
6.1	Introdução	75
6.2	Aquisição dos <i>Video Streams</i>	75
6.3	Resultados Obtidos	79
6.4	Tratamento Automático de Oclusões	93
6.5	Conclusões	94
7	CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS	95
7.1	Introdução	95
7.2	Principais Contribuições	96
7.3	Publicações	97
7.4	Trabalhos futuros	97
Referências Bibliográficas		99
Apêndice		104
A	Aquisição e processamento	104
A.1	Sistema de sincronismo	104
	A.1.1 Rotina interna do micro controlador	104
	A.1.2 Rotina interna da interface C Sharp	107
A.2	Processamento do <i>video stream</i>	113
B	Descrição dos equipamentos utilizados	122
B.1	Webcam	122
B.2	LEDs	124

Lista de Figuras

1.1	<i>Representação anatômica do sistema nervoso.</i>	2
1.2	<i>Diferentes tamanhos e posições de pupila podem e devem definir diferentes procedimentos cirúrgicos.</i>	3
2.1	<i>Diagrama simplificado de um corte transversal do olho humano.</i>	8
2.2	<i>Parâmetros medidos em um ciclo pupilar: latência (T1), máxima velocidade de constrição (VCmax), tempo para a máxima constrição (T3), amplitude (AMP), raio inicial (R1), raio mínimo (R2) e máxima aceleração de constrição (ACmax).</i>	11
2.3	<i>Percorso nervoso visual.</i>	12
3.1	<i>Etapas do rastreamento de objetos.</i>	18
3.2	<i>Exemplo de rastreamento baseado em espaço de cor.</i>	19
3.3	<i>Exemplo de rastreamento baseado em contorno ativo.</i>	20
3.4	<i>Exemplo de rastreamento baseado em pontos característicos.</i>	21
3.5	<i>Exemplo de representação hierárquica do corpo humano.</i>	25
4.1	<i>Extração de forma utilizando subtração e limiarização.</i>	29
4.2	<i>Template matching.</i>	30
4.3	<i>Template matching no domínio de Fourier.</i>	35
4.4	<i>Transformada de Hough para linhas.</i>	37
4.5	<i>Aplicação da TH para linhas.</i>	38
4.6	<i>Linha na forma polar.</i>	39
4.7	<i>Transformada de Hough para círculos.</i>	41
4.8	<i>Aplicação da TH para círculos.</i>	43
4.9	<i>Uso da TH para círculos.</i>	44
4.10	<i>Definição dos eixos da elipse.</i>	46
4.11	<i>Aplicação da TH para elipses.</i>	47

4.12	<i>Espaço de parâmetros reduzido para TH de linhas.</i>	49
4.13	<i>Definição da primeira e da segunda derivada para um círculo.</i>	51
4.14	<i>Geometria do ângulo da primeira e da segunda derivada direcional.</i>	53
4.15	<i>Redução do espaço de parâmetros da TH para círculos.</i>	55
4.16	<i>Geometria do ângulo da primeira e da segunda derivada direcional.</i>	56
4.17	<i>Redução do espaço de parâmetros para a TH em elipses.</i>	60
5.1	<i>Medidas extraídas do output pupilométrico. 1-Tempo de latência do alcance do plateau; 2-Tempo de latência para o início da constrição; 3-Duração da constrição; 4-Tempo de latência para a maior constrição; 5-Raio da pupila no alcance do plateau; 6-Raio da pupila para a máxima constrição; 7-Raio da pupila antes do disparo do flash.</i>	64
5.2	<i>Comparação entre os modelos de Moon e Spencer; Groot e Gebhard; e Pokorny e Smith.</i>	66
5.3	<i>Curvas características do LED modelo LX61 da Cinestec Comp. Eletrônicos.</i>	68
5.4	<i>Esquema do hardware de controle.</i>	69
5.5	<i>Interface para controle do PWM (controle da intensidade luminosa).</i>	69
5.6	<i>Diagrama de blocos do processo de sincronismo entre os frames e o sistema de iluminação.</i>	70
5.7	<i>Diagrama de blocos do tracking da pupila. Algoritmo utilizado na detecção do raio e da posição da pupila do protótipo proposto.</i>	73
5.8	<i>Detecção de bordas por Canny.</i>	73
5.9	<i>Interface para ajuste dos parâmetros da pupila.</i>	74
6.1	<i>Fontes de iluminação com diferentes comprimentos de onda.</i>	76
6.2	<i>Pupil light reflex (PLR) com fontes de luz de diferentes intensidades e comprimentos de onda.</i>	78
6.3	<i>Comparação entre as saídas pupilométricas para os diversos comprimentos de onda λ.</i>	80
6.4	<i>Comparação entre as saídas pupilométricas para os diversos comprimentos de onda λ.</i>	82
6.5	<i>Comparação entre as saídas pupilométricas para os diversos comprimentos de onda λ.</i>	83
6.6	<i>Tratamento automático de oclusões.</i>	93
B.1	<i>Webcam Logitech.</i>	124

Lista de Tabelas

2.1	<i>Variação do diâmetro da pupila na ausência de luz.</i>	14
4.1	<i>Métodos de extração de formas.</i>	28
5.1	<i>Parâmetros medidos na pupilometria dinâmica.</i>	63
6.1	<i>Valores dos parâmetros encontrados em função da intensidade luminosa para LED branco.</i>	85
6.2	<i>Valores dos parâmetros encontrados em função da intensidade luminosa para LED vermelho.</i>	86
6.3	<i>Valores dos parâmetros encontrados em função da intensidade luminosa para LED verde.</i>	87
6.4	<i>Valores dos parâmetros encontrados em função da intensidade luminosa para LED azul.</i>	88
6.5	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	89
6.6	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	89
6.7	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	90
6.8	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	90
6.9	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	91
6.10	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	91
6.11	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	92
6.12	<i>Comparação dos resultados do sistema com os modelos de simulação pupilar.</i>	92

Lista de Códigos

A.1	<i>Controle PWM da fonte de iluminação.</i>	104
A.2	<i>Interface de controle para aquisição do vídeo.</i>	107
A.3	<i>Interface de controle para o rastreamento.</i>	113

Lista de Abreviações

1D	Uma dimensão
2D	Duas dimensões
3D	Três dimensões
CIE	Comissão internacional de iluminação
EW	Núcleo de neurônios Edinger-Westphal localizado no sistema nervoso central
FFT	Transformada rápida de Fourier (<i>Fast Fourier Transform</i>)
FPS	Quadros por segundo
HF	Alta frequência (<i>High frequency</i>)
LED	Diodo emissor de luz (light-emitting diode)
LF	Baixa frequência (<i>Low frequency</i>)
LUX	Unidade do SI para luminância
NPR	<i>Near pupil response</i>
PWM	Modulação por largura de pulso (<i>Pulse width modulation</i>)
SNA	Sistema Nervoso Autônomo
SNC	Sistema Nervoso Central
SNP	Sistema Nervoso Parassimpático
SNS	Sistema Nervoso Simpático

TCH	Transformada Circular de Hough
TF	Transformada de Fourier
TH	Transformada de Hough

INTRODUÇÃO

1.1 Introdução

A pupila realiza movimentos em diversas situações como, por exemplo, durante o ajuste de foco, devido ao reflexo à luz, em situações de medo, mudança no nível de atenção, entre outros [1]. Esses movimentos correspondem a contrações ou mioses e dilatações ou midríases.

Ocorre miose, por exemplo, no ajuste de foco, isto é, quando se observa um objeto distante e passa-se então a observar um objeto próximo. Nesse caso o diâmetro da pupila aumenta de forma involuntária por ação do sistema nervoso. Consequentemente ocorre midríase no processo inverso, ou seja, quando se observa um objeto próximo e passa-se então a observar um objeto distante.

Mais precisamente a pupila dilata pela ativação do sistema nervoso autônomo simpático (SNS) e contrai por ativação do sistema nervoso autônomo parassimpático (SNP) [1], [2]. O diagrama da Figura 1.1 representa a divisão anatômica do sistema nervoso. Ele se divide inicialmente em sistema nervoso central (SNC) e em sistema nervoso periférico (SNP).

O SNC é formado pelo encéfalo e pela medula espinhal cuja função é de receber, transmitir, elaborar e armazenar informações. Já o sistema nervoso periférico é formado pelos nervos e gânglios e é responsável pela ligação das diversas partes do corpo ao sistema nervoso central [3].

O sistema nervoso autônomo (SNA) que deriva do sistema nervoso periférico possui a função de regular as atividades involuntárias do nosso organismo, como por exemplo, pressão, temperatura e batimentos cardíacos. Ele por sua vez pode ser dividido em sistema nervoso autônomo simpático e parassimpático. O primeiro é utilizado quando o organismo requer respostas rápidas e o último nos casos opostos [3].

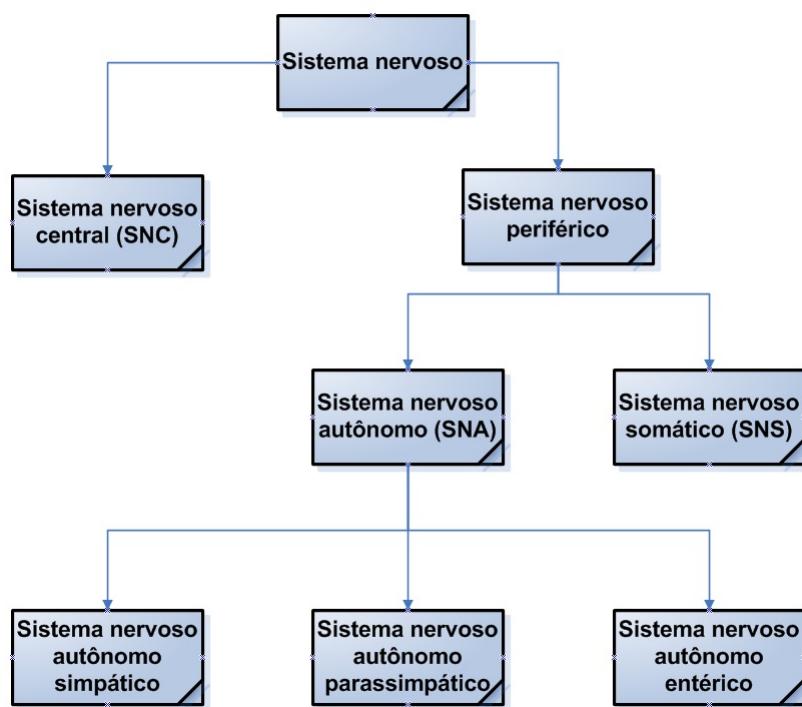


Figura 1.1: *Representação anatômica do sistema nervoso.*

Por isso, o mecanismo de contração e dilatação da pupila pode ser considerado um método não invasivo para avaliar a atividade do sistema nervoso autônomo [4].

A caracterização dos movimentos pupilares também encontra relevância nas cirurgias refrativas. Essas cirurgias são feitas para corrigir miopia, astigmatismo, presbiopia e hipermetropia. Nesses casos, para que ocorra a formação de imagens mais nítidas na região central da retina é necessário retirar camadas de células e moldar a córnea com o auxílio de um feixe de laser [5].

Conforme mostra a Figura 1.2, adaptada de [6], para níveis de iluminação fotópica (> 4 LUX) a pupila adquire forma com diâmetro pequeno (≈ 2 mm). Nesse caso, para um

feixe de laser ajustado em torno de 6 mm , por exemplo, conforme a Figura 1.2(a), maior que a região operada, toda a área de entrada de luz será corrigida [6].

Por outro lado, para níveis de iluminação escotópica e mesopica ($< 4\text{ LUX}$) os diâmetros pupilares podem chegar a 8 mm e portanto maior que a região do laser, de acordo com as Figuras 1.2(b) e (c). Em ambos os casos, regiões de entrada de luz não corrigidas podem influenciar na visão noturna do paciente [6].

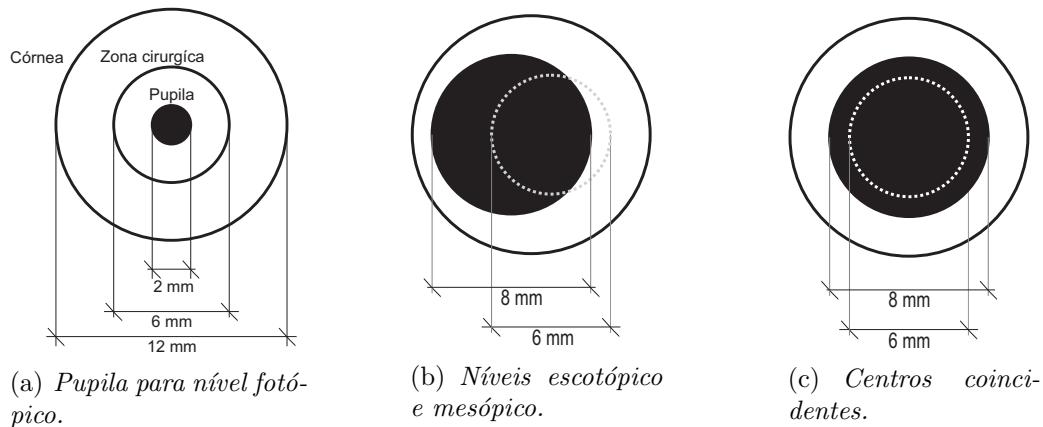


Figura 1.2: Diferentes tamanhos e posições de pupila podem e devem definir diferentes procedimentos cirúrgicos.

1.2 Objetivos deste Trabalho

Nesse âmbito, o objetivo deste trabalho é mostrar que a Transformada Circular de Hough pode ser utilizada como método de *Video Tracking* baseado em extração de contorno e que pode ser aplicada na pupilometria sem a necessidade de generalizar a forma da pupila e sem o tratamento de sobreposições de objetos.

Outro objetivo desta dissertação é desenvolver um dispositivo protótipo micro controlado automático capaz de rastrear a pupila, em vídeo, extrair suas principais medidas e relacionar seu movimento com a variação do comprimento de onda e da intensidade luminosa incidente. Esse dispositivo juntamente com sua interface pode ser utilizado em estudos clínicos relacionados com patologias do sistema visual e do sistema nervoso.

1.3 Estado da Arte

O pupilômetro é o equipamento utilizado para medir o diâmetro da pupila. Ele tem sido objeto de grande interesse nas cirurgias refrativas, pois o diâmetro da área a ser operada é planejado de acordo com a área da pupila do paciente [5].

Na pupilometria fotográfica pode-se adquirir uma imagem 3D do olho onde é mapeada toda a superfície anterior e posterior da córnea. É importante salientar que os aparelhos desse tipo são relativamente lentos, pois inicialmente eles adquirem a imagem e posteriormente efetuam o processamento em 3 dimensões.

Na pupilometria televisiva é utilizada iluminação infravermelha e câmeras sensíveis a essa radiação. Desse modo, as medidas são adquiridas em ambientes fotópicos onde a pupila não sofre influência da iluminação externa.

Para avaliar o comportamento da pupila e realizar diagnósticos de doenças relacionadas com sua variação de diâmetro utilizam-se estímulos luminosos com características variáveis como, por exemplo, o comprimento de onda, a intensidade luminosa e a taxa de repetição [7]. Nesses casos, quanto maior a resolução das imagens do pupilômetro mais variações serão detectadas.

Os pupilômetros de última geração utilizam computadores de alta velocidade de processamento, pois os sensores utilizados são de alta definição (cerca de $0,01\text{ mm}$ de resolução linear) podendo superar uma frequência de 50 Hz para a gravação dos dados [8].

Além disso, os algoritmos de tratamento, como os que utilizam as séries de Fourier, permitem uma aproximação aos pontos de fronteira da pupila e descreve a maneira correta de associá-los de forma não circular. Quanto maior a sensibilidade maior o custo computacional [9].

Desse modo, a tendência do mercado é apostar em equipamentos portáteis, de fácil utilização e rápidos, mesmo com a desvantagem em termos de resolução e refinamento dos algoritmos de tratamento [8].

1.4 Estrutura deste Trabalho

Esta dissertação está dividida em três partes: a primeira corresponde aos quatro primeiros capítulos, onde é apresentada a introdução do trabalho e a explicação dos conceitos teóricos necessários ao estudo. A segunda parte descreve o desenvolvimento de todo o sistema; e finalmente, a terceira compreende o estudo realizado com o sistema já desenvolvido, o tratamento e a discussão de resultados.

O Capítulo 1 apresenta introdução deste trabalho, os objetivos, o estado da arte da tecnologia envolvida e a estrutura desta dissertação. Finalmente são realizadas as considerações finais deste capítulo.

O Capítulo 2 mostra os conceitos do sistema visual humano com ênfase na pupila e por fim são realizadas as considerações finais deste capítulo.

Os Capítulos 3 e 4 descrevem as principais ferramentas do processamento digital de imagens e vídeo utilizadas no projeto. Destaca-se o rastreamento de objetos em vídeo (*video tracking*) e as técnicas de detecção de objetos em imagens digitais através da Transformada de Hough.

O Capítulo 5 descreve os principais modelos para simulação da dinâmica pupilar, o modelo de aquisição das imagens, o hardware desenvolvido, as etapas de processamento dos algoritmos implementados e as interfaces do software. Finalmente, são apresentadas as considerações finais deste capítulo.

O Capítulo 6 descreve os experimentos realizados e os resultados obtidos. Por conseguinte são explicados os motivos dos procedimentos tomados e os problemas encontrados ao longo do projeto.

O Capítulo 7 mostra as conclusões e contribuições desta dissertação além dos trabalhos futuros que poderão ser desenvolvidos.

Os apêndices mostram algumas especificações técnicas dos componentes utilizados no sistema proposto e as principais etapas do software desenvolvido.

1.5 Considerações Finais deste Capítulo

Este capítulo apresenta o objetivo deste trabalho, que é determinar a aplicabilidade da Transformada Circular de Hough como método de extração de contorno em *video tracking*. Outro objetivo é o desenvolvimento de um equipamento protótipo automático capaz de rastrear a pupila e estabelecer o seu comportamento em função do comprimento de onda e da intensidade de iluminação aplicada. O capítulo finaliza com o estado da arte da pupilometria e a estrutura desta dissertação.

O próximo capítulo mostra os conceitos do sistema visual humano com ênfase na pupila e por fim, são realizadas as considerações finais do capítulo.

MOVIMENTOS PUPILARES

2.1 Introdução

Conforme descrito em [10], vários fatores podem alterar ou iniciar o processo de movimentação da pupila. Basicamente esses movimentos são: (i) a acomodação, que ocorre devido ao processo de foco, (ii) o reflexo pupilar a luz (*Pupil Light Reflex* - PLR), que é o reflexo da pupila para adaptar o sistema visual a iluminação ambiente e (iii) o *hippus*, uma oscilação que ocorre constantemente como parte de um processo de convergência disparado pela acomodação e pelo PLR.

Além disso, encontra-se na literatura vários fatores que podem afetar a dinâmica dos movimentos pupilares, como por exemplo, o uso de drogas, doenças, fatores emocionais, respiração e batidas do coração, interesse e curiosidade, idade, cor da íris, nível de consciência, o local onde um raio de luz atinge o cristalino, (na borda ou no centro), comprimento de onda da luz incidente e os padrões espaciais.

Ainda, conforme [10], o processo de acomodação/desacomodação é devido ao relaxamento/contração do músculo ciliar, mostrado na Figura 2.1, adaptada de [11]. Para o ajuste de foco (*depth of field*) o músculo ciliar pode alterar a forma da pupila pela mudança de pressão na câmara posterior. Como ocorre o processo de acomodação como um todo ainda é um mistério para a ciência, mas existem quatro teorias que tentam explicar essa ação:

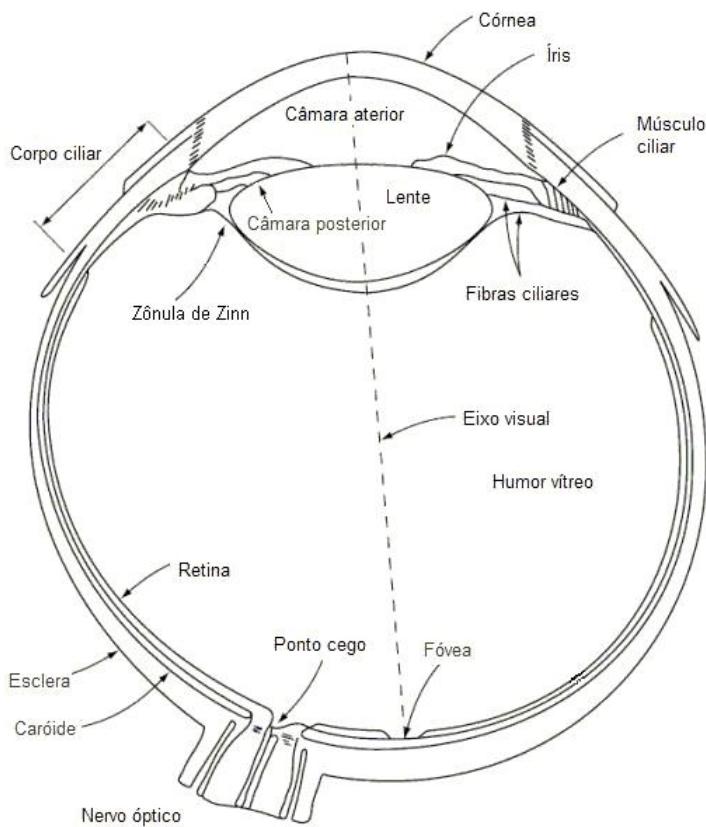


Figura 2.1: *Diagrama simplificado de um corte transversal do olho humano.*

- a de Hermann von Helmholtz, a mais famosa e aceita teoria para os processos de acomodação, mostra que quando uma pessoa olha para um objeto distante, os músculos ciliares relaxam puxando as lentes e tornando-as mais finas. Quando a pessoa olha para um objeto próximo, os músculos ciliares contraem-se tornando a lente mais convexa [12];
- a de William Bates mostra que a acomodação é proveniente da ação de dois dos cinco músculos extraoculares: o superior oblíquo e o inferior oblíquo [13];
- a de Ronald Schachar mostra o oposto da ideia de Helmholtz, quando o músculo ciliar puxa a lente, somente a parte central torna-se mais convexa, enquanto que a parte periférica torna-se mais plana [14];
- a de D. Jackson Coleman propõe que as lentes, o zonule of Zinn e a câmara anterior

comprimem um diafragma entre a camada anterior e o vítreo do olho. Em outras palavras, o músculo não altera diretamente a forma da lente, mas sim a pressão intraocular [15].

Como já citado no Capítulo 1 deste trabalho, a pupila controla a quantidade de luz que entra nos olhos protegendo assim, a retina de uma carga muito intensa de luz. O PLR é um processo ligado à intensidade de luz incidente na retina e pode ser influenciado por fatores como: o uso de medicamentos, o estado emocional do indivíduo, o contraste da imagem percebida, além de produzir resultados diferentes de acordo com o comprimento de onda.

Uma evidência percebida pelo do PLR é que o reflexo é síncrono entre as duas pupilas. A falta desse sincronismo é devido a algum mau funcionamento do sistema visual humano. Na literatura esse sincronismo é chamado de reflexo consensual [16].

O PLR é um exemplo claro de homeostase em seres humanos, ou seja, um mecanismo dinâmico de equilíbrio controlado por sistemas de regulação inter-relacionados. Esse processo visa estabelecer o equilíbrio do sistema contrariando qualquer mudança com a finalidade de manter a capacidade do ser humano de adaptação ao meio. Ele modela as ações do corpo em dois passos distintos: percepção e ajuste. Na íris, a percepção ocorre quando a luz chega à retina e a informação é enviada ao cérebro, e o ajuste quando o cérebro envia sinais para fechar ou abrir a pupila.

As pupilas estão em constante movimento de contração e relaxamento, o movimento pulsátil denominado *hippus*, é determinado pelo balanço entre o sistema nervoso simpático e o parassimpático [2]. Ao longo da adaptação ao estímulo e mesmo em uma iluminação estável ou na escuridão total ocorre uma variação de cerca de 0,2 Hz [8]. Estas variações possuem uma sincronia com a respiração e com o ritmo regular circulatório e neural. O ritmo da respiração (*High Frequency* - HF) é a marcação para o nervo parassimpático, enquanto os batimentos cardíacos possuem uma relação maior com o nervo simpático (*Low Frequency* - LF) [10].

Uma variação anormal na acomodação é resultado de algum problema no sistema nervoso, que pode ser ocasionado por alcoolismo, diabetes, AIDS, síndrome de down, depres-

são, mal de Alzheimer, hipertensão intercranial, medo e assim por diante. O PLR e o hippus são um processo automático, uma reação reflexa, ou seja, diferente da acomodação, eles não são comandado pelo cérebro, mas sim pelo sistema nervoso autônomo (SNA) [10].

2.2 Midríase: dilatação pupilar

De forma geral, durante a midríase, o esfíncter da íris relaxa e o músculo dilatador da íris contrai.

Na Figura 2.2, extraída de [17], observa-se um ciclo pupilar. Pode-se notar um declive menos acentuado da curva de relaxação pupilar, comparativamente ao declive da curva de contração pupilar. Esse fenômeno ocorre devido ao fato de o esfíncter da íris ser mais forte que o músculo dilatador da íris, logo a dilatação da pupila não ocorre imediatamente após o relaxamento do esfíncter.

Por meio da inibição do núcleo de neurônios Edinger-Westphal, localizado no sistema nervoso central, ocorre a relaxação do esfíncter da íris, conforme ilustra a Figura 2.3. Esses processos inibitórios são desencadeados pelos neurônios do sistema nervoso simpático [2], [14]. Assim, quando o SNS está atuando sobre o Núcleo de neurônios EW seu sinal parassimpático é suprimido resultando numa relaxação relativa do esfíncter da íris e, consequentemente, na dilatação da pupila. Para o caso de ausência do sinal inibitório os neurônios preganglionicos atuam a altas frequências causando contração. Essa situação pode ser observada, por exemplo, durante o sono, sob efeito de narcóticos ou sob efeito de outros fatores já mencionados.

No entanto, mesmo durante o processo de midríase, se um estímulo na forma de luz intensa incidir na pupila, um conjunto de impulsos neuronais chega imediatamente ao núcleo Edinger-Westphal vindo da retina e sobrepoê-se ao domínio da inibição do núcleo, causando contração [8]. A atividade nervosa simpática atua como um catalizador da dilatação pupilar, porém não é o único requisito para que ocorra midríase [2], [8].

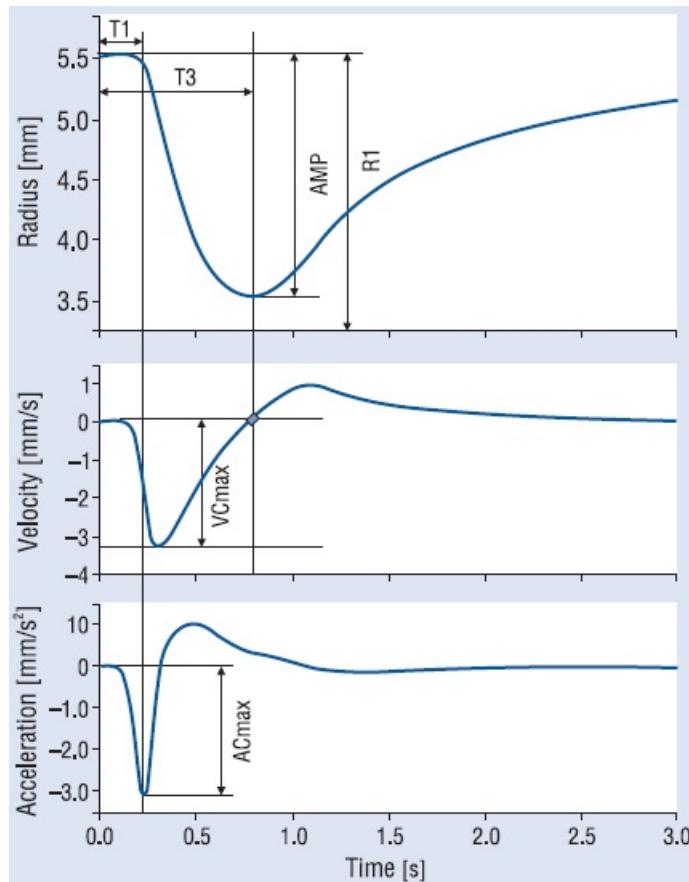


Figura 2.2: Parâmetros medidos em um ciclo pupilar: latência (T_1), máxima velocidade de constrição (VC_{max}), tempo para a máxima constrição (T_3), amplitude (AMP), raio inicial (R_1), raio mínimo (R_2) e máxima aceleração de constrição(AC_{max}).

2.3 Miose: contração pupilar

Quando as fibras musculares do esfíncter pupilar contraem ocorre miose. No reflexo pupilar à luz os axônios das células ganglionares da retina de cada olho transmitem a informação da luz que percorre as sinapses neuronais para os neurônios localizados no núcleo pretectal, na região central do cérebro e por fim ao núcleo Edinger-Westphal esquerdo e direito, conforme a Figura 2.3, adaptada de [18].

Por sua vez, os neurônios dos núcleos Edinger-Westphal enviam os seus axônios parassimpáticos ao longo dos nervos oculomotores, onde se relacionam com os gânglios ciliares através de sinapse. Os neurônios dos gânglios ciliares emitem axônios parassimpáticos

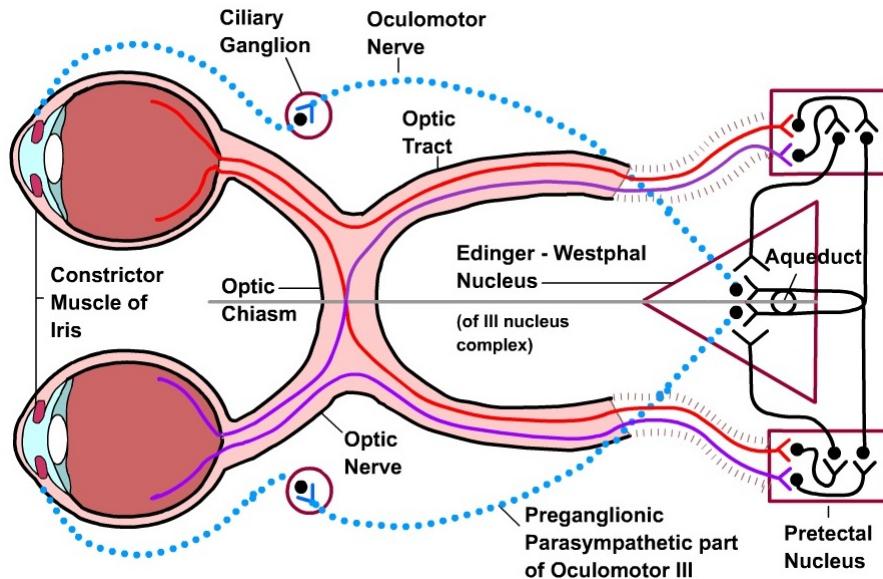


Figura 2.3: *Percorso nervoso visual.*

que percorrem os nervos ciliares até ao globo ocular, onde se relacionam com o músculo esfíncter da íris por sinapse, conforme a Figura 2.3.

Normalmente a miose ocorre devido ao reflexo à luz (PLR) e também na visão de um objeto próximo, (*Near Pupil Response - NPR*). Fisiologicamente as duas respostas são diferentes apenas em suas divisões aferentes, pois para o *NPR* ocorre a ativação do tronco cerebral rostral além do envolvimento do núcleo EW [2], [8].

Os fotorreceptores também contribuem com os reflexos pupilares. Os cones, altamente sensíveis às informações de cores, tornam-se dominantes no controle das respostas pupilares em ambientes iluminados. Por outro lado, os bastonetes não estão envolvidos na visão em cores e são sensíveis em ambientes com baixa iluminação [11].

Dessa forma, os bastonetes são responsáveis pela capacidade que a pupila tem em exibir pequenas contrações de baixa amplitude, como resposta a luz de baixa intensidade em ambiente de adaptação ao escuro (alta sensibilidade pupilar). Por outro lado, os cones são responsáveis pelas contrações pupilares de maior amplitude (visíveis a olho nu) [8].

2.4 Pupilometria

Um ciclo pupilar pode ser alcançado por meio do movimento de contração da pupila seguido por uma expansão e finalmente retornando ao estado inicial. Existem outros tipos de simulações, por exemplo, o *pupil escape* é o nome considerado quando uma pupila dilatada sofre a ação de um estímulo de luz e contrai e, após alguns segundos ou minutos, retorna ao seu estado inicial. Quando a intensidade de luz é muito grande e a pupila não retorna ao seu estado inicial, denomina-se *pupil capture* [10].

A pupilometria mede diversos componentes: amplitude máxima, latência, velocidade de dilatação e contração e tamanho máximo e mínimo. A amplitude corresponde à diferença entre o tamanho inicial e o mínimo da pupila durante o PLR. Sabe-se que o cristalino deixa de ser totalmente transparente com a idade, reduzindo a quantidade de luz que chega à retina. Como a amplitude é proporcional ao logaritmo de todo o fluxo de luz que chega na retina, ela acaba tornando-se, também, dependente da idade do indivíduo. Essa perda é estimada em cerca de 0,4 mm por década a partir dos 20 anos [10].

Smith e Dewhirst [19] estudaram a relação da idade com as dimensões pupilares e realizaram testes em 163 indivíduos com idades entre 15 e 92 anos. A Tabela 2.1, adaptada de [19], mostra os resultados obtidos.

Latência é o tempo que a pupila leva para reagir a um estímulo de luz. Em observações clínicas, a latência pode identificar objetivamente os atrasos no processamento visual, que são proporcionais ao dano causado ao sistema visual. A latência comparada a amplitude é menos volátil e é menos afetada pelas propriedades mecânicas da íris, no entanto ela ainda é afetada. Para definir matematicamente o instante que a latência termina, utiliza-se a segunda derivada da função de tamanho da pupila em relação ao tempo. Durante a maior inclinação no gráfico de velocidade (1^{a} derivada) existe um canal na função de aceleração (2^{a} derivada). O pico desse canal é o ponto onde a latência termina. Existem evidências que a latência e a amplitude são governadas pela ativação parassimpática junto com a velocidade de contração. A velocidade de dilatação é governada pela ativação simpática.

Assim como a amplitude e a latência, o tamanho da pupila, em estado de descanso,

Tabela 2.1: *Variação do diâmetro da pupila na ausência de luz.*

Variação de idade	Média	Diâmetro da pupila(mm)			Diâmetro da pupila(%)		
(Idade)	(Idade)	Mín	Esperado	Máx	Mín	Esperado	Máx
15-19	17	6,0	7,4	8,8	52,0	63,3	74,6
20-24	22	5,8	7,2	8,6	50,5	61,8	73,1
25-29	27	5,6	7,0	8,4	49,0	60,3	71,5
30-34	32	5,4	6,8	8,2	47,5	58,8	70,0
35-39	37	5,2	6,6	8,0	46,0	57,3	68,5
40-44	42	5,0	6,4	7,8	44,5	55,8	67,0
45-49	47	4,8	6,2	7,6	43,0	54,3	65,5
50-54	52	4,6	6,0	7,4	41,5	52,7	64,0
55-59	57	4,4	5,8	7,2	40,0	51,2	62,5
60-64	62	4,2	5,6	7,0	38,5	49,7	61,0
65-69	67	4,0	5,4	6,8	37,0	48,2	59,5
70-74	72	3,8	5,2	6,6	35,4	46,7	58,0

é maior para pessoas mais velhas. A velocidade máxima de dilatação e de contração é menor nessas pessoas (o que indica perda da força de atuação dos nervos simpáticos e parassimpáticos). A ciência ainda não determinou o que faz essas alterações ocorrerem, mas como o cristalino torna-se menos transparente com a idade, obviamente, ele deixa passar menos luz para a retina, e assim, menos contração. Uma das hipóteses seria o aumento da influência parassimpática ou a perda da influência simpática. É importante salientar também, que o tamanho máximo da pupila não depende do tamanho da íris, mas sim da sua inervação [10].

Atualmente existem vários modelos que simulam a dinâmica da pupila e vários estudos que determinam seus parâmetros para o propósito de pesquisa e da prática clínica. Alguns desses modelos e parâmetros serão apresentados no Capítulo 5 deste trabalho com o propósito de testar e validar a saída pupilométrica do sistema proposto.

2.5 Considerações Finais deste Capítulo

Este capítulo apresenta os principais movimentos realizados pela pupila e os fatores que podem influenciar sua dinâmica. Descreve sucintamente os movimentos de acomodação, PLR e hippus e os processos de miose e midríase além de suas relações com o sistema nervoso.

O próximo capítulo apresenta um panorama das principais abordagens de rastreamento de objetos (*tracking*). Essa ferramenta é a base desta dissertação a medida que o processamento só ocorre após a identificação do alvo em cada *frame* do *video stream*.

VIDEO TRACKING

3.1 Introdução

Video tracking têm recebido grande atenção nos últimos anos, principalmente devido a variedade de aplicações possíveis, como por exemplo, interfaces homem máquina, sistemas de segurança, realidade virtual e codificação de vídeo. Qualquer que seja a aplicação o resultado final depende da representação utilizada para descrever o alvo rastreado, podendo nesse caso ser um contorno do objeto, um conjunto de coordenadas 2D do seu centro de massa, sua posição 3D e assim por diante.

Alguns sistemas de *tracking* utilizam sensores ou transmissores para marcar o objeto a ser rastreado e são chamados de *Active Object Tracking*. No entanto, estes são considerados intrusivos e adequados para ambientes controlados o que faz do rastreamento passivo mais desejável, apesar deste ser bem mais difícil de promover.

No *tracking* passivo, objeto deste trabalho, alguns critérios de representação podem ser utilizados para a detecção - forma, posição, cor, textura e assim por diante.

O desafio torna-se ainda maior nos casos em que o objeto a ser rastreado possui estrutura deformável e padrões de cor variáveis, como no caso da íris. Além disso, o movimento da íris pode ser afetado por fatores inesperados que usualmente impossibilitam qualquer tipo de predição a partir de imagens. A íris reage muito rapidamente a variações na iluminação (na ordem de milissegundos para a contração) e sua cor é resultante da diferente absorção

da luz que incide sobre as células pigmentadas de sua superfície anterior, onde os padrões são tomados de forma aleatória e não estão relacionados com nenhum fator genético.

3.2 Rastreamento de Objetos Rígidos

O rastreamento de objetos 2D visa recuperar o movimento no plano de projeção da imagem do objeto que em geral move no espaço 3D. Por outro lado, o rastreamento de objetos 3D visa estimar o movimento real em 3D do objeto a partir de informações em 2D de uma ou mais câmeras.

A estrutura do objeto a ser rastreado é outra característica que afeta o tipo de movimento que precisa ser estimado. Nesse caso, dependendo do tipo de objeto, pode-se ter o rastreamento de objetos rígidos ou deformáveis/articulados.

3.2.1 Rastreamento de objetos rígidos 2D

O rastreamento de objetos rígidos em duas dimensões (2D) busca estimar a trajetória do objeto no plano da imagem à medida que ele se move na cena. A Figura 3.1 resume as principais etapas do sistema de rastreamento utilizado neste trabalho.

Segundo [20], os métodos de rastreamento de objetos rígidos em 2D constituem blocos básicos para outras categorias de algoritmos de rastreamento e podem ser classificados em diferentes categorias, de acordo com as ferramentas utilizadas:

- métodos baseado em regiões;
- métodos baseado em contornos;
- métodos baseado em pontos característicos;
- métodos baseado em modelos.

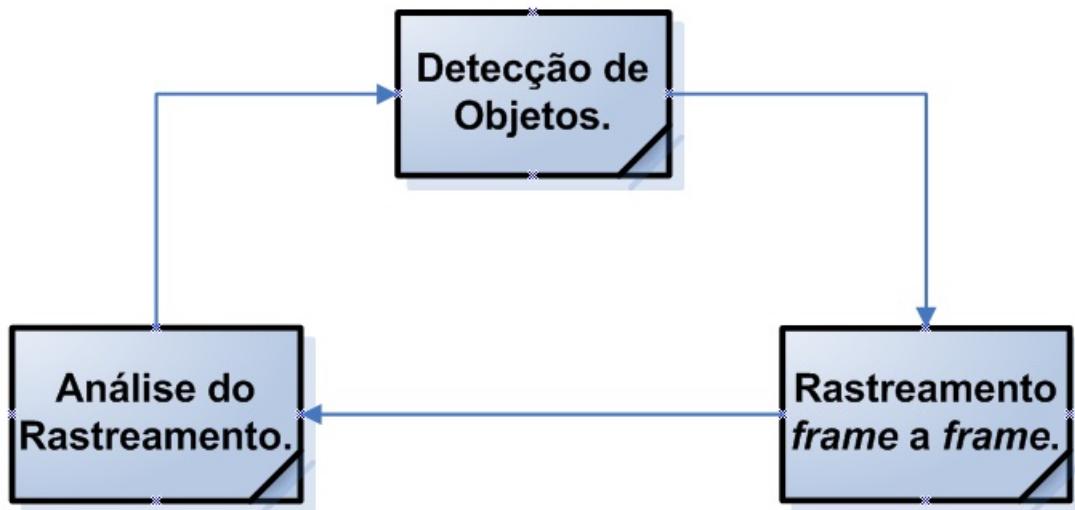


Figura 3.1: *Etapas do rastreamento de objetos.*

3.2.1.1 Rastreamento de Objetos Baseado em Regiões

O rastreamento de objetos baseado em regiões é geralmente uma maneira eficiente de interpretar e analisar o movimento observado em uma sequência de vídeo. Uma região da imagem pode ser definida como um conjunto de pixels com características homogêneas. Ela pode ser obtida por segmentação de imagens, podendo ser baseada em diferentes características do objeto (cor, textura, bordas) e/ou sobre o movimento observado nos quadros da sequência de vídeo. Essencialmente, a região seria a área da imagem coberta pela projeção do objeto de interesse no plano da imagem. Alternativamente, uma região pode ser a “caixa” que delimita o objeto sob exame [21], [22].

A informação de cor mostra ser uma alternativa efetiva para o rastreamento de objetos baseado em região de interesse por tornar o processamento mais rápido em tempo real, isto é, 20 a 30 *frames* por segundo (FPS) [23].

O maior problema da segmentação e do *tracking* baseados em cor refere-se à sua eficiência contra variações de iluminação principalmente em cenas ao ar livre [23].

Um modo de se obter certo grau de invariabilidade de iluminação é usando apenas os valores de cromaticidade do espaço de cor HSV (*Hue-Saturation-Value*), ou seja, utilizar apenas as componentes H e S.

A Figura 3.2, adaptada de [24], ilustra uma segmentação a partir da modelagem eficiente da cor do objeto. A modelagem correta permite que o objeto seja diferenciado de outros objetos presentes na cena e também do fundo da imagem.



(a) *Imagen original.*



(b) *Imagen segmentada.*

Figura 3.2: *Exemplo de rastreamento baseado em espaço de cor.*

Um algoritmo de segmentação de cor pode ser concebido em três etapas: a escolha de um espaço de cor adequado, a modelagem do objeto sobre o espaço de cor selecionado, e o método utilizado para classificar os pixels individuais do objeto e do não-objeto. Os métodos de modelagem de cor podem ser grosseiramente classificados como paramétricos (usando uma única Gaussiana ou uma mistura de Gaussianas) e não paramétrico (baseado em histograma, tais como as tabelas de pesquisa e os mapas de probabilidade de Bayes) [20], [23], [25].

3.2.1.2 Rastreamento de Objetos Baseado em Contorno

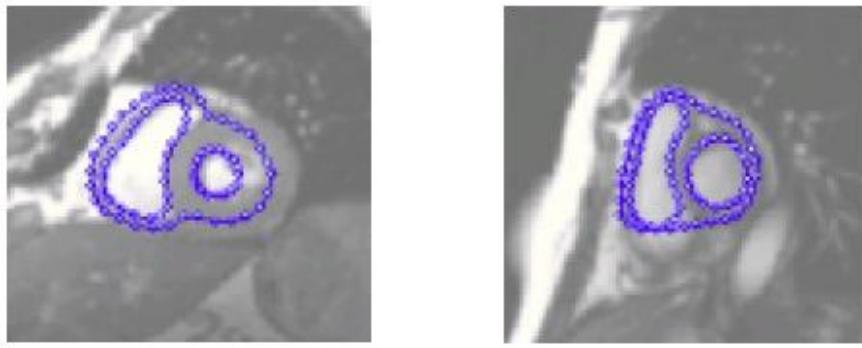
Diversos trabalhos utilizam as informações de contorno para rastrear objetos ao longo do tempo e recuperar as informações de posição e forma do objeto ao longo da sequência de vídeo [26], [27].

Geralmente as técnicas baseadas em contorno envolvem cálculos de distâncias, logo tais métodos de modelagem são mais complicados do que a modelagem por regiões, por exemplo, usando cores. No entanto, o rastreamento por contorno geralmente é mais robusto do que algoritmos baseados em rastreamento por região, pois ele pode ser adaptado para lidar com oclusão parcial e informações de contorno e é, na maioria dos casos, mais insensível à

variação de iluminação [23].

Contornos ativos, também conhecido como *snakes*, têm sido amplamente utilizados por pesquisadores para realizar a segmentação de objetos e monitoramento. Um algoritmo de contorno ativo deforma dinamicamente o contorno para “casar” com as características da imagem tais como linhas, bordas, limites, e assim por diante. *Snakes* consistem de curvas paramétricas elásticas cuja deformação é sujeita a forças internas (contorno de forças elásticas) e forças externas (devido ao conteúdo da imagem e outras restrições) [20].

A Figura 3.3, adaptada de [28], ilustra a utilização de uma *Snake* em um procedimento de segmentação em ressonância magnética cardíaca.



(a) Segmentação em ressonância magnética cardíaca.
 (b) Ajuste do contorno (snake) à borda.

Figura 3.3: Exemplo de rastreamento baseado em contorno ativo.

3.2.1.3 Rastreamento de Objetos Baseado em Pontos Característicos

Conforme [20], o rastreamento baseado em pontos característicos pode ser definido como a tentativa de recuperar os parâmetros de movimento de um ponto em uma sequência de vídeo, mais precisamente os parâmetros associados com a translação planar do ponto, pois pontos no espaço 2D não giram nem transladam com respeito à profundidade. Mais formalmente, $A = A_0, A_1, \dots, A_{n-1}$ denota os N quadros em uma sequência de vídeo e $\mathbf{m}_i(x_i, y_i), i = 0 \dots N - 1$ é a posição do ponto característico nesses quadros. A tarefa é determinar um vetor de movimento $\mathbf{d}_i(d_{x,i}, d_{y,i})$ que melhor determina a posição do ponto

característico no próximo *frame*, $\mathbf{m}_{i+1}(x_{i+1}, y_{i+1})$ que é $\mathbf{m}_{i+1} = \mathbf{m}_i + \mathbf{d}_i$. O objeto a ser rastreado é geralmente definido por uma “caixa” delimitadora ou um envoltório convexo dos pontos rastreados, conforme a Figura 3.4, adaptada de [29].

Um dos problemas típicos encontrado em rastreamentos baseado em pontos característicos pode ser a detecção de pontos atípicos individuais, contudo algoritmos para essa abordagem podem ser construídos de forma muito eficiente se for considerado pontos com características salientes [30].

Esses pontos podem ser, por exemplo, aqueles associados com informações de alta curvatura local, isto é, cantos e bordas. A invariância da vizinhança do ponto (ou descriptor correspondente) às variações de iluminação e alterações de ponto de vista corresponde a outra característica desejável. Muitos pontos característicos (também chamados de pontos chave) têm sido propostos na literatura e para maiores detalhes [20], [29].

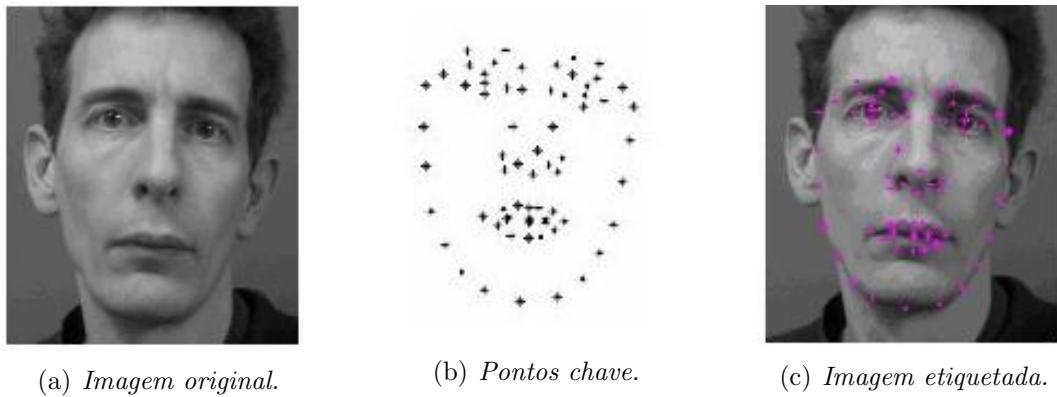


Figura 3.4: Exemplo de rastreamento baseado em pontos característicos.

3.2.1.4 Rastreamento de Objetos Baseado em Modelos

Vários pesquisadores utilizam técnicas baseadas em modelos para rastreamento de objetos em 2D [20], [25], [31]. Esse tipo abordagem está relacionado com o rastreamento baseado em região porque um modelo é essencialmente uma forma da região a ser rastreada na imagem.

Por razões práticas é necessário conhecer o modelo a ser buscado nas imagens (etapa de

inicialização). Por exemplo, os modelos podem ser específicos, ou seja, uma instância de uma classe de objetos ou podem ser um modelo estatístico extraído de uma base de dados.

A correspondência de modelo *Template Matching* pode ser definida como o processo de busca da imagem alvo para determinar a região da imagem que se assemelha ao modelo, baseado em uma medida de similaridade ou distância. Essencialmente, a região do modelo deve passar por uma transformação geométrica que minimizaria a distância da medida usada no processo de comparação. O objetivo de um algoritmo de correspondência de modelo é estimar os parâmetros de tal transformação *off-line* [20], [31].

Algumas técnicas do *Template matching* serão detalhadas no Capítulo 4 desta dissertação. Existem, obviamente, outros modelos para o rastreamento de objetos, por exemplo, o rastreamento Bayesiano, filtros de Kalman, filtros estendidos de Kalman e particle filters.

3.2.2 Tratamento de Oclusões

Um dos maiores desafios dos algoritmos de rastreamento de objetos em sequências de imagens é a oclusão ou auto-oclusão do objeto [32]. Uma oclusão ocorre quando parte ou até mesmo todo o objeto rastreado não é visível em todos os *frames* da sequência. Por exemplo, essa situação pode ocorrer devido a presença de um ou mais objetos estáticos na cena ou devido a objetos com movimentos relativos opostos.

Dependendo da aplicação um sistema de rastreamento pode ignorar o tratamento de oclusões ou simplesmente tratá-lo como ruído no processo correspondente. No entanto, uma série de abordagens tenta lidar com oclusões de forma mais avançada.

Existem várias maneiras de lidar com as oclusões. Por exemplo, a utilização de sistemas de rastreamento redundantes que reinicia o processo de busca do objeto sempre que este estiver ocluído. Outra estratégia pode ser a divisão do objeto de interesse em partes menores e rastreá-las. Obviamente nesses sistemas o esforço computacional é maior.

Outros métodos podem ser empregados, como por exemplo, os métodos de filtragem estocástica, métodos baseados em pontos característicos como o *Corner finding* e métodos baseados em estimativas de velocidades como o *Optical flow* [33].

Com relação a sistemas de rastreamento com múltiplas câmeras o problema da oclusão se torna mais simples, pois um objeto ocluído em uma visão pode ser totalmente visível em outra. Esses sistemas combinam as informações de todas as câmeras ao determinar a “melhor vista”. Quando uma câmera perde o alvo devido à oclusão, a informação é obtida a partir de outras câmeras de monitoramento que também estão rastreando o objeto [20].

3.2.3 Rastreamento de Objetos Rígidos 3D

O rastreamento de objetos rígidos 3D pode ser definido como o processo de determinação da posição de um objeto rígido no espaço 3D em relação a um sistema de coordenadas e sua orientação em relação ao mundo de coordenadas do sistema [34]. As imagens podem ser obtidas de uma ou mais câmeras de vídeo.

Dessa forma, no rastreamento de objetos rígidos 3D é preciso estimar um total de seis parâmetros, embora em certas aplicações, determina-se apenas a posição (ou seja, considerando o objeto como um ponto de massa) ou apenas os parâmetros de orientação [35].

Uma das mais importantes aplicações de rastreamento de objetos rígidos 3D é o monitoramento de cabeça 3D (estimação de pose), que é usada como uma etapa de pré-processamento ou bloco de construção em reconhecimento e verificação facial, análise de expressão facial, animação avatar, interação homem maquina, e modelo baseado em sistemas de codificação.

Um número significativo de técnicas para rastreamento de objetos rígidos em 3D é baseado em modelos, ou seja, envolvem o modelamento da geometria 3D (geralmente enriquecido com informações de textura) do objeto de interesse ao derivar informações 3D a partir de projeções 2D do objeto sobre a sequência de vídeo [20], [25].

Certos métodos utilizam técnicas de rastreamento 2D para obter o movimento 2D no plano da imagem do objeto de interesse e, em seguida, um filtro de Kalman para obter os parâmetros de movimento 3D. Outra abordagem para a estimativa de pose (combinação da posição e orientação do objeto) emprega o rastreamento 2D de importantes características

faciais (canto dos olhos e nariz). A variância das relações projetivas cruzadas dos cantos dos olhos e estatísticas antropométricas são posteriormente usadas para calcular a orientação em relação ao plano da câmera [36].

3.3 Rastreamento de Objetos Articulados

Normalmente os objetos articulados são representados por estruturas compostas de partes rígidas (*links*) ligadas por articulações de forma hierárquica, conforme a Figura 3.5, adaptada de [20]. É comum representar seres humanos ou animais dessa forma para extrair das imagens informações de alto nível, como por exemplo, reconhecimento de gestos e a compreensão comportamental animal. O rastreamento preciso das estruturas articuladas correspondentes em 3D se faz necessário. Portanto, a maioria dos algoritmos de rastreamento articulado monitora o objeto em um espaço 3D. Além disso, mesmo se o objetivo é rastrear um objeto articulado em 2D (ou seja, nos quadros de uma sequência de vídeo), a metodologia é semelhante ao do rastreamento articulado em 3D [20].

3.3.1 Rastreamento de Objetos Articulados 3D

As abordagens de rastreamento de objetos articulados 3D podem ser livres de modelo ou baseado em modelos. No primeiro caso uma abordagem *bottom-up* é utilizada para combinar informações extraídas localmente da imagem (bordas, cantos), para criar estruturas coerentes, como os membros do corpo humano. Obviamente, essa abordagem exige que as estruturas reconstruídas sejam constantemente visíveis nas imagens.

Na abordagem baseada em modelo é empregado um modelo do objeto articulado. A complexidade do modelo depende da precisão necessária para uma aplicação específica. O corpo humano, por exemplo, é representado por partes rígidas (membros semelhantes) conectado uns aos outros nas articulações.

O aprofundamento sobre o rastreamento de objetos 3D articulados está além do escopo

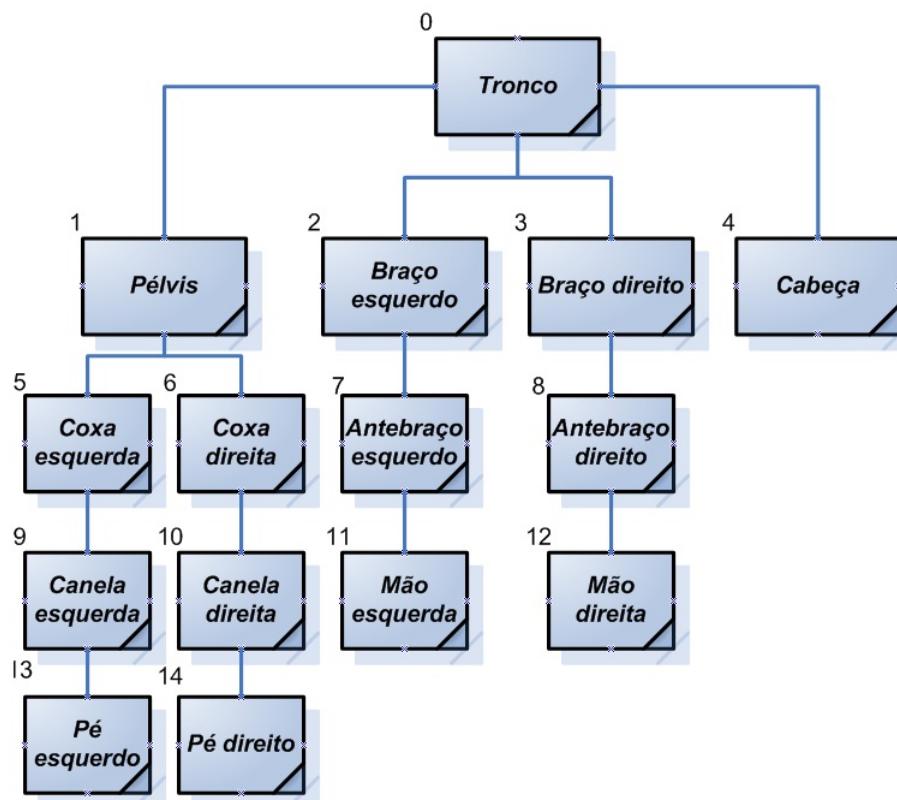


Figura 3.5: *Exemplo de representação hierárquica do corpo humano.*

deste trabalho. Pode-se encontrar maiores detalhes sobre essa abordagem, como por exemplo, algoritmos, modelamento 3D de objetos e tratamento de restrições de movimento em [20], [25], [35], [36].

3.4 Considerações Finais

Este capítulo apresenta os principais conceitos do rastreamento de objetos rígidos 2D e 3D e o tratamento de oclusões. Finaliza apresentando, de forma sucinta, alguns métodos do rastreamento de objetos articulados 3D que também engloba o rastreamento de objetos articulados 2D.

O próximo capítulo mostra algumas técnicas de extração de características em imagens digitais, como por exemplo, a Transformada Circular de Hough, utilizada nesse trabalho

como ferramenta do rastreamento baseado em contorno.

EXTRAÇÃO DE CARACTERÍSTICA EM IMAGENS DIGITAIS POR MEIO DA CORRESPONDÊNCIA DE FORMA

4.1 Introdução

A extração de características de alto nível diz respeito a encontrar formas em imagens digitais. Para ser capaz de reconhecer rostos automaticamente, por exemplo, uma abordagem possível é por meio da extração de componentes característicos. Isto requer a extração de componentes como os olhos, a boca, as orelhas e o nariz que são as características principais de um rosto.

Para encontrá-las pode-se utilizar a forma; boca e sobrancelhas podem ser aproximadas por duas linhas enquanto os olhos por formas elipsoides e assim por diante. A extração de formas implica em encontrar a posição, a orientação e o tamanho. Esse processo assemelha-se à maneira como o ser humano percebe o mundo. Nesse processo as imagens complexas podem ser decompostas em estruturas com formas mais simples.

Na extração de características, geralmente procura-se formas com propriedades invariantes, ou seja, independente das condições de escolha. Essas técnicas devem encontrar

¹Extraído de: *Feature extraction and image processing* - Nixon, M.S. and Aguado, A.S. 2008, Academic Press.

Tabela 4.1: *Métodos de extração de formas.*

Método de extração de forma	Técnica
Brilho do pixel	Limiar de imagem Subtração de imagem
<i>Template matching</i>	Modelo de intensidade
Domínios de imagem e Fourier	Modelos binários
	Transformada de Hough
	Linhas
	Formas quadráticas
	Invariante
	Formas arbitrárias

formas confiáveis, de maneira robusta e independente dos parâmetros que controlam a aparência de uma forma, além de eliminar, em princípio, as mudanças no nível de iluminação.

Desde que ocorra contraste entre uma forma e seu fundo (*background*) pode-se dizer que a forma existe e pode ser detectada. Satisfazendo a condição de iluminação o próximo parâmetro de relevância é a posição. A forma deve ser encontrada onde quer que ela apareça – *position-invariance*, e muitas vezes independente de sua rotação – *rotation-invariance* (assumindo que o objeto ou a câmera tenha uma orientação desconhecida).

Alcançada essas duas propriedades de invariância pode-se então detectar um objeto de qualquer tamanho, passível de sofrer mudanças físicas ou variação na distância entre ele e a câmera. Contudo, além disso, devem ser considerados os métodos auxiliares para a eliminação de ruídos, a eliminação de imagens redundantes e as oclusões devem ser considerados.

Após a detecção do objeto segue sua extração que é a etapa mais complexa. A Tabela 4.1, extraída de [25], mostra algumas técnicas que serão descritas a seguir.

4.2 Limiarização (*Thresholding*) e *Subtraction*

A limiarização é uma das técnicas mais simples e consiste na classificação dos pixels de uma imagem de acordo com a especificação de um ou mais limiares. Considerando que os pixels do objeto e os pixels do fundo tenham seus níveis de cinza separados em dois grupos dominantes uma maneira óbvia de extrair o objeto do fundo é por meio da seleção de um

limiar τ que separe esses dois grupos.

A limiarização é claramente sensível a variações no nível de iluminação. A menos que o limiar possa se adaptar qualquer técnica falhará. A limiarização após a normalização do histograma da imagem torna o processo menos sensível a ruído, porém ainda sensível a variações na iluminação. Outra grande dificuldade que se aplica a técnica de limiarização refere-se à existência de mais de uma forma na imagem original, podendo ocorrer oclusão entre elas ou não. Sua atração é a simplicidade, pois não exige muito esforço computacional.

Uma abordagem alternativa é a subtração do fundo antes da limiarização. Isso pressupõe que o fundo seja conhecido com precisão, caso contrário, muitos detalhes aparecerão na imagem resultante. Além disso, a subtração é inviável na presença de ruído. A Figura 4.1, adaptada de [25], ilustra essa abordagem.

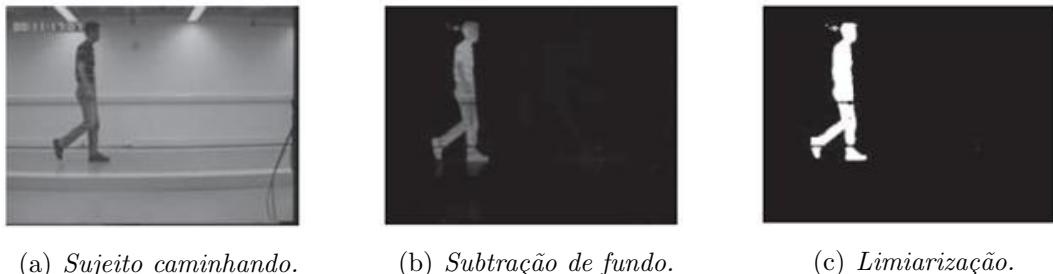
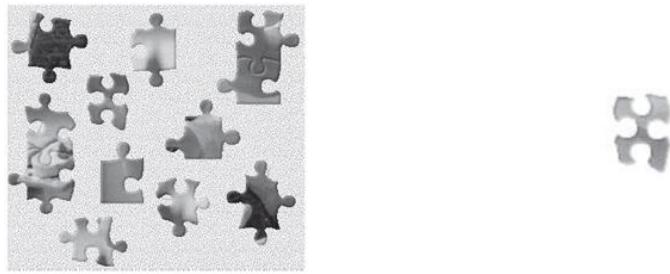


Figura 4.1: Extração de forma utilizando subtração e limiarização.

4.3 Modelo de Correspondência *Template Matching*

O *template matching* é conceitualmente um processo simples. Baseia-se na tentativa de corresponder uma sub-imagem (*template*) que contenha a forma desejada a uma região da imagem original, conforme Figura 4.2, adaptada de [25]. Dessa forma centraliza-se o *template* em um ponto da imagem contabilizando a quantidade de pontos que correspondem-se. O procedimento é repetido por toda a imagem e então o ponto de maior peso corresponde ao modelo na imagem.

Obviamente esse processo pode ser generalizado para localizar, por exemplo, *templates*



(a) *Imagen contendo formas.*

(b) *Template da forma alvo.*

Figura 4.2: *Template matching.*

de diferentes tamanhos e orientações. Nesses casos é necessário testar todos os *templates*, de rotações e tamanhos esperados, para determinar a melhor correspondência.

Formalmente o *template matching* pode ser definido como um método de estimativa de parâmetros. Os parâmetros definem a posição do *template*. Pode-se definir o *template* como uma função discreta \mathbf{T}_{xy} que toma valores em uma janela, isto é, as coordenadas dos pontos $(x, y) \in \mathbf{W}$. Por exemplo, para um *template* 2×2 tem-se o seguinte conjunto de pontos $\mathbf{W} = (0, 0), (0, 1), (1, 0), (1, 1)$.

Considera-se que cada pixel da imagem \mathbf{I}_{xy} seja corrompido por ruído Gaussiano aditivo de média zero e desvio padrão δ desconhecido. Então, a probabilidade de um ponto (i, j) do *template* corresponder a um pixel na posição $(x, y) \in \mathbf{W}$ é obtida pela distribuição normal mostrada na Equação (4.1).

$$p_{i,j}(x, y) = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-\frac{1}{2}(\frac{I_{x+i,y+j}-T_{x,y}}{\sigma})^2} \quad (4.1)$$

Desde que o ruído afete cada pixel de forma independente a probabilidade do *template* estar na posição (i, j) é a probabilidade combinada de cada pixel que o *template* cobre, conforme a Equação (4.2).

$$L_{i,j} = \left(\frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp^{-\frac{1}{2}\sum_{(x,y) \in W} (\frac{I_{x+i,y+j}-T_{x,y}}{\sigma})^2} \quad (4.2)$$

onde:

n - número de pixels do *template*.

Essa função é chamada de função de verossimilhança, geralmente expressa na forma logarítmica, como mostrado na Equação (4.3).

$$\ln(L_{i,j}) = n \ln \frac{1}{\sqrt{2\pi\sigma}} - \frac{1}{2} \sum_{(x,y) \in W} \left(\frac{I_{x+i,y+j} - T_{x,y}}{\sigma} \right)^2 \quad (4.3)$$

Na estimacão da máxima verossimilhança escolhe-se o parâmetro que maximiza a função, isto é, a posição que minimiza a taxa de variação da função objeto.

$$\frac{\partial \ln(L_{ij})}{\partial i} = 0 \quad (4.4a)$$

$$\frac{\partial \ln(L_{ij})}{\partial j} = 0 \quad (4.4b)$$

isto é,

$$\sum_{(x,y) \in W} I_{x+i,y+j} - T_{x,y} \frac{\partial I_{x+i,y+j}}{\partial i} = 0 \quad (4.5a)$$

$$\sum_{(x,y) \in W} I_{x+i,y+j} - T_{x,y} \frac{\partial I_{x+i,y+j}}{\partial j} = 0 \quad (4.5b)$$

logo,

$$\min e = \sum_{(x,y) \in W} (I_{x+i,y+j} - T_{x,y})^2 \quad (4.6)$$

Assim, estimar a máxima verossimilhança é equivalente a escolher a posição do *template* que minimiza o erro quadrático (o quadrado da diferença entre os pontos do *template* e os pontos correspondentes da imagem). A posição onde o *template* melhor encaixa-se à imagem é a posição estimada do *template* dentro da imagem. Pode-se também utilizar outro critério de medida de erro como, por exemplo, o erro absoluto.

Pode-se derivar uma forma alternativa do critério do erro quadrático considerando que

a Equação (4.6) seja escrita conforme a Equação (4.7).

$$\min e = \sum_{x,y \in W} (I_{x+i,y+j}^2 - 2I_{x+i,y+j}T_{x,y} + T_{x,y}^2) \quad (4.7)$$

O último termo da Equação (4.7) não depende da posição (i, j) do *template*. Ele é constante e não pode ser minimizado. Logo o ponto ótimo dessa equação pode ser obtido minimizando

$$\min e = \sum_{x,y \in W} (I_{x+i,y+j}^2 - 2I_{x+i,y+j}T_{x,y}) \quad (4.8)$$

Se o primeiro termo da Equação (4.8) for aproximadamente constante então o segundo termo representa a medida de similaridade entre a imagem e o *template*. Pode-se dessa forma maximizar a correlação cruzada entre o *template* e a imagem. Assim a melhor posição pode ser computada pela Equação (4.9).

$$\max e = \sum_{x,y \in W} I_{x+i,y+j} - T_{x,y} \quad (4.9)$$

No entanto, o termo quadrado na Equação (4.8) pode variar com a posição e dessa forma a correspondência definida pela Equação (4.9) pode ser fraca. Além disso, a variação da função de correlação cruzada é dependente do tamanho do *template* e não invariante às mudanças de iluminação.

Um caso particular do *template matching* ocorre quando a imagem e o modelo são binários. A utilização de imagens binárias reduz consideravelmente o custo computacional, pois cada termo da Equação (4.6) pode assumir apenas dois valores: Um (1) quando $I_{x+i,y+j} = T_{x,y}$, e zero (0) caso contrário. Logo a Equação (4.6) pode ser escrita conforme a Equação (4.10).

$$\max e = \sum_{x,y \in W} \overline{I_{x+i,y+j} \oplus T_{x,y}} \quad (4.10)$$

A maioria das aplicações requer muitos graus de liberdade como a rotação (orientação),

a escala (tamanho), ou as deformações de perspectiva. A rotação pode ser tratada por meio da rotação do *template* ou da utilização de coordenadas polares; a invariância em escala pode ser alcançada utilizando *templates* de diferentes tamanhos. Dessa forma, com mais parâmetros de interesse o espaço acumulador torna-se maior, ou seja, sua dimensão aumenta para cada parâmetro extra.

A posição invariante implica em um espaço de parâmetros 2D enquanto a posição e a escala invariantes requerem espaço 3D. Logo o custo computacional do *template matching* é bastante alto. Se o *template* for de tamanho $m \times m$ e a imagem de tamanho $N \times N$ então os m^2 pixéis são comparados com todos os pontos da imagem (exceto as bordas) resultando para o caso de posição invariante em um custo computacional da ordem de N^2m^2 .

4.4 *Template Matching* Utilizando Transformada de Fourier (TF)

Template Matching via Transformada de Fourier utiliza a dualidade entre a convolução e a multiplicação. Essa dualidade estabelece que uma multiplicação no domínio do espaço corresponda à convolução no domínio da frequência e vice versa. Assim, é explorada a maior velocidade de processamento alcançada pelo algoritmo da Transformada Rápida de Fourier (FFT). Logo, com o objetivo de encontrar uma determinada forma em uma imagem pode-se calcular a correlação cruzada por meio da multiplicação no domínio da frequência.

No entanto, o processo de *matching* por meio da Equação (4.9) é na verdade uma correlação e não uma convolução, assim é necessário expressar essa correlação em termos de uma convolução. A Equação (4.9) pode ser escrita conforme a Equação (4.11).

$$I \otimes T = \sum_{x,y \in W} I_{x',y'} T_{x'-i,y'-j} \quad (4.11)$$

onde:

$$x' = x + i;$$

$$y' = y + j.$$

A convolução é definida conforme a Equação (4.12).

$$I * T = \sum_{x,y \in W} I_{x',y'} T_{i-x',j-y'} \quad (4.12)$$

Assim, para construir o *Template matching* no domínio da frequência pode-se expressar a Equação (4.11) em termos da Equação (4.12). Isto pode ser alcançado considerando a Equação (4.13).

$$I \otimes T = I * T' = \sum_{x,y \in W} I_{x',y'} T'_{i-x',j-y'} \quad (4.13)$$

onde,

$$T' = T_{-x,-y} \quad (4.14)$$

Ou seja, a correlação é equivalente a convolução quando o *template* é transformado conforme a Equação (4.14). Assim a Equação (4.13) pode ser escrita conforme a Equação (4.15).

$$I * T' = F^{-1}(F(I)F(T')) \quad (4.15)$$

onde:

F - Transformada de Fourier (calculada pela FFT);

F^{-1} - Transformada de Fourier inversa.

Pode-se observar que as transformações possuem as mesmas dimensões mesmo que a forma do *template* seja geralmente menor que a imagem. Nesse caso a solução é incluir zeros (*zero-padding*) para fornecer ao *template* o mesmo tamanho da imagem.

A Figura 4.3, adaptada de [25], ilustra o resultado do *Template Matching* no domínio de Fourier. A Figura 4.3(a) mostra o *template* depois do processo de inversão e *zero-padding*. A TF da imagem e do *template* correspondem as Figuras 4.3(b) e 4.3(c), respectivamente.

Essas transformadas são então multiplicadas ponto a ponto resultando na imagem da Figura 4.3(d). A Transformada inversa da Figura 4.3(d) resulta na Figura 4.3(e) que mostra onde o *template* melhor adapta-se a imagem.

A imagem resultante contém alguns pontos de máximo (em branco). Isso pode ser explicado pelo fato do algoritmo não levar em conta o segundo termo da Equação (4.8). Além disso, a forma pode, parcialmente, corresponder a alguns outros padrões na imagem. A Figura 4.3(f) mostra um *zoom* da região onde o valor máximo está localizado. Pode-se verificar nessa figura que o pico é bem definido.

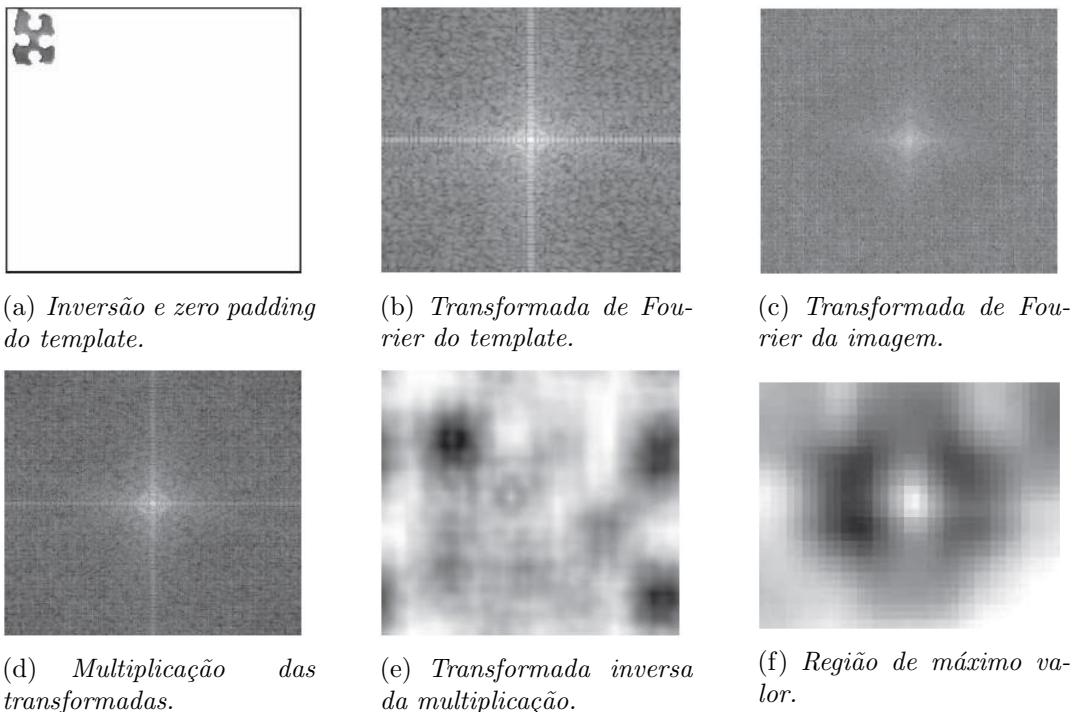


Figura 4.3: *Template matching no domínio de Fourier.*

4.5 Transformada de Hough (TH)

A transformada da Hough (TH) (Hough, 1962) é uma técnica que localiza formas em imagens digitais. Em particular ela é usada na extração de linhas, círculos e elipses. No caso de linhas sua definição matemática é equivalente a Transformada de Radon. A TH foi

introduzida por Hough e usada para encontrar sinais de bolhas. Todavia, Rosenfeld notou seu potencial no processamento de imagens (Rosenfeld, 1969).

A principal vantagem da TH está no fato dela ser bem mais rápida que o *template matching*. Sua implementação define um mapeamento dos pontos da imagem em um espaço acumulador (*Hough space*). O mapeamento é alcançado de maneira eficiente baseado na função que descreve a forma procurada. Esse mapeamento requer muito menos esforço computacional que o *template matching*. Todavia, a TH ainda requer considerável capacidade de armazenamento e processamento.

4.5.1 TH para linhas

Em uma parametrização cartesiana, pontos colineares em uma imagem com coordenadas (x, y) estão relacionados por sua inclinação m e seu ponto de intersecção c de acordo com a Equação (4.16).

$$y = mx + c \quad (4.16)$$

A Equação (4.16) pode ser escrita na forma homogênea conforme a Equação (4.17).

$$Ay + Bx + 1 = 0 \quad (4.17)$$

onde:

$$A = \frac{-1}{c};$$

$$B = \frac{m}{c}.$$

Então uma linha pode ser definida considerando um par de valores (A, B) .

A Equação (4.17) pode ser vista como um sistema de equações simplesmente reescritas em termos de parametrização cartesiana como mostra a Equação (4.18).

$$c = -x_i m + y_i \quad (4.18)$$

Logo, para determinar uma linha deve-se encontrar os parâmetros (m, c) ou (A, B) que satisfaça as Equações (4.17) e (4.18) respectivamente. Pode-se observar que o sistema é geralmente sobredeterminado, isto é, existem mais equações do que incógnitas. Então deve-se encontrar a solução mais próxima que satisfaça todas as equações simultaneamente.

A relação entre um ponto (x_i, y_i) em uma imagem e a linha da Equação (4.18) é ilustrada na Figura 4.4. Os pontos (x_i, y_i) e (x_j, y_j) na Figura 4.4(a) definem as linhas U_i e U_j na Figura 4.4(b), respectivamente. O ponto (A, B) no espaço de parâmetros é comum a essas duas retas associadas aos pontos (x_i, y_i) e (x_j, y_j) . Na verdade, todos os pontos colineares em uma imagem interceptam-se no mesmo ponto concorrente (A, B) independente da parametrização da linha utilizada.

A TH resolve esse sistema de uma forma muito eficiente simplesmente armazenando as soluções potenciais em uma matriz acumuladora (matriz de votos). O contador é feito traçando todas as linhas para cada ponto (x_i, y_i) . Cada traço incrementa um elemento na matriz e então o problema de extração de uma linha torna-se um problema de localização de um máximo no espaço acumulador. Essa estratégia é robusta e tem demonstrado ser eficiente na presença de ruído e oclusões.

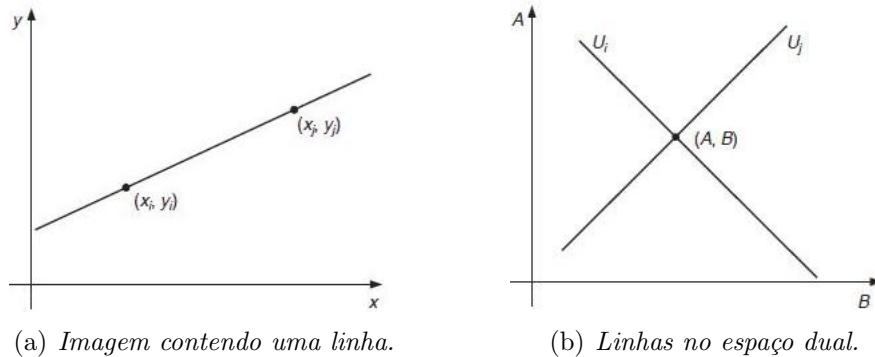


Figura 4.4: Transformada de Hough para linhas.

Os eixos no espaço dual representam os parâmetros da linha. No caso de parametrização cartesiana m pode assumir uma gama infinita de valores uma vez que as linhas podem variar de horizontal para vertical. Uma vez que os votos são armazenados na matriz é possível considerar as técnicas de *antialiasing* e melhorar a estratégia de armazenamento.

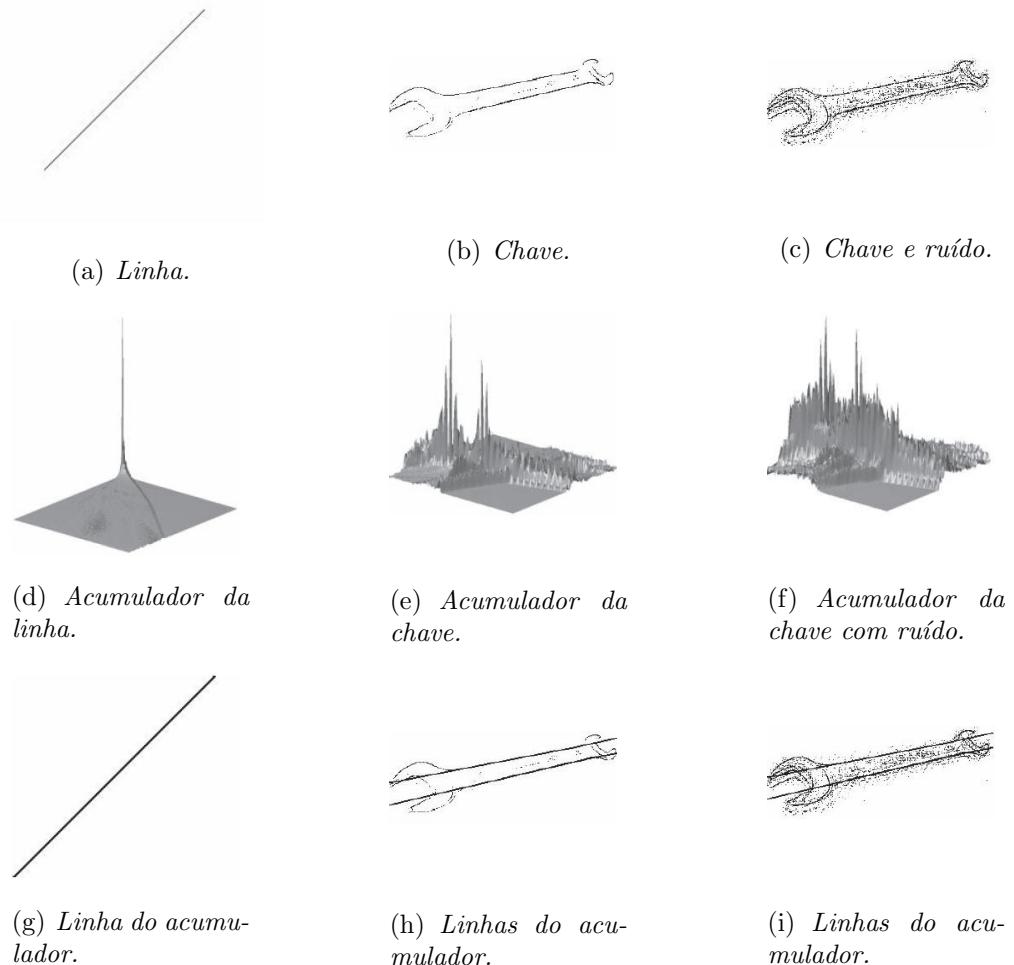


Figura 4.5: Aplicação da TH para linhas.

A magnitude do pico é proporcional ao número de pixels na linha gerada, de acordo com a Figura 4.5, adaptada de [25]. As bordas da chave nas Figuras 4.5(b) e 4.5(c) definem duas linhas principais. A imagem da Figura 4.5(c), por conter mais ruído, apresenta um número maior de votos, porém o valor de pico encontra-se no mesmo lugar.

Pode-se verificar nas Figuras 4.5(g), 4.5(h) e 4.5(i) que a TH fornece uma estimativa correta dos parâmetros que especificam as linhas. Todavia, o problema de não linearidade dos parâmetros e a discretização produzem ruído no acumulador.

Um dos principais problemas na execução da TH é a definição de um espaço acumulador adequado. Por outro lado seu custo computacional depende do número de pontos de bordas

(n_e) e do comprimento das linhas formadas no espaço de parâmetros (l), com um custo de $O(n_e l)$, consideravelmente menor do que o custo do *Template matching*.

Uma alternativa para evitar os problemas da parametrização cartesiana na TH é basear a função de mapeamento em uma parametrização alternativa. Uma das técnicas mais comprovadas é a chamada “Parametrização Pé Normal” (*foot-of-normal*) que determina uma linha considerando um ponto (x, y) como uma função perpendicular à linha que passa pela origem da imagem. Esta forma também é conhecida como TH Polar.

O ponto onde essa linha intercepta a linha na imagem é obtido pela Equação (4.19).

$$\rho = x \cos(\theta) + y \sin(\theta) \quad (4.19)$$

onde:

θ - ângulo da linha normal à linha da imagem;

ρ - distância entre a origem e o ponto onde as linhas se cruzam, como ilustrado na Figura 4.6, adaptada de [25].

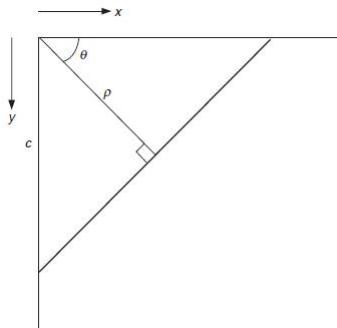


Figura 4.6: *Linha na forma polar*.

Lembrando que duas linhas são perpendiculares se o produto de suas inclinações m é -1 , e considerando a geometria do arranjo da Figura 4.6, obtém-se a Equação (4.20).

$$c = \frac{\rho}{\sin(\theta)} \quad (4.20a)$$

$$m = -\frac{1}{\tan(\theta)} \quad (4.20b)$$

Assim, tem-se uma função de mapeamento diferente onde os votos são expressos de forma senoidal em um acumulador 2D em termos de ρ e θ .

A vantagem desse mapeamento alternativo é que os valores dos parâmetros ρ e θ são limitados. A variação de ρ está dentro de 180° e seus possíveis valores são dados pelo tamanho da imagem, pois o comprimento máximo da linha é de $\sqrt{M^2 + N^2}$, onde M e N são os lados da imagem.

4.5.2 TH para Círculos

A TH pode ser estendida introduzindo a equação da curva no processo de detecção. Esta equação pode ser considerada na forma explícita ou paramétrica. De forma explícita a TH pode ser definida considerando a equação de um círculo conforme a Equação (4.21).

$$(x - x_0)^2 + (y - y_0)^2 = r^2 \quad (4.21)$$

A Equação (4.21) define um conjunto de pontos (x, y) , centrados na região (x_0, y_0) e raio r . Essa equação, novamente, pode ser vista de duas maneiras: como um locus de pontos (x, y) em uma imagem, ou como um conjunto de pontos (x_0, y_0) centrados em (x, y) com raio r .

A Figura 4.7, adaptada de [25], ilustra as duas definições. Cada ponto da borda define um conjunto de círculos no espaço do acumulador. Estes círculos são definidos para todos os valores possíveis do raio e são centrados nas coordenadas do ponto de borda. A Figura 4.7(b) mostra três círculos definidos por três pontos localizados na borda. Esses círculos são dados para um determinado valor de r . Na verdade, cada ponto da borda define círculos para os outros valores do raio. Isto implica que o espaço acumulador é tridimensional e que os pontos de borda são mapeados em um cone no espaço acumulador. A Figura

4.7(c) ilustra esse acumulador. Depois de reunir todos os pontos de borda em evidência o máximo no espaço acumulador corresponde aos parâmetros do círculo original da imagem. O procedimento é o mesmo descrito para a TH de linha, porém os votos são gerados pela Equação (4.21).

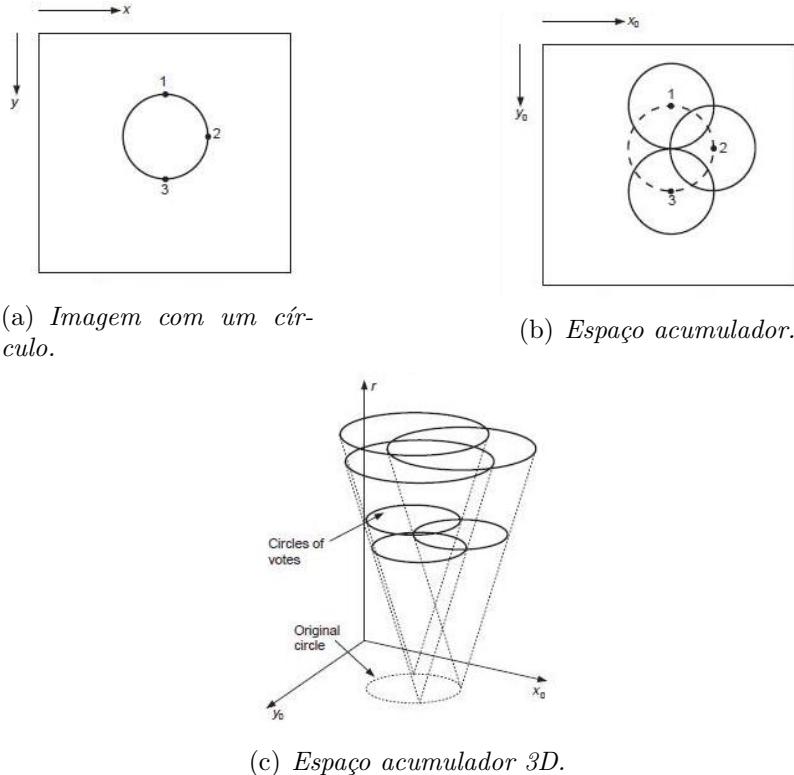


Figura 4.7: Transformada de Hough para círculos.

A Equação (4.21) pode ser definida na forma paramétrica, como mostrado na Equação (4.22).

$$x = x_0 + r \cos(\theta) \quad (4.22a)$$

$$y = y_0 + r \sin(\theta) \quad (4.22b)$$

Observe que θ não é um parâmetro livre, mas define o traço da curva. O traço da curva (ou superfície) é comumente referenciado como função de espalhamento pontual.

Essas equações definem pontos no espaço acumulador dependente do raio r . A vantagem dessa representação é que ela permite ser resolvida por parâmetros. Então a TH mapeada é definida pela Equação (4.23).

$$x_0 = x - r \cos(\theta) \quad (4.23a)$$

$$y_0 = y - r \sin(\theta) \quad (4.23b)$$

O acumulador é na verdade 2D, em termos dos parâmetros do centro e para um raio fornecido como argumento fixo da função. Um círculo é formado variando θ de 0° a 360° . A discretização de θ controla o efeito da granulação. Um incremento pequeno cobre o espaço de parâmetros de forma mais suave, muito grande resulta em uma cobertura esparsa. O espaço acumulador, inicialmente zero, é incrementado somente para pontos cujas coordenadas situam-se dentro do intervalo especificado.

A aplicação para a TH do círculo é ilustrada na Figura 4.8, adaptada de [25]. Nesta figura as bordas são completamente bem definidas. O pico do acumulador está no centro do círculo, além disso, nota-se que o algoritmo é capaz de resolver problemas de oclusão e ruído.

Outro exemplo de extração de círculos pela TH é mostrado na Figura 4.9, adaptada de [25]. Esta figura corresponde a uma imagem real de baixa resolução pré-processada pelo detector de bordas de Sobel. O raio, parâmetro de entrada do algoritmo, foi fixado em 5 pixels. Pode-se verificar que a TH encontra círculos com um número máximo de pontos; é possível incluir outras constantes de controle, como por exemplo, gradiente de direção para objetos com um perfil de iluminação.

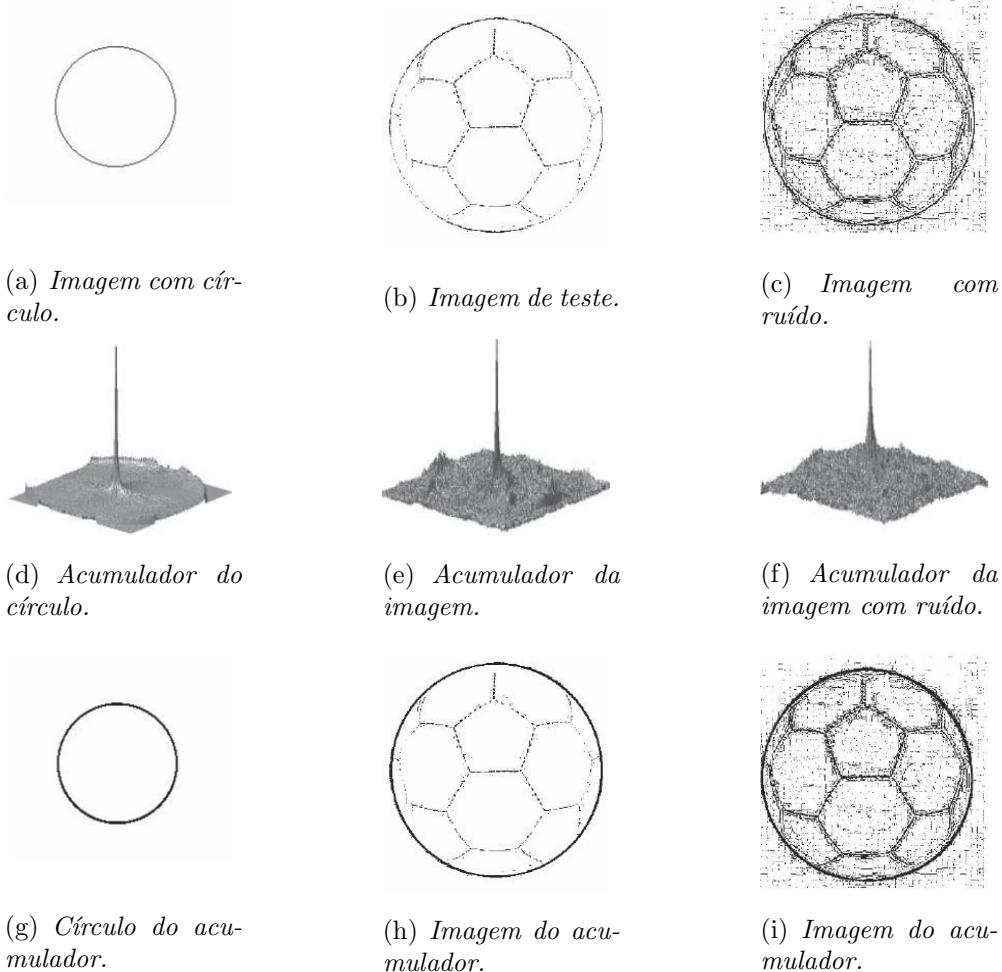


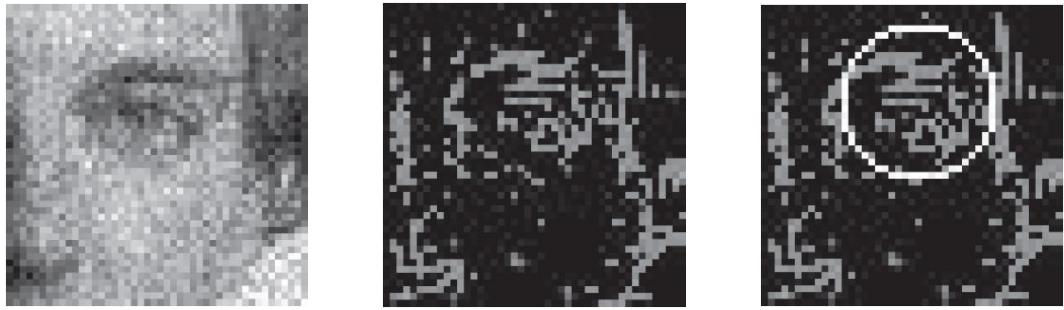
Figura 4.8: Aplicação da TH para círculos.

4.5.3 TH para Elipses

Círculos são muito importantes na detecção de formas, pois muitos objetos possuem forma circular. Em todo caso, devido ao ponto de vista da câmera nem sempre são encontrados círculos perfeitos nas imagens.

As imagens são formadas pelo mapeamento de uma forma no espaço 3D em um plano que corresponde a uma transformação de perspectiva. Neste processo um círculo é transformado em uma elipse.

Pode-se definir o mapeamento entre um círculo e uma elipse pela transformação de similaridade, como mostrado na Equação (4.24).



(a) Imagem do olho.

(b) Bordas de Sobel.

(c) TH para círculos.

Figura 4.9: Uso da TH para círculos.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\rho) & \sin(\rho) \\ -\sin(\rho) & \cos(\rho) \end{bmatrix} \begin{bmatrix} S_x \\ S_y \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4.24)$$

onde:

(x', y') - define as coordenadas do círculo;

ρ - representa a orientação;

(S_x, S_y) - fator de escala;

(t_x, t_y) - translação.

Se,

$$a_0 = t_x \quad (4.25a)$$

$$a_x = S_x \cos(\rho) \quad (4.25b)$$

$$b_x = \sin(\rho) \quad (4.25c)$$

$$b_0 = t_y \quad (4.25d)$$

$$a_y = -S_y \sin(\rho) \quad (4.25e)$$

$$S_y = \cos(\rho) \quad (4.25f)$$

Então, obtém-se o círculo deformado, como mostrado na Equação (4.26).

$$x = a_0 + a_x \cos(\theta) + b_x \sin(\theta) \quad (4.26a)$$

$$y = b_0 + a_y \cos(\theta) + b_y \sin(\theta) \quad (4.26b)$$

A Equação (4.26) corresponde à representação polar de uma elipse. Esta forma polar contém seis parâmetros $(a_0, b_0, a_x, b_x, a_y, b_y)$ que caracterizam a forma da elipse. θ não é um parâmetro livre e ele somente endereça um ponto particular pertencente à elipse. Contudo, um parâmetro é redundante desde que ele possa ser calculado considerando a ortogonalidade (independência) dos eixos da elipse (o produto $a_x b_x + a_y b_y = 0$ propriedade da elipse).

Dessa forma a elipse é definida pelo seu centro (a_0, b_0) e três dos parâmetros axiais (a_x, b_x, a_y, b_y) ; seu tamanho ao longo dos eixos (dois parâmetros) e sua rotação (um parâmetro). De fato os parâmetros axiais relacionam-se conforme as Equações (4.27).

$$\tan(\rho) = \frac{a_y}{a_x} \quad (4.27a)$$

$$a = \sqrt{a_x^2 + a_y^2} \quad (4.27b)$$

$$b = \sqrt{b_x^2 + b_y^2} \quad (4.27c)$$

onde:

(a, b) - eixos da elipse, conforme mostra a Figura 4.10.

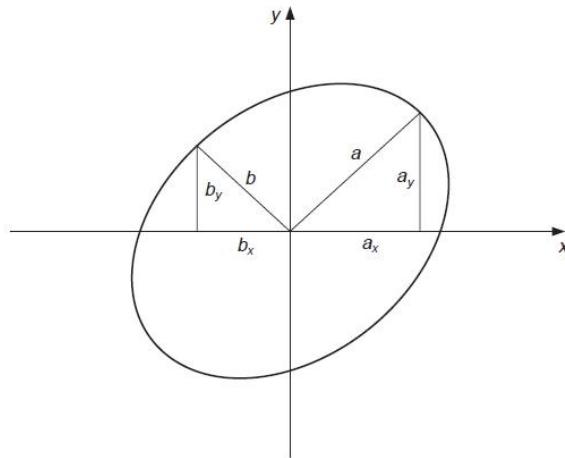


Figura 4.10: *Definição dos eixos da elipse.*

De maneira similar à Equação (4.22), as Equações (4.26) e (4.27) podem ser usadas para gerar a função de mapeamento na TH. Neste caso, a localização do centro da elipse é obtida pela Equação (4.28).

$$a_0 = x - a_x \cos(\theta) + b_x \sin(\theta) \quad (4.28a)$$

$$b_0 = y - a_y \cos(\theta) + b_y \sin(\theta) \quad (4.28b)$$

A localização é dependente de três parâmetros, logo o mapeamento define uma hiper superfície em um espaço 5D. Este espaço pode ser muito grande, por exemplo, se houver 100 valores possíveis para cada uma das cinco variáveis, então o espaço acumulador 5D conterá 10^{10} valores, cerca de 10 *GB* de armazenamento.

Devido ao interesse na detecção de elipses algumas técnicas não utilizam a Equação (4.28) de forma direta e assim, conseguem maior velocidade e menor armazenamento. Para esse caso o acumulador então reduz-se a um espaço 2D.

Na prática, para localizar uma elipse é necessário experimentar todos os prováveis valores de comprimento do eixo, o que torna o cálculo impossível a menos que se limite o processamento à apenas alguns poucos valores.

A Figura 4.11, adaptada de [25], ilustra o uso da TH para a elipse.

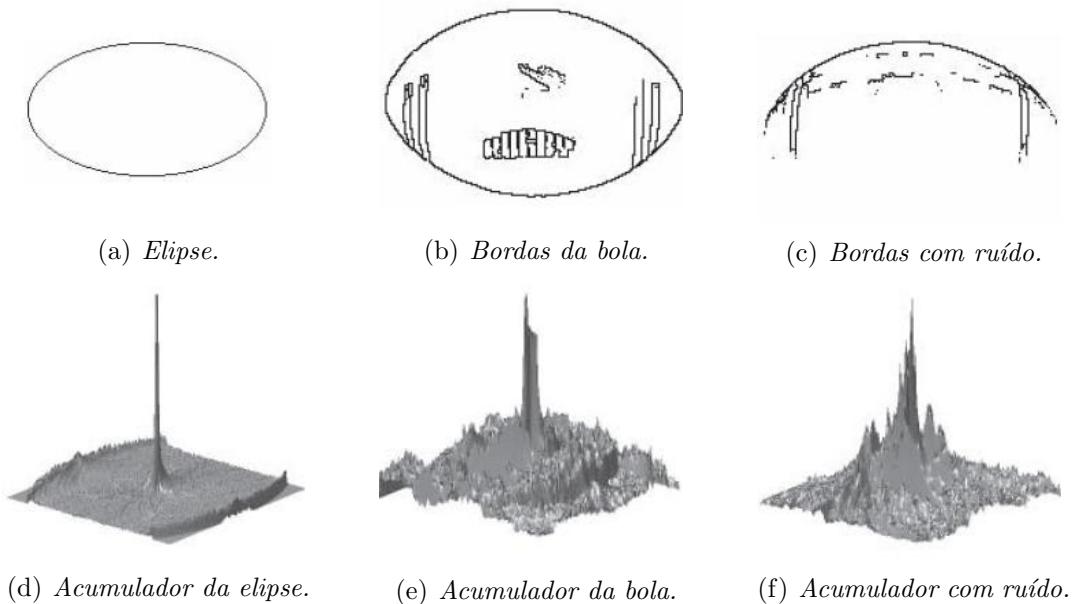


Figura 4.11: *Aplicação da TH para elipses.*

4.5.4 Decomposição do Espaço de Parâmetros

A TH retorna o mesmo resultado ótimo que o *template matching*, porém de forma mais rápida e com menor custo computacional. Conforme descrito nas seções anteriores deste capítulo o custo computacional aumenta com a complexidade da forma a ser detectada. Por esta razão muitas pesquisas relacionadas a TH tem como objetivo o desenvolvimento de técnicas destinadas a reduzir sua complexidade computacional.

Uma das estratégias para essa redução baseia-se no uso de propriedades geométricas que são, geralmente, definidas pela relação entre os pontos.

4.5.5 Redução do Espaço de Parâmetros para Linhas

Para uma linha o acumulador pode ser reduzido de 2D para 1D considerando que a inclinação m pode ser obtida a partir da própria imagem. A inclinação pode ser computada ou utilizando um gradiente de direção em um ponto ou considerando um par de pontos, isto é,

$$m = \delta \quad \text{ou} \quad m = \frac{y_2 - y_1}{x_2 - x_1} \quad (4.29)$$

onde:

δ - gradiente de direção no ponto.

Para o caso de dois pontos tem-se a Equação (4.30).

$$c = \frac{x_2 y_1 - x_1 y_2}{x_2 - y_1} \quad (4.30)$$

Logo, de acordo com a Equação (4.20), o parâmetro da representação polar para linhas, θ , é agora obtido pela Equação (4.31).

$$\theta = -\tan^{-1} \left[\frac{1}{\delta} \right] \quad (4.31a)$$

$$\theta = \tan^{-1} \left[\frac{x_1 - x_2}{y_2 - y_1} \right] \quad (4.31b)$$

Essas equações não dependem de ρ , logo ρ e θ são independentes. Além disso, o uso de técnicas estatísticas podem reduzir o espaço de pontos de forma que uma amostra represente toda a informação.

A Figura 4.12, adaptada de [25], mostra o acumulador para os dois parâmetros θ e ρ . O pico no segundo acumulador define duas linhas com inclinações similares.

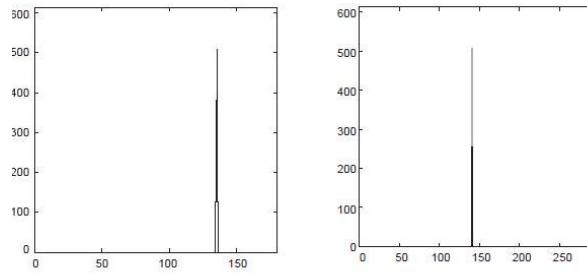
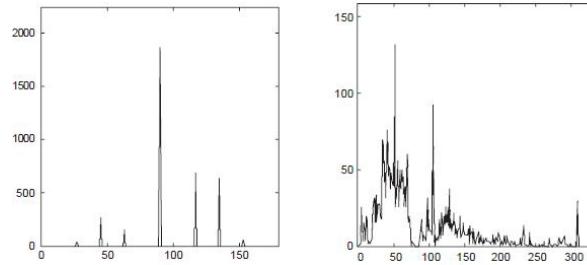

 (a) *Acumulador para Figura 4.5(a).*

 (b) *Acumulador para Figura 4.5(b).*

 Figura 4.12: *Espaço de parâmetros reduzido para TH de linhas.*

4.5.6 Redução do Espaço de Parâmetros para Círculos

Para o caso de linhas a relação entre a informação local e a inclusão de um grupo de pontos (par) é uma alternativa de descrição analítica simples. Para primitivas mais complexas é possível incluir algumas relações geométricas. Estas relações não são definidas para um conjunto arbitrário de pontos, mas incluem limitações angulares que definem as posições relativas entre elas. Em geral consideram-se propriedades geométricas diferentes para decompor o espaço de parâmetros de um círculo o que motiva o desenvolvimento de vários métodos.

Uma relação geométrica importante é obtida pela segunda derivada direcional. Essa relação pode ser obtida considerando que a equação paramétrica da circunferência, conforme a Equação (4.22), define uma função vetor de posição, conforme a Equação (4.32).

$$\omega(\theta) = x(\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y(\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.32)$$

onde,

$$x(\theta) = x_0 + r \cos(\theta) \quad (4.33a)$$

$$y(\theta) = y_0 + r \sin(\theta) \quad (4.33b)$$

A derivada da Equação (4.32) com respeito a θ define a primeira derivada direcional, ou seja, a Equação (4.34).

$$\nu'(\theta) = x'(\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y'(\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.34)$$

Derivando-se a Equação (4.34) obtém-se a segunda derivada direcional, conforme a Equação (4.35).

$$\nu''(\theta) = x''(\theta) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + y''(\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (4.35)$$

onde,

$$x'(\theta) = -r \sin(\theta) \quad (4.36a)$$

$$y'(\theta) = r \cos(\theta) \quad (4.36b)$$

e

$$x''(\theta) = -r \cos(\theta) \quad (4.37a)$$

$$y''(\theta) = -r \sin(\theta) \quad (4.37b)$$

A Figura 4.13, adaptada de [25], ilustra a definição da primeira e da segunda derivada direcional. A primeira define o vetor tangencial enquanto a segunda é similar à função vetor, mas com a direção inversa.

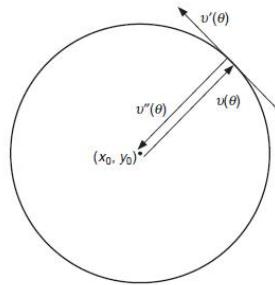


Figura 4.13: Definição da primeira e da segunda derivada para um círculo.

A tangente do ângulo da primeira derivada direcional é obtida pela Equação (4.38).

$$\phi(\theta) = \frac{y'(\theta)}{x'(\theta)} = -\frac{1}{\tan(\theta)} \quad (4.38)$$

e o ângulo, denotado por $\hat{\phi}'$ é encontrado pela Equação (4.39).

$$\hat{\phi}'(\theta) = \tan^{-1}(\phi'(\theta)) \quad (4.39)$$

Do mesmo modo, a tangente da segunda derivada direcional e o ângulo são obtidos pelas Equações (4.40) e (4.41).

$$\phi''(\theta) = \frac{y''(\theta)}{x''(\theta)} = \tan(\theta) \quad (4.40)$$

$$\hat{\phi}''(\theta) = \tan^{-1}(\phi''(\theta)) \quad (4.41)$$

Observando a definição de $\phi''(\theta)$ têm-se a Equação (4.42).

$$\phi''(\theta) = \frac{y''(\theta)}{x''(\theta)} = \frac{y(\theta) - y_0}{x(\theta) - x_0} \quad (4.42)$$

A Equação (4.42) define uma linha reta que passa pelos pontos $(x(\theta), y(\theta))$ e (x_0, y_0) e é, possivelmente, a relação mais importante na decomposição do espaço de parâmetro. A definição de linha é mais evidente rearranjando os termos, assim obtém-se a Equação (4.43).

$$y(\theta) = \phi''(\theta)(x(\theta) - x_0) + y_0 \quad (4.43)$$

A Equação (4.43) é independente do raio, logo ela pode ser utilizada para acumular evidências da localização de formas em um acumulador 2D. O mapeamento da TH é definido pela forma dual conforme a Equação (4.44).

$$y_0 = \phi''(\theta)(x_0 - x(\theta)) + y(\theta) \quad (4.44)$$

Isto é, de pose de um ponto na imagem $(x(\theta), y(\theta))$ e o valor de $\phi''(\theta)$ pode-se gerar uma linha de votos em um acumulador (x_0, y_0) . Uma vez que o centro do círculo é conhecido, então um acumulador 1D pode ser utilizado para alocar os raios.

O aspecto chave da decomposição do espaço é o método utilizado para obter o valor de $\phi''(\theta)$ a partir da imagem. São consideradas duas formas, primeiro, onde $\phi''(\theta)$ pode ser obtido pela informação da direção de borda e segundo a partir da informação de um par de pontos.

Utilizando-se as Equações (4.40) e (4.41) pode-se mostrar que as tangentes $\phi'(\theta)$ e $\phi''(\theta)$ são perpendiculares, então obtém-se a Equação (4.45).

$$\phi''(\theta) = -\frac{1}{\phi'(\theta)} \quad (4.45)$$

Logo o mapeamento da TH na Equação (4.44) pode ser escrito em termos do gradiente $\phi'(\theta)$ conforme a Equação (4.46).

$$y_0 = y(\theta) + \frac{x(\theta) - x_0}{\phi'(\theta)} \quad (4.46)$$

A Equação (4.46) tem uma interpretação geométrica simples que é ilustrada na Figura 4.14(a), adaptada de [25]. Pode-se verificar que a linha de votos passa pelos pontos $x(\theta), y(\theta)$ e (x_0, y_0) . A inclinação da linha é perpendicular à direção do gradiente de direção.

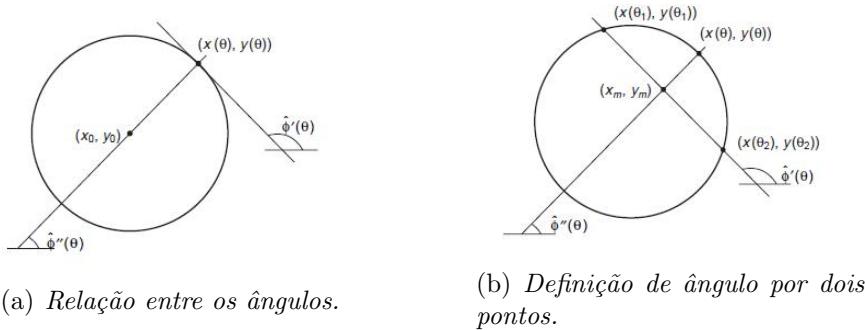


Figura 4.14: Geometria do ângulo da primeira e da segunda derivada direcional.

Uma decomposição alternativa pode ser obtida considerando a geometria mostrada na Figura 4.14(b), também adaptada de [25], onde tomados um par de pontos (x_1, y_1) e (x_2, y_2) , com $x_i = x(\theta_i)$, então a linha que passa pelos pontos tem a mesma inclinação da linha no ponto $(x(\theta), y(\theta))$. Consequentemente,

$$\phi'(\theta) = \frac{y_2 - y_1}{x_2 - x_1} \quad (4.47)$$

onde,

$$\theta = \frac{1}{2}(\theta_1 + \theta_2) \quad (4.48)$$

Baseado na Equação (4.26), obtém-se a Equação (4.49).

$$\phi''(\theta) = -\frac{x_2 - x_1}{y_2 - y_1} \quad (4.49)$$

O problema com o uso do par de pontos é que pela Equação (4.47) não se pode determinar a localização do ponto $(x(\theta), y(\theta))$. Por sorte, a linha de votos também passa pelo ponto médio da linha entre os dois pontos selecionados, assim, obtém-se a Equação (4.50).

$$x_m = \frac{1}{2}(x_1 + x_2) \quad (4.50a)$$

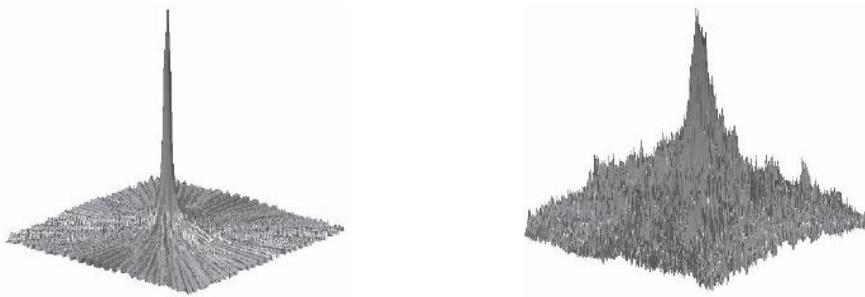
$$y_m = \frac{1}{2}(y_1 + y_2) \quad (4.50b)$$

Então, por substituição da Equação (4.47) na Equação (4.46) e substituindo-se o ponto $(x(\theta), y(\theta))$ por (x_m, y_m) , o mapeamento da TH pode ser expresso pela Equação (4.51).

$$y_0 = y_m + \frac{(x_m - x) - (x_2 - x_1)}{y_2 - y_1} \quad (4.51)$$

A Equação (4.51) não utiliza a informação do gradiente de direção, mas é baseada em um par de pontos. Isso é análogo à decomposição do espaço de parâmetros da linha apresentado pela Equação (4.31). Nesse caso, a inclinação pode ser computada utilizando o gradiente de direção ou, alternativamente, tomando um par de pontos. No caso do círculo, a tangente (e logo, o ângulo da segunda derivada direcional) pode ser computada pelo gradiente da direção. Em todo caso, é importante notar que existem algumas outras combinações de decomposição de espaço de parâmetros.

A Figura 4.15, adaptada de [25], mostra um acumulador. Observa-se claramente o pico que representa a localização do círculo. Picos menores correspondem aos círculos com poucos pontos, em geral, há um compromisso entre o espalhamento do pico e o ruído no acumulador.



(a) *Acumulador para Figura 4.8(a).*

(b) *Acumulador para Figura 4.8(b).*

Figura 4.15: *Redução do espaço de parâmetros da TH para círculos.*

4.5.7 Redução do Espaço de Parâmetros para Elipses

Parte da simplicidade da decomposição do espaço para círculos vem do fato que os círculos são (naturalmente) isotrópicos. No entanto, elipses possuem mais parâmetros livres e são geometricamente mais complexas. Logo suas propriedades geométricas consistem de relações mais complexas entre os pontos, as tangentes e os ângulos. Em todo caso, elas mantêm a relação geométrica definida pelo ângulo da segunda derivada.

De acordo com as Equações (4.32) e (4.34), vetor posição e derivada direcional de uma elipse, na forma da Equação (4.26), tem os componentes, conforme as Equações (4.52) e (4.53).

$$x'(\theta) = -a_x \sin(\theta) + b_x \cos(\theta) \quad (4.52a)$$

$$y'(\theta) = -a_y \sin(\theta) + b_y \cos(\theta) \quad (4.52b)$$

e

$$x''(\theta) = -a_x \cos(\theta) - b_x \sin(\theta) \quad (4.53a)$$

$$y''(\theta) = -a_y \cos(\theta) - b_y \sin(\theta) \quad (4.53b)$$

A tangente do ângulo da primeira e da segunda derivada direcional são encontradas pelas Equações (4.54) e (4.55), respectivamente.

$$\phi'(\theta) = \frac{y'(\theta)}{x'(\theta)} = \frac{-a_y \cos(\theta) + b_y \sin(\theta)}{-a_x \cos(\theta) + b_x \sin(\theta)} \quad (4.54)$$

e

$$\phi''(\theta) = \frac{y''(\theta)}{x''(\theta)} = \frac{-a_y \cos(\theta) - b_y \sin(\theta)}{-a_x \cos(\theta) - b_x \sin(\theta)} \quad (4.55)$$

Considerando-se as Equações (4.52) e (4.53) tem-se que a Equação (4.42) também é válida para a elipse.

$$\frac{y(\theta) - y_0}{x(\theta) - x_0} = \phi''(\theta)$$

A geometria da definição na Equação (4.42) para elipse é ilustrada na Figura 4.16, adaptada de [25]. Como no caso do círculo, essas equações definem uma linha que passa pelos pontos $(x(\theta), y(\theta))$ e (x_0, y_0) . Porém, no caso das elipses os ângulos $\hat{\phi}'$ e $\hat{\phi}''$ não são ortogonais, o que torna o processamento mais complexo.

Para obter $\phi''(\theta)$ estende-se a geometria apresentada na Figura 4.14(a), isto é, toma-se um par de pontos para definir a linha cuja inclinação define o valor de $\phi'(\theta)$ em outro ponto. Isto é ilustrado na Figura 4.14(b). A linha da Equação (4.42) passa pelo ponto médio (x_m, y_m) , no entanto, ela não é ortogonal com a linha tangente.

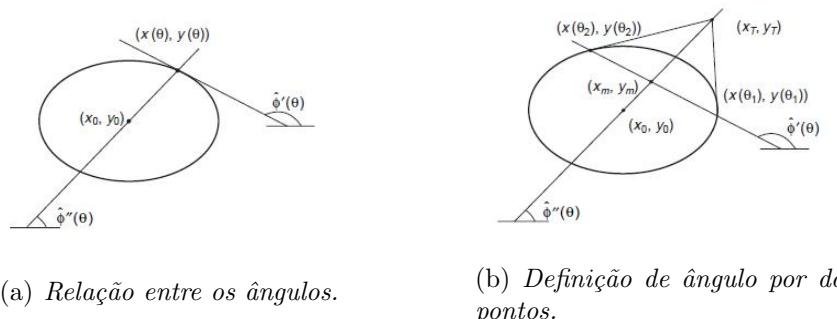


Figura 4.16: Geometria do ângulo da primeira e da segunda derivada direcional.

As relações mostradas na Figura 4.16(b) não dependem da orientação ou da posição da elipse, logo os três pontos podem ser definidos pelas Equações de (4.56).

$$x_1 = a_x \cos(\theta_1) \quad (4.56a)$$

$$x_2 = a_x \cos(\theta_2) \quad (4.56b)$$

$$x = a_x \cos(\theta) \quad (4.56c)$$

$$y_1 = b_x \sin(\theta_1) \quad (4.56d)$$

$$y_2 = b_x \sin(\theta_2) \quad (4.56e)$$

$$y = b_x \sin(\theta) \quad (4.56f)$$

O ponto $(x(\theta), y(\theta))$ é obtido pela intersecção da linha na Equação (4.42) com a elipse, isto é, pela Equação (4.57).

$$\frac{y(\theta) - y_0}{x(\theta) - x_0} = \frac{a_x y_m}{b_y x_m} \quad (4.57)$$

Substituindo-se os valores de (x_m, y_m) definidos como a média das coordenadas dos pontos (x_1, y_1) e (x_2, y_2) obtém-se a Equação (4.58).

$$\tan(\theta) = \frac{a_x}{b_y} \frac{b_y \sin(\theta_1) + b_y \sin(\theta_2)}{a_x \cos(\theta_1) + a_x \cos(\theta_2)} \quad (4.58)$$

e,

$$\tan(\theta) = \tan\left(\frac{1}{2}(\theta_1 + \theta_2)\right) \quad (4.59)$$

Da Equação (4.58) fica evidente que a relação da Equação (4.48) também pode ser utilizada para elipses. Baseado nesse resultado, a tangente do ângulo da segunda derivada direcional pode ser definida pela Equação (4.60).

$$\phi''(\theta) = \frac{b_y}{a_x} \tan(\theta) \quad (4.60)$$

Substituindo na Equação (4.57), obtém-se a Equação (4.61).

$$\phi''(\theta) = \frac{y_m}{x_m} \quad (4.61)$$

A Equação (4.61) é utilizada quando a elipse não é transladada. Se a elipse for transladada a tangente do ângulo pode ser escrita em termos dos pontos (x_m, y_m) e (x_T, y_T) pela Equação (4.62).

$$\phi''(\theta) = \frac{y_T - y_m}{x_T - x_m} \quad (4.62)$$

Considerando-se que o ponto (x_T, y_T) é o ponto de intersecção da linha tangente em (x_1, y_1) e (x_2, y_2) , obtém-se a Equação (4.63).

$$\phi''(\theta) = \frac{AC + 2BD}{2A + BC} \quad (4.63)$$

onde,

$$A = y_1 - y_2 \quad (4.64a)$$

$$B = x_1 - x_2 \quad (4.64b)$$

$$C = \phi_1 + \phi_2 \quad (4.64c)$$

$$D = \phi_1 \phi_2 \quad (4.64d)$$

onde:

ϕ_1 e ϕ_2 - inclinações da tangente para os pontos.

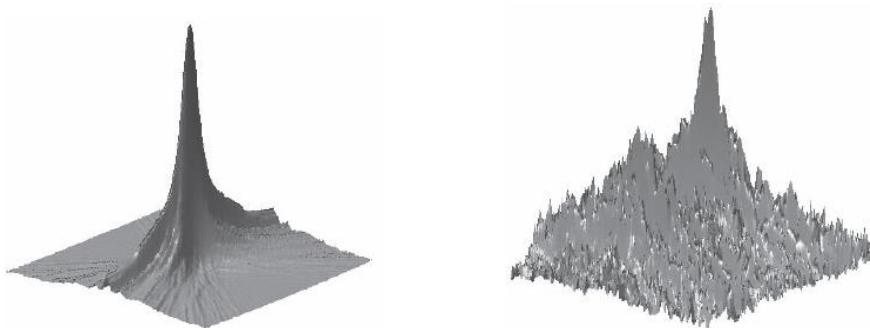
Finalmente, considerando-se a Equação (4.42), o mapeamento da TH para o parâmetro de centro é definido pela Equação (4.65).

$$y_0 = y_m + \frac{AC + 2BD}{2A + BC} (x_0 - x_m) \quad (4.65)$$

A Equação (4.65) pode ser utilizada para coletar evidências independente da escala e da rotação. Uma vez que a localização é conhecida, um espaço de parâmetros 3D é necessário para obter os parâmetros restantes. Em todo caso, esses parâmetros podem ser computados independentemente utilizando-se dois espaços de parâmetros 2D.

Pode-se evitar o uso do gradiente de direção da Equação (4.63) incluindo-se mais pontos. De fato, a tangente $\phi''(\theta)$ pode ser computada tomando-se quatro pontos. Em todo caso, a inclusão de mais pontos, geralmente leva a um aumento do ruído no acumulador.

A Figura 4.17, adaptada de [25], mostra um exemplo de acumulador. O pico representa a localização da elipse. Em geral, o gradiente da direção não é preciso, então a linha de votos não passa exatamente pelo centro da elipse o que força o pico se tornar mais amplo e com valores menores. Além disso, para evitar as instabilidades numéricas são selecionados pontos bem separados aumentando assim a probabilidade de não pertencerem a mesma elipse.

(a) *Acumulador para a Figura 4.11(a).*(b) *Acumulador para a Figura 4.11(b).*Figura 4.17: *Redução do espaço de parâmetros para a TH em elipses.*

4.6 Outras Extensões para a TH

A motivação para estender a TH é clara: manter o desempenho, mas melhorar a velocidade. Existem outras abordagens para reduzir a carga computacional da TH. Essas abordagens procuram melhorar a velocidade e reduzir a utilização de memória considerando regiões menores do espaço acumulador. Esses métodos incluem: TH rápida, TH adaptativa, TH randômica e técnicas piramidais. O principal problema das técnicas que reduzem o espaço de busca para aumentar a velocidade é que a forma errada muitas vezes é extraída (*phantom shape location*). Métodos alternativos incluem Fuzzy TH e abordagens analíticas.

4.7 Considerações Finais deste Capítulo

Este capítulo apresenta as principais técnicas de extração de características em imagens digitais por meio da correspondência de forma que é o núcleo da abordagem baseada em contorno no *video tracking* e a principal ferramenta utilizada neste trabalho. Mostra uma abordagem geral sobre as técnicas de limiarização, *template matching* e sobre as diversas transformadas de Hough e sua otimização.

O próximo capítulo descreve a construção do sistema, ou seja, o software e o hardware

utilizados para aquisição e processamento do *video stream*.

METODOLOGIA, AQUISIÇÃO E PROCESSAMENTO DO VIDEO STREAM

5.1 Introdução

Este capítulo descreve os principais modelos para a simulação da dinâmica pupilar, o modelo de aquisição das imagens, o hardware desenvolvido, as etapas de processamento dos algoritmos implementados e as interfaces do software.

5.2 Padrões Pupilares para o Propósito de Pesquisa e Prática Clínica

Os estudos de Fotiou [7] determinam alguns padrões para a pupilometria dinâmica. A Tabela 5.2 resume alguns desses parâmetros e a Figura 5.1 indica seus respectivos valores em uma saída do protótipo proposto.

Tabela 5.1: *Parâmetros medidos na pupilometria dinâmica.*

Parâmetro	Descrição
$(P/I)_a$	A razão entre o raio da pupila e o raio da íris antes do disparo do flash;
τ_a	O tempo de latência entre o disparo do flash e o início da constrição da pupila;
$(P/I)_b$ e τ_b	A razão P/I e o tempo de latência para a maior constrição da pupila;
$(P/I)_p$ e τ_p	A razão P/I e o tempo de latência do alcance do plateau ($P \approx 75\%$ do valor inicial);

5.3 Modelos para Simulação de Dinâmica Pupilar

Os modelos para a simulação da dinâmica pupilar mais simples consideram como entrada apenas a intensidade luminosa que chega a retina. Alguns modelos, mais elaborados, tratam a variação da pupila como um sistema dinâmico não linear e levam em consideração os estudos fisiológicos e anatômicos.

Normalmente esses modelos representam as curvas de aproximação para um conjunto de dados capturados em experimentos. Mesmo assim, os resultados para as entradas de diferentes intensidades luminosa incidentes na retina ainda são de difícil acesso e em sua maioria incompletos.

Nas próximas seções são apresentados alguns modelos para a dilatação pupilar baseados na intensidade da luz.

5.3.1 Modelo de Moon e Spencer

Moon e Spencer em [37] determinaram a variação do diâmetro da pupila em função da luminância de uma cena com base na aproximação de diversas curvas experimentais de diversos autores, obtendo-se a Equação (5.1):

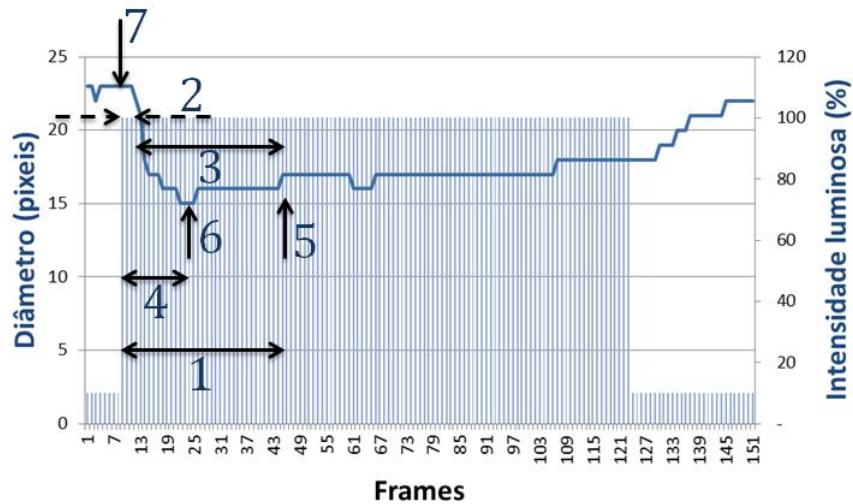


Figura 5.1: *Medidas extraídas do output pupilométrico.* 1-Tempo de latência do alcance do plateau; 2-Tempo de latência para o início da constrição; 3-Duração da constrição; 4-Tempo de latência para a maior constrição; 5-Raio da pupila no alcance do plateau; 6-Raio da pupila para a máxima constrição; 7-Raio da pupila antes do disparo do flash.

$$D = 4,9 - 3,0 \tanh[0,4(\log L + 0,5)] \quad (5.1)$$

onde:

D - diâmetro variando de 2 a 8 mm;

L - luminância de fundo de cena em mLb .

5.3.2 Modelo de Groot e Gebhard

Groot e Gebhard comprovaram em [38], por meio de uma análise mais detalhada dos dados disponíveis sobre a variação do diâmetro pupilar, que a taxa de variação entre seu valor máximo e mínimo não correspondia a uma tangente hiperbólica, ajustando-se assim a Equação (5.1) conforme a Equação (5.2).

$$\log D = 0,8558 - 0,000401(\log L + 8,1)^3 \quad (5.2)$$

onde:

D - diâmetro em mm ;

L - luminância de fundo de cena em mLb .

5.3.3 Modelo de Pokorny e Smith

Pokorny e Smith consideraram em [39] que o estímulo luminoso até então utilizado por outros pesquisadores na determinação da variação do diâmetro pupilar era relativamente incomum com relação à sua composição espectral, pois a luz incidente é controlada pelo tamanho da pupila e deve ser seletivamente atenuada pela lente e pelo pigmento macular assim, para o cálculo do diâmetro da pupila eles utilizaram a Equação (5.3).

$$D = 5 - 3 \tanh(0,4 \log L) \quad (5.3)$$

onde:

D - diâmetro em mm ;

L - luminância de fundo de cena em (cd/m^2) .

A Figura 5.2 mostra as curvas dos modelos de Moon e Spencer, Groot e Gebhart e Pokorny e Smith. As equações estão com as mesmas unidades, pois foram convertidas de mililambert para candelas por metro quadrado. Essa conversão é direta, pois 1 mililambert é igual a 3,18 candelas por metro quadrado.

5.3.4 Modelo de Ellis

Os estudos de Ellis apresentados em [40] determinaram a metodologia e os resultados de uma série de experimentos realizados em 19 voluntários. Nesse trabalho, por meio da variação da intensidade luminosa de um pupilômetro infravermelho, apresenta-se a latência média e as velocidades máximas de contração e dilatação da pupila para uma determinada intensidade luminosa, conforme as Equações (5.4), (5.5) e (5.6), respectivamente.

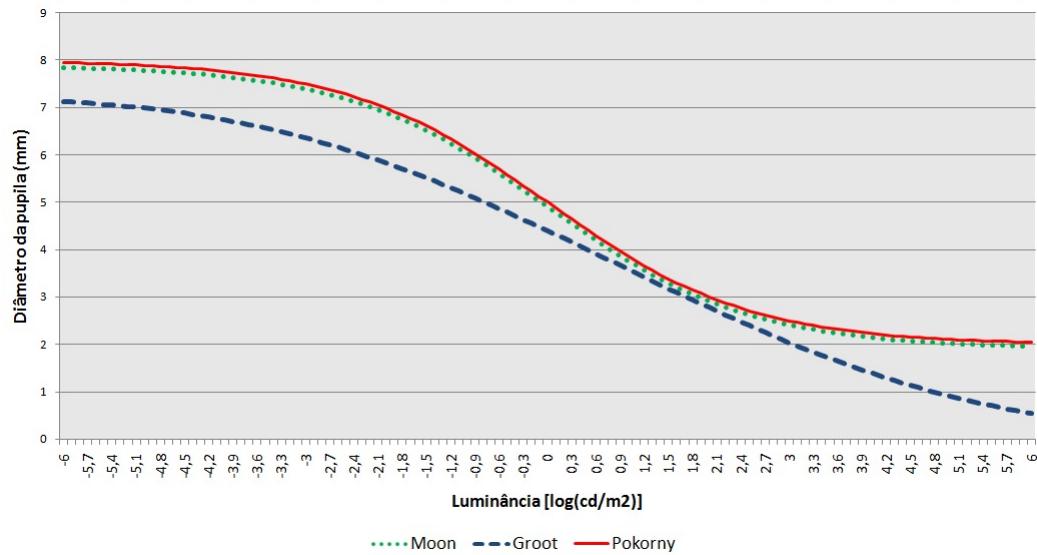


Figura 5.2: Comparação entre os modelos de Moon e Spencer; Groot e Gebhard; e Pokorny e Smith.

$$\tau(I) = 477 - 150,3I + 27,4I^2 \quad (5.4)$$

$$V_c(I) = 0,15 + 2,0I - 0,17I^2 \quad (5.5)$$

$$V_d(I) = 0,16 + 0,72I - 0,07I^2 \quad (5.6)$$

onde:

τ - tempo de latência em ms ;

V_c - velocidade de contração em mm/s ;

V_d - velocidade de dilatação em mm/s ;

I - intensidade luminosa expressa em unidades logarítmicas de cd/m^2 .

5.3.5 Modelo de Link e Stark

No modelo de Link e Stark [16] tanto a intensidade quanto a frequência da fonte de luz (pulsos de onda quadrada) aplicada à íris foram variadas. Assim, por meio desses estudos foi possível encontrar o tempo de latência da pupila como mostrado na Equação (5.7).

$$\tau(R, I) = 253 - 14 \ln(I) + 70R - 29R \ln(I) \quad (5.7)$$

onde:

τ - latência em ms ;

I - luminância medida em fL ;

R - frequência da fonte de luz em Hz .

Outros modelos podem ser encontrados na literatura, como por exemplo o modelo de Longtin e Milton que é bem mais complexo que os descritos anteriormente. Trata-se de um modelo neurofisiológico que considera, além da intensidade da luz atual, a intensidade de um momento anterior, e como é um modelo temporal, também modela os movimentos de híppus por meio de uma Equação *delay-differential*. Para maiores detalhes consultar [10].

5.4 Sincronizando os *Frames* do *Video Stream* com a Fonte de Iluminação

Para sincronizar a informação de luminância com um determinado *frame* da sequência de vídeo foi elaborado um dispositivo micro controlado de tal forma que, a partir das curvas características do LED, conforme os exemplos das Figuras 5.3(a), (b) e (c) (fornecidas pelo fabricante), e do controle de sua corrente direta, fosse possível levantar os parâmetros de intensidade luminosa e comprimento de onda em cada instante.

A Figura 5.4 mostra o esquema do circuito construído para o protótipo proposto neste trabalho. Foi utilizado o microcontrolador PIC 18F4550 e 4 LEDs de 5 mm do modelo

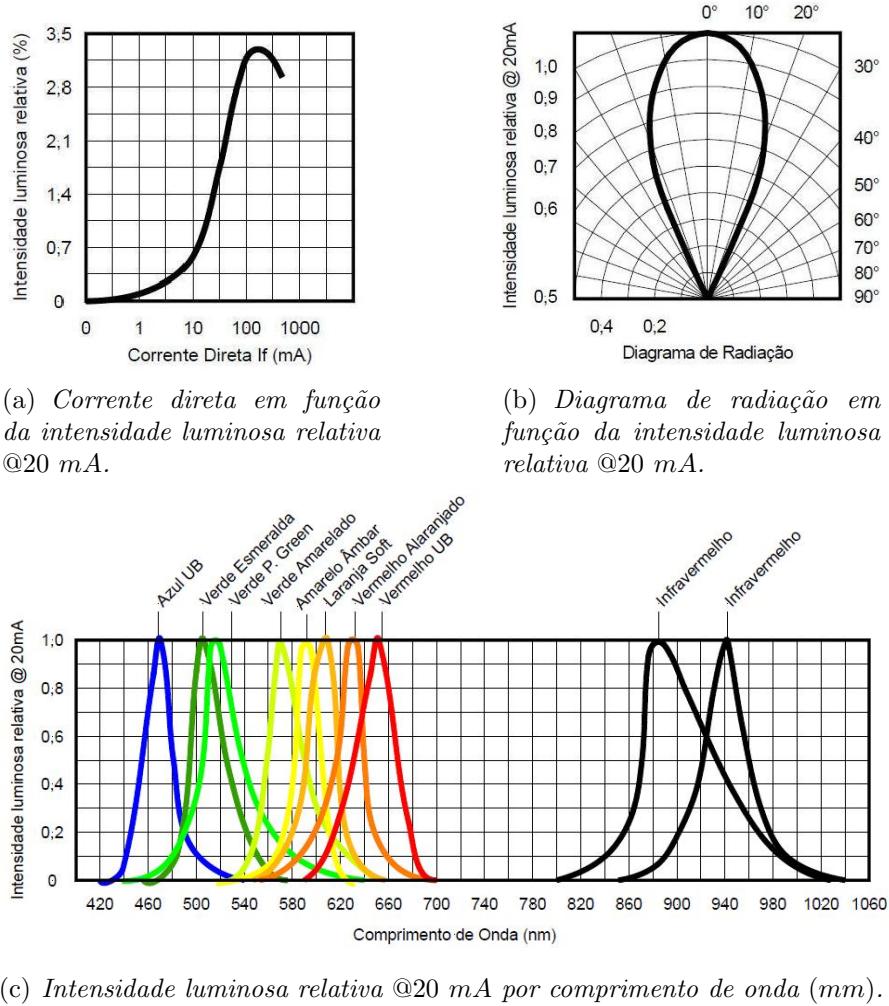


Figura 5.3: Curvas características do LED modelo LX61 da Cinestec Comp. Eletrônicos.

LC503 da marca Cree nas cores vermelho, verde, azul e branco. A rotina interna do PIC e outros detalhes do circuito são apresentados na seção A.1 do Apêndice A.

A Figura 5.5 mostra a interface escrita em C sharp que permite ao usuário controlar a saída de modulação por largura de pulso (PWM) do micro controlador e a câmera. Assim, o software associa a cada *frame* do vídeo, a largura do pulso PWM aplicado, em cada instante no sistema de iluminação. As alterações dessa largura de pulso podem ser realizadas em qualquer instante durante a gravação do *video stream*.

O PWM é um método popular de controle de luminosidade de LED devido à linearidade do ajuste de brilho. No PWM, a corrente efetiva é variada pela mudança da largura

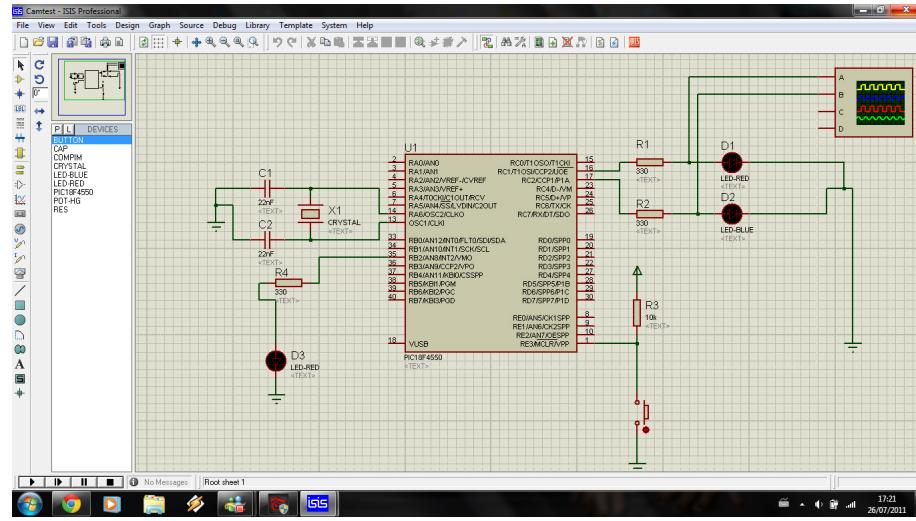


Figura 5.4: Esquema do hardware de controle.

do pulso de corrente (t), e a frequência de modulação é constante e igual a $(1/T)$. A temperatura da junção dos LEDs é um parâmetro importante porque tem uma influência significativa sobre a cor, sobre o fluxo luminoso e sobre a vida útil dos LEDs. As características fotométricas, colorimétricas e térmicas podem ser investigadas durante o controle de luminosidade via PWM. É usual uma alta frequência modulada para impedir a percepção de pulsos individuais de luz, causando o efeito de cintilação (*flicker*) [41]. Normalmente, a luz de saída é controlada com potência entre 5% e 95% de seu valor máximo.

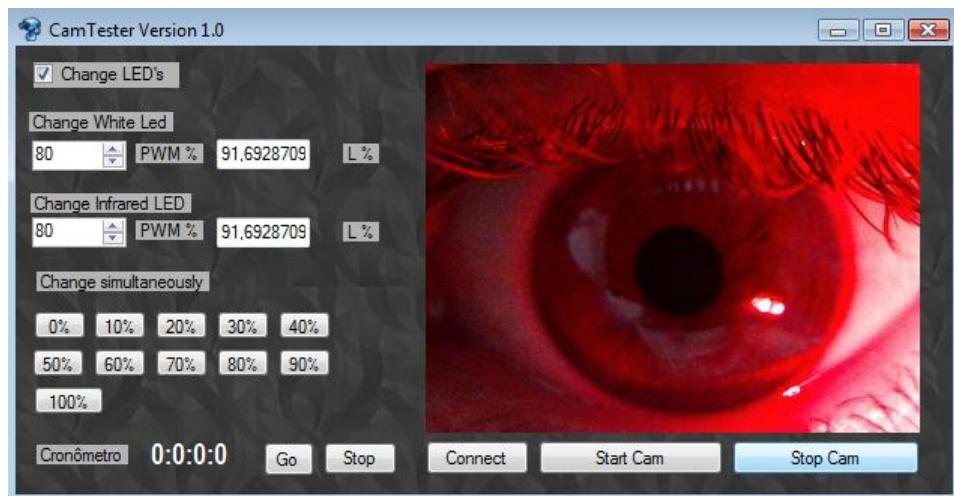


Figura 5.5: Interface para controle do PWM (controle da intensidade luminosa).

Todo o processo de sincronismo pode ser visualizado no diagrama de blocos da Figura 5.6. Por meio da interface mostrada na Figura 5.5 o usuário fornece ao sistema as informações iniciais da fonte de iluminação utilizada, como por exemplo, cor e potência máxima.

Em seguida, o usuário estabelece a porcentagem inicial da largura do PWM ou a porcentagem de brilho percebido. Então, cada *frame* capturado passa a ter a informação inicial da largura do PWM.

Se a potência inicial (%PWM) for modificada pelo usuário o sistema atualiza o valor associando-o ao próximo *frame*. O *frame* capturado é então armazenado e enviado para o sistema de rastreamento descrito na Seção 5.5.

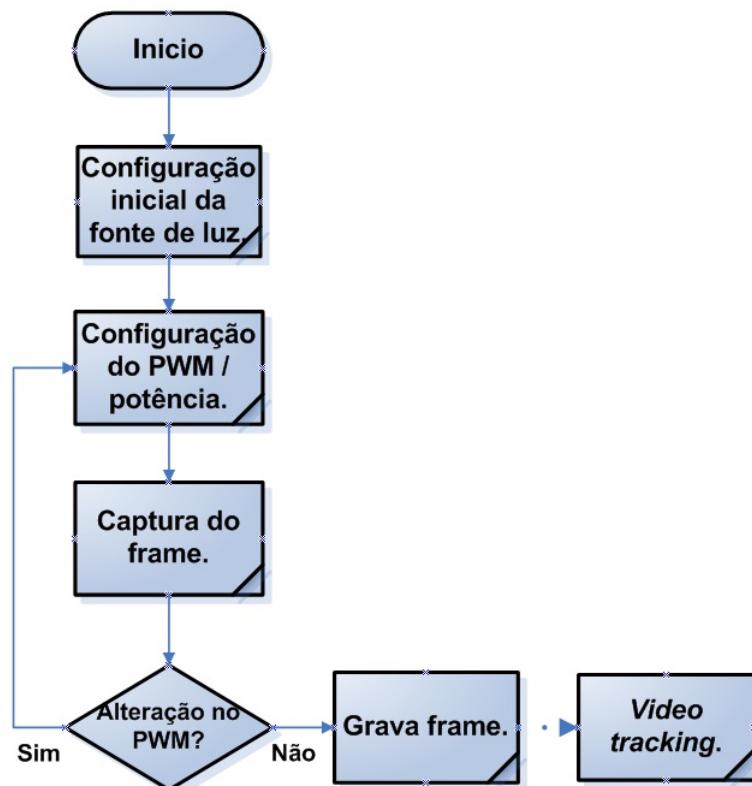


Figura 5.6: Diagrama de blocos do processo de sincronismo entre os frames e o sistema de iluminação.

5.4.1 Controle do Brilho

Outra informação disponível na interface em C Sharp é a percepção visual. Para mudar o brilho de qualquer fonte de luz é necessário levar em conta como o olho humano percebe essa alteração. o ser humano possui melhor sensibilidade em baixa luminância do que em alta, assim fica evidente que a percepção humana à alteração de brilho não é linear.

A largura de pulso PWM controla a tensão aplicada no LED e consequentemente sua potência. No protótipo proposto pode-se variar a largura do pulso de 0 a 100% com uma resolução de 10 bits. Contudo, para que as variações das larguras de pulso do PWM correspondam às variações de brilho no LED, percebida pelo olho humano, é necessário compensar ou corrigir seu valor. Assim, o relatório CIE 1931 traz a correção, conforme as Equações (5.8) e (5.9).

$$L^* = 116 \left(\frac{Y}{Y_n} \right)^{\frac{1}{3}} - 16 \quad \text{para} \quad \frac{Y}{Y_n} > 0,008856 \quad (5.8)$$

e

$$L^* = 903,3 \left(\frac{Y}{Y_n} \right) \quad \text{para} \quad \frac{Y}{Y_n} \leq 0,008856 \quad (5.9)$$

onde:

L^* - função de potência de luminosidade modificada (CIE 1976);

Y - luminância CIE proporcional a intensidade (logo, proporcional à largura do pulso PWM);

Y_n - luminância de referência do branco (ou máxima intensidade - fundo de escala do PWM).

Dessa forma, para uma largura de pulso de 50%, por exemplo, tem-se metade da intensidade luminosa do LED, porém um brilho percebido obtido pelas Equações (5.8) ou (5.9). Finalmente, o sistema armazena cada *frame* referenciando em seu nome a ordem em que ele está no *video stream*, o valor PWM e o brilho percebido atual.

5.5 Processamento do *Video Stream*

Com o controle ótimo da fonte de iluminação aplicada à íris pode-se então elaborar vários conjuntos de ciclos pupilares, como por exemplo, o *pupil escape* e o *pupil capture*, já mencionados. O problema agora é determinar a resposta da pupila que, diferentemente de como acontece em imagens estáticas, exige um método de captura e de extração de características mais robusto em tempo de vídeo, ou seja, mais veloz, imune a travamentos e oclusões, além de boa sensibilidade e precisão.

Com base no que foi exposto nos Capítulos 3 e 4 deste trabalho é possível que a abordagem de *video tracking* com extração de características baseada em contorno seja a solução do problema acima.

5.5.1 Pré processamento

A câmera de vídeo utilizada no protótipo proposto é uma *webcam* da marca Logitech, conforme apresentado no apêndice B.2. Essa câmera tem capacidade para $1Mbps$ na resolução de tela de 1280×720 . Devido às limitações da interface de comunicação, tamanho e resolução da imagem e do hardware utilizado, a taxa de FPS, em média é em torno de 21 FPS. A Figura 5.7 apresenta o diagrama de blocos do algoritmo de processamento de imagens que é executado em forma de loop durante toda a duração do processo de captura do protótipo proposto neste trabalho.

De forma simplificada, na Figura 5.7 o primeiro bloco controla os limiares de busca, ou seja, o raio máximo, o raio mínimo e o comprimento mínimo da circunferência no caso de obstrução. O bloco seguinte controla o fluxo de imagem a ser processado. A seguir, cada quadro de entrada é transformado para sua versão em escala de cinza com o intuito de reduzir o custo computacional.

No quarto bloco e no quinto bloco aplica-se o algoritmo de Canny para detecção das bordas e uma suavização da imagem para a eliminação dos pixeis esparsos, respectivamente. A Figura 5.8(a) apresenta a imagem de um *frame* qualquer em escala de cinza, a Figura

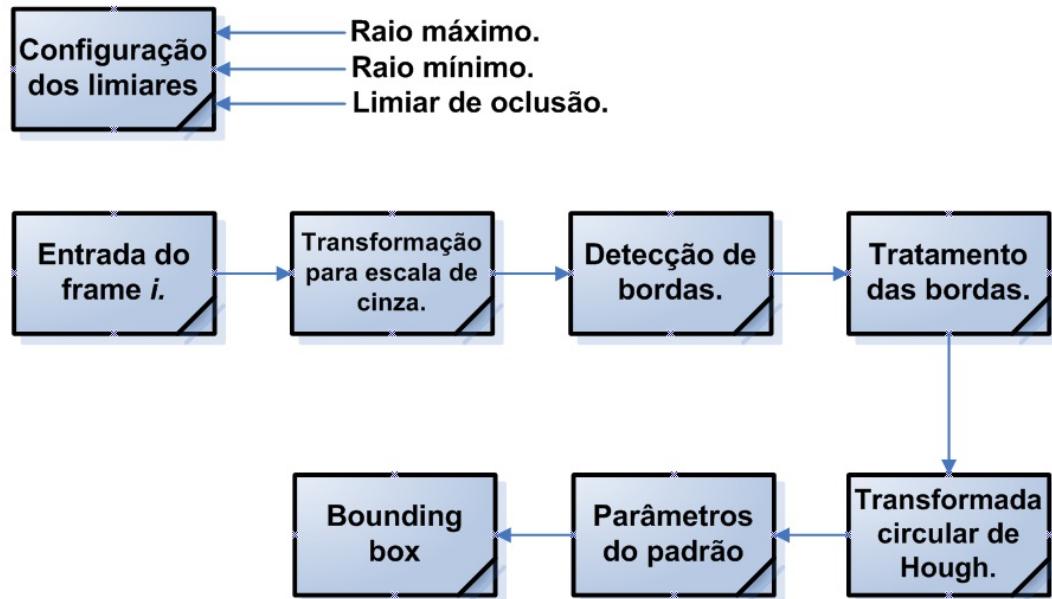
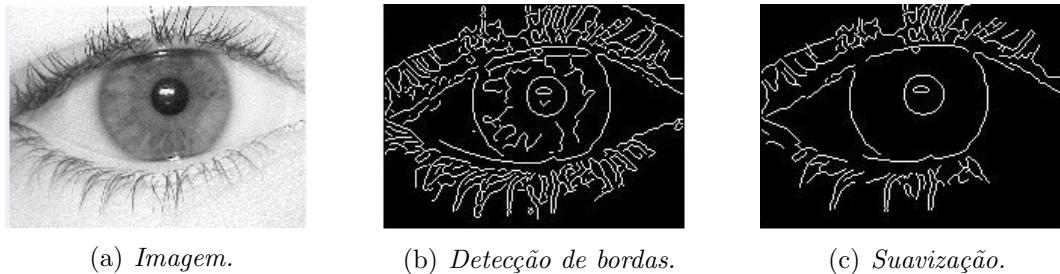


Figura 5.7: Diagrama de blocos do tracking da pupila. Algoritmo utilizado na detecção do raio e da posição da pupila do protótipo proposto.

5.8(b) a imagem após a detecção de bordas e a Figura 5.8(c) a imagem suavizada.



(a) *Imagen.*

(b) *Detecção de bordas.*

(c) *Suavização.*

Figura 5.8: *Detecção de bordas por Canny.*

Na Figura 5.7, após a suavização da imagem, aplica-se a transformada circular de Hough, conforme descrito no Capítulo 4 deste trabalho, para a detecção da pupila segundo os limiares estabelecidos no primeiro bloco. Os parâmetros da pupila são dados pelo círculo que está dentro dos limites admissíveis do raio. Esses valores são configurados em tempo de execução pela interface do algoritmo, construída em Matlab, conforme Figura 5.9.

No bloco “Parâmetros do padrão” todos os valores gerados pelo sistema proposto são armazenados em uma matriz $R(i, j)$. Nesta matriz encontram-se os valores da posição do

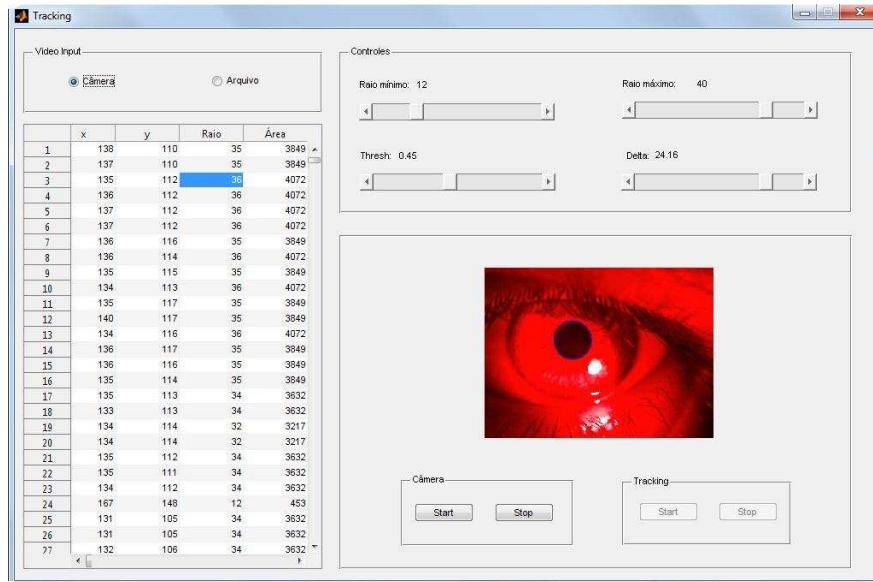


Figura 5.9: Interface para ajuste dos parâmetros da pupila.

objeto ((x, y) em pixels), raio do objeto (em pixels), posição na sequência de vídeo (*frame*), % do PWM (intensidade luminosa) e cor do LED.

A finalidade do último bloco da Figura 5.7 é delimitar o objeto de interesse em cada *frame* (*boundingbox*) e atualizar seu raio e suas coordenadas de posição.

5.6 Considerações Finais deste Capítulo

Este capítulo descreve os principais modelos para simulação da dinâmica pupilar e as estratégias de hardware e software utilizadas nessa dissertação para a extração de características em imagens digitais através de um *video stream*.

O próximo capítulo descreve os experimentos realizados e os resultados obtidos. Finalmente, são realizadas as conclusões desses resultados.

APLICAÇÃO DO SISTEMA PROPOSTO

6.1 Introdução

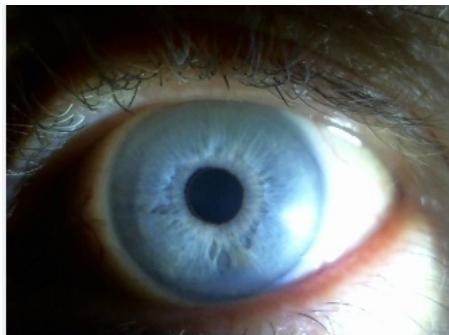
Este capítulo descreve os experimentos realizados e os resultados obtidos pelo protótipo proposto neste trabalho. Este capítulo também mostra os resultados obtidos pelo algoritmo proposto no tratamento de oclusões. Finalmente, são realizadas as conclusões desses resultados.

6.2 Aquisição dos *Video Streams*

Para a realização dos experimentos foram capturadas sequências de vídeo contendo a região da íris de 8 voluntários, sendo 3 (37,5%) do sexo masculino e 5 (62,5%) do sexo feminino. A idade média dos voluntários variou de 20 a 68 anos, com média de 30,75 anos e mediana de 27,5 anos. Nessas sequências, após um tempo para a acomodação da pupila, um degrau de intensidade variável foi aplicado e logo em seguida a intensidade inicial da fonte de luz foi restabelecida.

Além disso, cada sequência foi repetida com LEDs de diferentes cores, logo diferentes comprimentos de onda. A Figura 6.1 ilustra algumas aquisições.

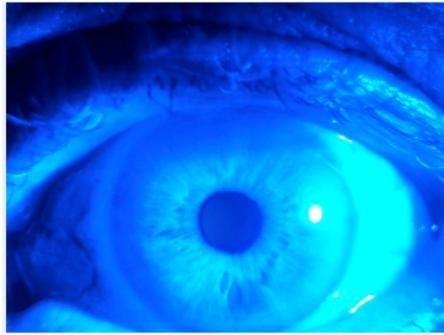
A finalidade dessa condição experimental é identificar e estudar (conforme os modelos



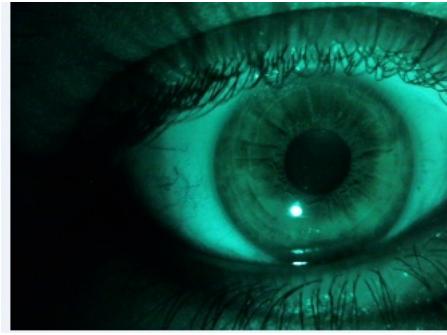
(a) Comprimento de onda entre 400 nm e 800 nm.



(b) Comprimento de onda dominante $\lambda_D = 624(@i_F = 20 \text{ mA})$.



(c) Comprimento de onda dominante $\lambda_D = 470(@i_F = 20 \text{ mA})$.



(d) Comprimento de onda dominante $\lambda_D = 527(@i_F = 20 \text{ mA})$.

Figura 6.1: Fontes de iluminação com diferentes comprimentos de onda.

de simulação discutidos) a relação entre o movimento pupilar e a luminância, além da latência (tempo de resposta do sistema nervoso autônomo) e das velocidades de contração e dilatação.

As imagens foram tomadas em diferentes horários, a maioria delas adquiridas no laboratório de ótica, sala 1X47, da Faculdade de Física da Universidade Federal de Uberlândia. A única exigência, em todos os experimentos, foi a de que todos os voluntários passassem por um período de acomodação de no mínimo 2 minutos, conforme os trabalhos de [7] e [42]. Nesse período de acomodação a iluminação ambiente era a menor possível com a finalidade de obter a máxima abertura da pupila (midríase).

Após o período de acomodação a imagem de uma das pupilas foi registrada por aproximadamente um minuto antes do disparo do degrau enquanto a outra, separadamente,

permanecia sob baixa iluminação. Dessa forma, o diâmetro de saída do pupilômetro permanece praticamente constante conforme mostra a Figura 6.2.

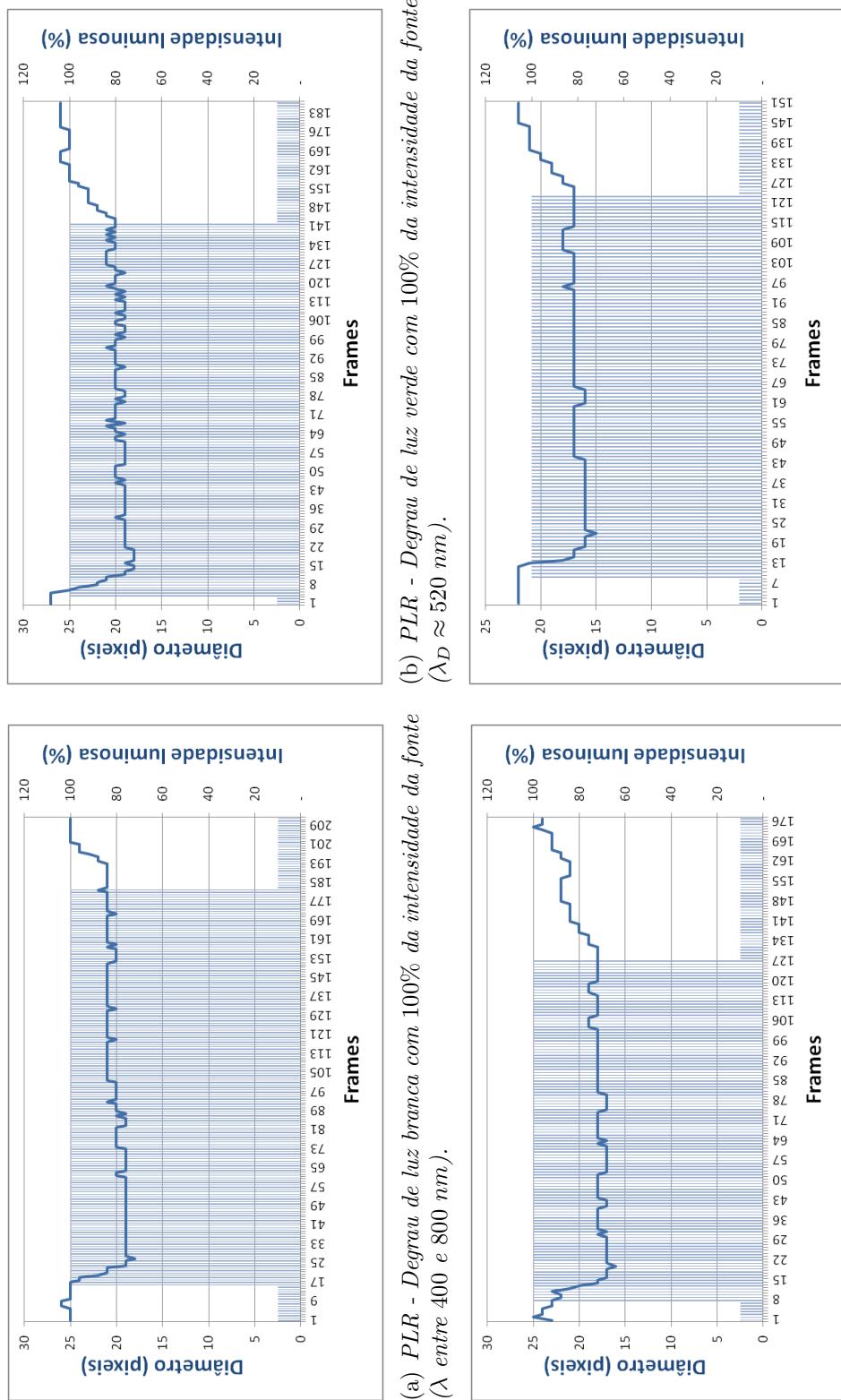


Figura 6.2: Pupil light reflex (PLR) com fontes de luz de diferentes intensidades e comprimentos de onda.

Nos gráficos da Figura 6.2 pode-se observar no eixo horizontal principal à esquerda a linha de variação do diâmetro da pupila em pixels e no eixo horizontal secundário à direita a intensidade do degrau de iluminação aplicado em porcentagem. No eixo vertical encontra-se a posição do frame ao longo do vídeo.

Com as informações obtidas pelo sistema proposto e representadas na Figura 5.1 do Capítulo 5 deste trabalho é possível extrair os padrões descritos na Tabela 5.2 (*Parâmetros medidos na pupilometria dinâmica segundo [7]*) e comparar o comportamento das imagens capturadas com as equações que representam os modelos pupilares descritas na seção 5.3 - *Modelos para simulação de dinâmica pupilar* do Capítulo 5 deste trabalho.

A análise estatística inclui o cálculo de média, valor máximo e mínimo e desvio padrão para todas as variáveis.

6.3 Resultados Obtidos

Em todos os voluntários que participaram dos experimentos a resposta pupilar teve forma similar à Figura 6.2, ou seja, devido um estímulo ocorre constrição após um período de latência. Logo em seguida à constrição ocorre dilatação até que a pupila alcance um patamar. Assim que o estímulo é retirado a pupila volta a dilatar podendo ou não retornar ao diâmetro inicial.

Durante o PLR a amplitude aumenta com o aumento da intensidade do estímulo luminoso, porém apesar dessa relação, os dados não apresentam comportamento linear, atingindo amplitudes mínimas para intensidades intermediárias. O mesmo comportamento pode ser visto na dilatação.

Por outro lado, os tempos de latência antes do degrau e o de máxima constrição decrescem com o aumento do estímulo luminoso e como antes, apresentam comportamento não linear.

O tamanho inicial da pupila em relação à íris, antes do degrau, ficou entre 24 e 39%, e os maiores valores referentes à iluminação em torno do vermelho (550 – 650 nm) e do

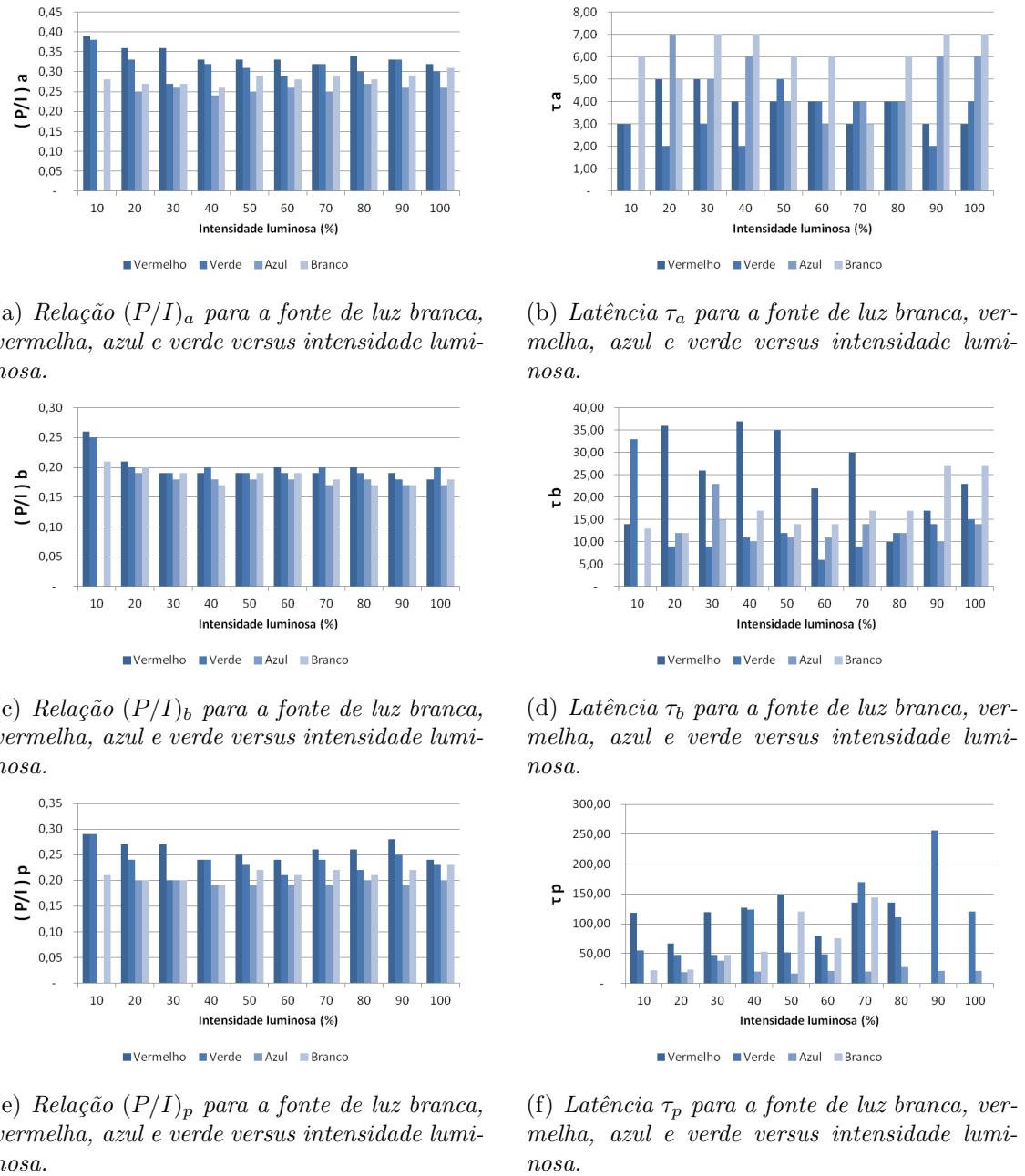


Figura 6.3: Comparação entre as saídas pupilográficas para os diversos comprimentos de onda λ .

verde ($450 - 600\text{ nm}$). A mesma relação para a situação de máxima constrição ficou entre 17 e 26%. A comparação entre os valores da Figura 6.3(a) e 6.3(c) comprova que a pupila contrai mais para a luz com baixas frequências. As funções que determinam a sensibilidade dos comprimentos de onda são descritas pela norma CIE/ISO 10527.

O tempo médio de latência variou de 388 a 110 ms , enquanto o tempo médio de latência para a máxima constrição variou de 2 s a 500 ms . Ambos os tempos apresentaram comportamento não linear em função da intensidade, onde as curvas mais acentuadas foram devido à iluminação em torno do vermelho ($550 - 650\text{ nm}$) e do verde ($450 - 600\text{ nm}$) conforme mostra as Figuras 6.3(b) e 6.3(d).

Pode-se observar na Figura 6.3(f) que o tempo de alcance do *plateau* (75% do diâmetro inicial) aumenta consideravelmente com o aumento da intensidade chegando ao ponto de não ser determinado para as maiores intensidades do branco e do vermelho. Especificamente para o azul verifica-se um comportamento não linear, com valores menores para as intensidades intermediárias.

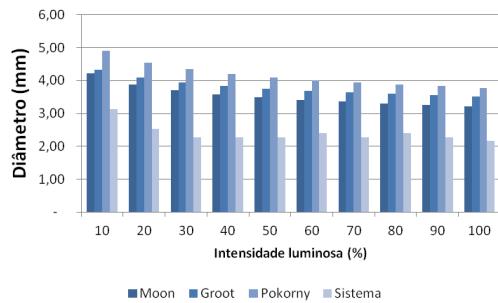
Para a fonte de luz azul o algoritmo de detecção não foi capaz de encontrar as bordas do objeto devido ao baixo contraste entre as regiões vizinhas.

O método e o equipamento desenvolvidos neste trabalho apresentaram menor desvio para os resultados de diâmetro quando comparados ao modelo de Moon e Spencer [37] para todos os comprimentos de onda, no entanto o erro foi menor para a fonte de iluminação em torno do verde (450 a 600 nm). O maior desvio ocorreu para o modelo de Pokorny e Smith [39] em todos os comprimentos de onda e ainda mais acentuado para a fonte azul (400 a 600 nm).

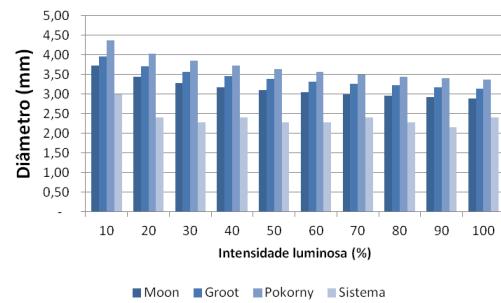
Considerando, conforme descrito em [39], que o diâmetro da pupila é afetado por diversos fatores intrínsecos e extrínsecos, tais como: idade, batimento cardíaco, respiração, dor, nível de iluminação e composição espectral e que as únicas variáveis controladas no experimento foram as duas últimas, observou-se que o sistema é sensível a ponto de detectar oscilações do movimento *hippus*.

É importante ressaltar que as variações causadas por esses fatores reforçam ainda mais a necessidade de conhecimento personalizado de cada pupila para a programação de cirur-

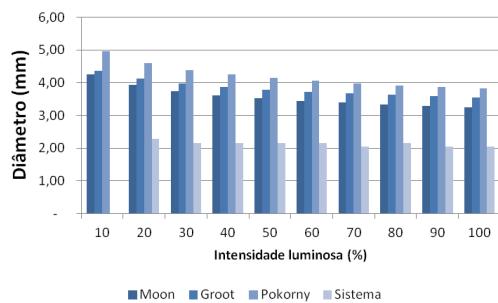
gias refrativas, uma vez que as cirurgias são realizadas na córnea e tem uma zona óptica característica determinada pela área de aplicação central do laser, conforme exemplificado na Figura 1.1 do Capítulo 1 deste trabalho.



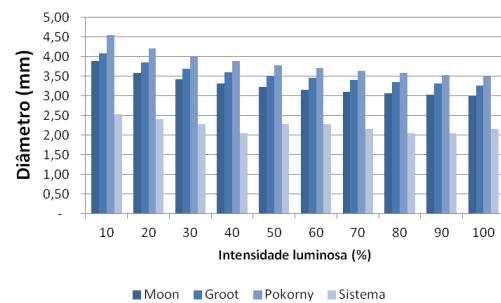
(a) Diâmetro da pupila D em mm para a fonte de luz branca versus intensidade luminosa.



(b) Diâmetro da pupila D em mm para a fonte de luz vermelha versus intensidade luminosa.



(c) Diâmetro da pupila D em mm para a fonte de luz azul versus intensidade luminosa.



(d) Diâmetro da pupila D em mm para a fonte de luz verde versus intensidade luminosa.

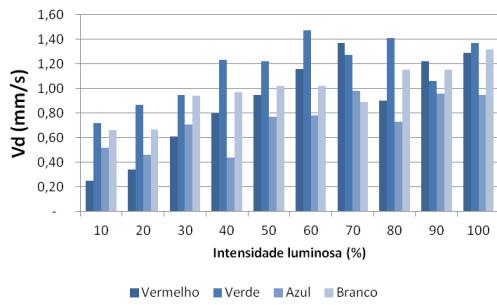
Figura 6.4: Comparação entre as saídas pupilométricas para os diversos comprimentos de onda λ .

A velocidade de constrição e a velocidade de dilatação aumentam com o aumento da intensidade do estímulo. Ambas apresentam o mesmo comportamento observado no modelo de Ellis, ou seja, alcançam seus valores máximos para as máximas amplitudes.

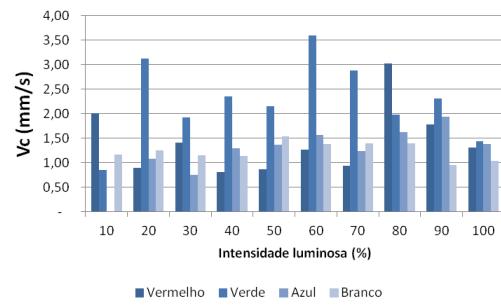
A velocidade de constrição variou de 0,75 a 3,6 mm/s e os maiores valores foram nas intensidades intermediarias nas iluminações em torno do vermelho (550 – 650 nm) e do verde (450 – 600 nm), comprovando a relação com a máxima amplitude.

A velocidade de dilatação variou de 0,25 a 1,47 mm/s sendo os maiores valores nas intensidades intermediarias referentes às iluminações em torno do vermelho (550 – 650 nm) e do verde (450 – 600 nm), comprovando também a relação com a máxima amplitude.

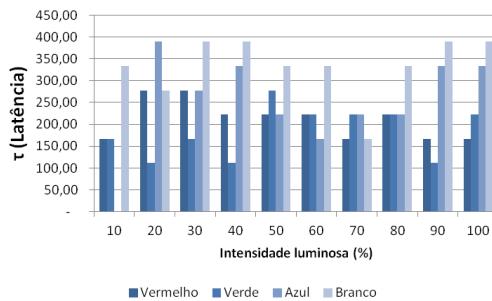
Os valores encontrados para as velocidades acompanham o comportamento esperado, porém a distância entre eles e o valores esperados, obtidos pelo modelo de Ellis, exige um estudo mais aprofundado. Considera-se o espaço amostral utilizado como uma limitação além da simplicidade dos equipamentos.



(a) *Velocidade de dilatação da pupila V_d em mm/s para a fonte de luz vermelha, verde, azul e branca versus intensidade luminosa.*



(b) *Velocidade de contração da pupila V_c em mm/s para a fonte de luz vermelha, verde, azul e branca versus intensidade luminosa.*



(c) *Tempo de latência da pupila τ em ms para a fonte de luz vermelha, verde, azul e branca versus intensidade luminosa.*

Figura 6.5: *Comparação entre as saídas pupilométricas para os diversos comprimentos de onda λ .*

Devido aos riscos existentes não foram realizados neste trabalho experimentos relacionando a intensidade com o tempo de exposição à luz. Segundo, Alpern et al [43] a variação de intensidade da luz ou a sua duração segue a lei de reciprocidade (intensidade x tempo de exposição = constante).

Alpern et al [43] também afirmaram que não é necessário ativar os cones na retina para que ocorra reação da pupila, o tempo de latência nos experimentos decaiu conforme aumentava-se a intensidade e duração e a curva relacionando a energia do estímulo com a

mudança no tamanho da pupila é dividida em duas partes. Esta divisão ocorre quando, além dos bastonetes (visão escotópica), os cones sentem o estímulo de luz (visão fotópica). No entanto, a participação em detalhes dos bastonetes durante o PLR ainda é difícil de ser explicada.

Com base nos *datasheets* o sistema proposto neste trabalho opera nas visões mesópica e fotópica, variando de $1,1\text{ cd/m}^2$ a 34 cd/m^2 . O olho humano usa escotópico puro com luminância abaixo de $0,034\text{ cd/m}^2$, e usa fotópico puro acima de $3,4\text{ cd/m}^2$.

As Tabelas de 6.3 a 6.3 mostram em detalhes os valores utilizados na construção dos gráficos das Figuras 6.3, 6.4 e 6.5.

Tabela 6.1: Valores dos parâmetros encontrados em função da intensidade luminosa para LED branco.

Comprimento de onda $\lambda = 400 \sim 800 \text{ nm}$											
I(%)		10	20	30	40	50	60	70	80	90	100
$(P/I)_a$	Mín	0,25	0,23	0,24	0,24	0,26	0,25	0,26	0,24	0,23	0,27
	Máx	0,33	0,31	0,32	0,32	0,34	0,34	0,32	0,32	0,33	0,33
	Méd	0,28	0,27	0,27	0,26	0,29	0,28	0,29	0,28	0,29	0,31
τ_a	Mín	4,00	3,00	9,00	5,00	3,00	4,00	2,00	2,00	5,00	4,00
	Máx	5,00	3,00	14,00	8,00	7,00	7,00	5,00	8,00	9,00	8,00
	Méd	6,00	5,00	7,00	7,00	6,00	6,00	3,00	6,00	7,00	7,00
$(P/I)_b$	Mín	0,20	0,19	0,17	0,16	0,18	0,17	0,17	0,17	0,16	0,17
	Máx	0,24	0,24	0,21	0,23	0,24	0,24	0,25	0,24	0,25	0,22
	Méd	0,21	0,20	0,19	0,17	0,19	0,19	0,18	0,18	0,17	0,18
τ_b	Mín	8,00	4,00	16,00	10,00	8,00	4,00	10,00	8,00	22,00	20,00
	Máx	18,00	15,00	28,00	24,00	15,00	15,00	22,00	22,00	29,00	29,00
	Méd	13,00	12,00	15,00	17,00	14,00	14,00	17,00	17,00	27,00	27,00
$(P/I)_p$	Mín	0,20	0,18	0,19	0,15	0,18	0,16	0,20	0,18	0,15	0,18
	Máx	0,24	0,22	0,23	0,25	0,25	0,22	0,26	0,23	0,20	0,25
	Méd	0,21	0,20	0,20	0,19	0,22	0,21	0,22	0,21	0,22	0,23
τ_p	Mín	15,00	8,00	22,00	44,00	91,00	55,00	112,00	88,00	116,00	-
	Máx	32,00	54,00	52,00	55,00	154,00	87,00	166,00	-	-	-
	Méd	22,00	23,00	48,00	53,00	120,00	75,00	144,00	-	-	-

Tabela 6.2: Valores dos parâmetros encontrados em função da intensidade luminosa para LED vermelho.

Comprimento de onda $\lambda = 550 \sim 650 \text{ nm}$											
	I(%)	10	20	30	40	50	60	70	80	90	100
$(P/I)_a$	Mín	0,33	0,33	0,35	0,32	0,33	0,30	0,31	0,34	0,29	0,30
	Máx	0,38	0,39	0,38	0,35	0,34	0,34	0,33	0,35	0,39	0,33
	Méd	0,39	0,36	0,36	0,33	0,33	0,33	0,32	0,34	0,33	0,32
τ_a	Mín	4,00	3,00	4,00	2,00	2,00	3,00	1,00	3,00	3,00	1,00
	Máx	5,00	7,00	7,00	5,00	5,00	4,00	4,00	6,00	5,00	2,00
	Méd	3,00	5,00	5,00	4,00	4,00	3,00	4,00	3,00	3,00	3,00
$(P/I)_b$	Mín	0,23	0,19	0,16	0,18	0,16	0,19	0,18	0,20	0,18	0,15
	Máx	0,27	0,22	0,22	0,22	0,20	0,22	0,22	0,20	0,19	0,19
	Méd	0,26	0,21	0,19	0,19	0,19	0,20	0,19	0,20	0,19	0,18
τ_b	Mín	15,00	13,00	16,00	11,00	11,00	6,00	13,00	6,00	17,00	11,00
	Máx	18,00	35,00	29,00	47,00	38,00	19,00	39,00	22,00	17,00	35,00
	Méd	14,00	36,00	26,00	37,00	35,00	22,00	30,00	10,00	17,00	23,00
$(P/I)_p$	Mín	0,25	0,20	0,24	0,22	0,20	0,19	0,24	0,19	0,25	0,19
	Máx	0,30	0,29	0,32	0,27	0,31	0,27	0,29	0,29	0,29	0,28
	Méd	0,29	0,27	0,27	0,24	0,25	0,24	0,26	0,26	0,28	0,24
τ_p	Mín	99,00	63,00	86,00	79,00	79,00	65,00	56,00	88,00	45,00	105,00
	Máx	125,00	81,00	130,00	182,00	167,00	261,00	142,00	144,00	-	-
	Méd	118,00	67,00	119,00	127,00	148,00	80,00	135,00	135,00	-	-

Tabela 6.3: Valores dos parâmetros encontrados em função da intensidade luminosa para LED verde.

Comprimento de onda $\lambda = 450 \sim 600 \text{ nm}$											
I(%)		10	20	30	40	50	60	70	80	90	100
$(P/I)_a$	Mín	0,36	0,30	0,26	0,30	0,28	0,27	0,32	0,28	0,31	0,28
	Máx	0,39	0,34	0,27	0,33	0,32	0,31	0,32	0,31	0,33	0,33
	Méd	0,38	0,33	0,27	0,32	0,31	0,29	0,32	0,30	0,33	0,30
τ_a	Mín	2,00	1,00	2,00	2,00	2,00	2,00	3,00	2,00	2,00	2,00
	Máx	6,00	4,00	3,00	2,00	7,00	5,00	4,00	5,00	3,00	5,00
	Méd	3,00	2,00	3,00	2,00	5,00	4,00	4,00	4,00	2,00	4,00
$(P/I)_b$	Mín	0,21	0,19	0,17	0,19	0,17	0,18	0,18	0,17	0,17	0,19
	Máx	0,26	0,20	0,21	0,20	0,20	0,21	0,21	0,20	0,20	0,22
	Méd	0,25	0,20	0,19	0,20	0,19	0,19	0,20	0,19	0,18	0,20
τ_b	Mín	21,00	6,00	7,00	7,00	11,00	5,00	9,00	10,00	9,00	14,00
	Máx	38,00	12,00	12,00	14,00	14,00	8,00	9,00	15,00	15,00	15,00
	Méd	33,00	9,00	9,00	11,00	12,00	6,00	9,00	12,00	14,00	15,00
$(P/I)_p$	Mín	0,25	0,17	0,18	0,22	0,19	0,19	0,19	0,19	0,17	0,18
	Máx	0,33	0,26	0,21	0,28	0,24	0,22	0,26	0,23	0,22	0,25
	Méd	0,29	0,24	0,20	0,24	0,23	0,21	0,24	0,22	0,25	0,23
τ_p	Mín	35,00	32,00	36,00	99,00	28,00	44,00	127,00	96,00	182,00	96,00
	Máx	68,00	52,00	52,00	153,00	65,00	75,00	202,00	166,00	301,00	134,00
τ_p	Méd	55,00	48,00	48,00	124,00	52,00	49,00	170,00	111,00	256,00	120,00

Tabela 6.4: Valores dos parâmetros encontrados em função da intensidade luminosa para LED azul.

Comprimento de onda $\lambda = 400 \sim 600 \text{ nm}$											
I(%)		10	20	30	40	50	60	70	80	90	100
$(P/I)_a$	Mín	-	0,20	0,21	0,23	0,24	0,23	0,23	0,26	0,24	0,25
	Máx	-	0,30	0,29	0,27	0,29	0,30	0,28	0,27	0,28	0,29
	Méd	-	0,25	0,26	0,24	0,25	0,26	0,25	0,27	0,26	0,26
τ_a	Mín	-	6,00	4,00	4,00	3,00	2,00	3,00	3,00	5,00	4,00
	Máx	-	9,00	7,00	7,00	5,00	5,00	5,00	6,00	7,00	7,00
	Méd	-	7,00	5,00	6,00	4,00	3,00	4,00	4,00	6,00	6,00
$(P/I)_b$	Mín	-	0,18	0,18	0,16	0,16	0,17	0,16	0,17	0,16	0,16
	Máx	-	0,21	0,22	0,21	0,19	0,21	0,20	0,22	0,20	0,20
	Méd	-	0,19	0,18	0,18	0,18	0,18	0,17	0,18	0,17	0,17
τ_b	Mín	-	8,00	6,00	6,00	8,00	9,00	9,00	6,00	7,00	11,00
	Máx	-	14,00	17,00	15,00	12,00	12,00	15,00	14,00	11,00	15,00
	Méd	-	12,00	23,00	10,00	11,00	11,00	14,00	12,00	10,00	14,00
$(P/I)_p$	Mín	-	0,14	0,16	0,13	0,14	0,16	0,14	0,15	0,15	0,18
	Máx	-	0,22	0,23	0,20	0,20	0,21	0,23	0,23	0,22	0,23
	Méd	-	0,20	0,20	0,19	0,19	0,19	0,19	0,20	0,19	0,20
τ_p	Mín	-	16,00	30,00	16,00	15,00	15,00	18,00	22,00	12,00	18,00
	Máx	-	25,00	42,00	28,00	27,00	28,00	26,00	30,00	30,00	29,00
τ_p	Méd	-	19,00	38,00	20,00	17,00	21,00	20,00	27,00	21,00	21,00

Tabela 6.5: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 400 \sim 800 \text{ nm}$							
	Mín	Diâmetro (mm)			Máx	Desvio		
		Moon	Groot	Pokorny		Moon	Groot	Pokorny
10%	2,40	3,89	4,09	4,55	2,88	1,37	1,57	2,03
20%	2,28	3,58	3,84	4,20	2,88	1,18	1,44	1,80
30%	2,04	3,41	3,69	4,01	2,52	1,13	1,41	1,73
40%	1,92	3,30	3,59	3,88	2,76	1,26	1,55	1,84
50%	2,16	3,22	3,51	3,78	2,88	0,94	1,23	1,50
60%	2,04	3,16	3,45	3,70	2,88	0,88	1,17	1,42
70%	2,04	3,11	3,39	3,64	3,00	0,95	1,23	1,48
80%	2,04	3,06	3,35	3,58	2,88	1,02	1,31	1,54
90%	1,92	3,02	3,30	3,54	3,00	0,98	1,26	1,50
100%	2,04	2,99	3,27	3,49	2,64	0,83	1,11	1,33

Tabela 6.6: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 550 \sim 650 \text{ nm}$							
	Mín	Diâmetro (mm)			Máx	Desvio		
		Moon	Groot	Pokorny		Moon	Groot	Pokorny
10%	2,76	4,22	4,33	4,91	3,24	1,10	1,21	1,79
20%	2,28	3,89	4,09	4,55	2,64	1,37	1,57	2,03
30%	1,92	3,70	3,94	4,34	2,64	1,42	1,66	2,06
40%	2,16	3,58	3,84	4,20	2,64	1,30	1,56	1,92
50%	1,92	3,49	3,76	4,10	2,40	1,21	1,48	1,82
60%	2,28	3,41	3,69	4,01	2,64	1,01	1,29	1,61
70%	2,16	3,35	3,64	3,94	2,64	1,07	1,36	1,66
80%	2,40	3,30	3,59	3,88	2,40	0,90	1,19	1,48
90%	2,16	3,26	3,55	3,83	2,28	0,98	1,27	1,55
100%	1,80	3,22	3,51	3,78	2,28	1,06	1,35	1,62

Tabela 6.7: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 450 \sim 600 \text{ nm}$							
	Mín	Diâmetro (mm)			Máx	Desvio		
		Moon	Groot	Pokorny		Moon	Groot	Pokorny
10%	2,52	3,73	3,96	4,37	3,12	0,73	0,96	1,37
20%	2,28	3,44	3,72	4,04	2,40	1,04	1,32	1,64
30%	2,04	3,28	3,57	3,85	2,52	1,00	1,29	1,57
40%	2,28	3,18	3,47	3,73	2,40	0,78	1,07	1,33
50%	2,04	3,10	3,39	3,63	2,40	0,82	1,11	1,35
60%	2,16	3,04	3,32	3,56	2,52	0,76	1,04	1,28
70%	2,16	2,99	3,27	3,50	2,52	0,59	0,87	1,10
80%	2,04	2,95	3,22	3,45	2,40	0,67	0,94	1,17
90%	2,04	2,92	3,18	3,40	2,40	0,76	1,02	1,24
100%	2,28	2,89	3,14	3,36	2,64	0,49	0,74	0,96

Tabela 6.8: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 400 \sim 600 \text{ nm}$							
	Mín	Diâmetro (mm)			Máx	Desvio		
		Moon	Groot	Pokorny		Moon	Groot	Pokorny
10%	-	4,26	4,36	4,95	-	4,26	4,36	4,95
20%	2,16	3,93	4,12	4,59	2,52	1,65	1,84	2,31
30%	2,16	3,74	3,97	4,39	2,64	1,58	1,81	2,23
40%	1,92	3,62	3,87	4,24	2,52	1,46	1,71	2,08
50%	1,92	3,52	3,79	4,14	2,28	1,36	1,63	1,98
60%	2,04	3,45	3,73	4,05	2,52	1,29	1,57	1,89
70%	1,92	3,39	3,67	3,98	2,40	1,35	1,63	1,94
80%	2,04	3,34	3,62	3,92	2,64	1,18	1,46	1,76
90%	1,92	3,29	3,58	3,86	2,40	1,25	1,54	1,82
100%	1,92	3,25	3,54	3,82	2,40	1,21	1,50	1,78

Tabela 6.9: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 400 \sim 800 \text{ nm}$					
	$V_d(\text{mm/s})$		$V_c(\text{mm/s})$		$\tau(\text{ms})$	
	Média	Esperado	Média	Esperado	Média	Esperado
10%	0,66	0,42	1,16	0,89	333,33	423,82
20%	0,67	0,62	1,26	1,43	277,78	387,33
30%	0,94	0,73	1,15	1,74	388,89	368,28
40%	0,97	0,80	1,14	1,95	388,89	355,80
50%	1,02	0,86	1,54	2,11	333,33	346,71
60%	1,02	0,90	1,39	2,24	333,33	339,66
70%	0,89	0,94	1,40	2,35	166,67	333,97
80%	1,15	0,97	1,40	2,44	333,33	329,24
90%	1,15	1,00	0,96	2,52	388,89	325,22
100%	1,32	1,02	1,04	2,59	388,89	321,75

Tabela 6.10: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 550 \sim 650 \text{ nm}$					
	$V_d(\text{mm/s})$		$V_c(\text{mm/s})$		$\tau(\text{ms})$	
	Média	Esperado	Média	Esperado	Média	Esperado
10%	0,25	0,22	2,01	0,31	166,67	465,27
20%	0,34	0,42	0,90	0,89	277,78	423,82
30%	0,61	0,54	1,41	1,21	277,78	401,87
40%	0,80	0,62	0,82	1,43	222,22	387,33
50%	0,95	0,68	0,86	1,60	222,22	376,64
60%	1,16	0,73	1,28	1,74	222,22	368,28
70%	1,37	0,77	0,94	1,85	166,67	361,49
80%	0,90	0,80	3,02	1,95	222,22	355,80
90%	1,22	0,83	1,78	2,04	166,67	350,94
100%	1,29	0,86	1,31	2,11	166,67	346,71

Tabela 6.11: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 450 \sim 600 \text{ nm}$					
	$V_d(\text{mm/s})$		$V_c(\text{mm/s})$		$\tau(\text{ms})$	
	Média	Esperado	Média	Esperado	Média	Esperado
10%	0,72	0,52	0,85	1,16	166,67	404,86
20%	0,87	0,71	3,12	1,70	111,11	370,86
30%	0,95	0,81	1,92	1,99	166,67	353,28
40%	1,23	0,89	2,36	2,20	111,11	341,84
50%	1,22	0,94	2,16	2,35	277,78	333,55
60%	1,47	0,98	3,60	2,48	222,22	327,16
70%	1,27	1,02	2,88	2,58	222,22	322,02
80%	1,41	1,05	1,98	2,67	222,22	317,77
90%	1,06	1,08	2,31	2,75	111,11	314,18
100%	1,37	1,10	1,44	2,81	222,22	311,08

Tabela 6.12: *Comparação dos resultados do sistema com os modelos de simulação pupilar.*

I(%)	Comprimento de onda $\lambda = 400 \sim 600 \text{ nm}$					
	$V_d(\text{mm/s})$		$V_c(\text{mm/s})$		$\tau(\text{ms})$	
	Média	Esperado	Média	Esperado	Média	Esperado
10%	0,52	0,19	-	0,23	-	470,83
20%	0,46	0,40	1,08	0,81	388,89	428,75
30%	0,71	0,51	0,75	1,14	277,78	406,43
40%	0,44	0,59	1,30	1,37	333,33	391,63
50%	0,77	0,65	1,37	1,54	222,22	380,74
60%	0,78	0,70	1,57	1,67	166,67	372,23
70%	0,98	0,74	1,23	1,79	222,22	365,29
80%	0,73	0,78	1,62	1,89	222,22	359,49
90%	0,96	0,81	1,94	1,97	333,33	354,52
100%	0,95	0,83	1,39	2,05	333,33	350,19

6.4 Tratamento Automático de Oclusões

Durante a aquisição dos vídeos, cada voluntário foi posicionado de forma que todo o *frame* fosse ocupado pela imagem de um de seus olhos. No entanto, em alguns casos o objeto rastreado ficou recoberto pela pálpebra, conforme mostra a sequência da Figura 6.6. Nesses casos, em média, as informações de quatro *frames* referentes à posição e área do objeto perderam-se e, consequentemente, causaram alterações nos valores dos tempos de latência e nas velocidades.



(a) *Objeto rastreado.*

(b) *Obstrução parcial.*

(c) *Obstrução total.*



(d) *Obstrução total.*

(e) *Obstrução parcial.*

(f) *Objeto rastreado.*

Figura 6.6: *Tratamento automático de oclusões.*

Porém, mesmo que o objeto não seja encontrado nessas cenas os *frames* em questão devem ser considerados para compor a base de tempo e não distorcer os valores de latência e de velocidade. Fica claro aqui que a oclusão nesse processo não pode ser considerada

como ruído ou mesmo desconsiderada.

Nesse ponto a transformada circular de Hough mostrou sua robustez. Por exemplo, na sequência da Figura 6.6 os *frames* de 6.6(b) a 6.6(e) não foram detectados pelo algoritmo, pois o mesmo foi ajustado para retornar apenas as matrizes dos círculos cujos perímetros fossem maior do que um determinado limiar eliminando assim, aqueles parcialmente ou totalmente ocluídos.

Como estratégia, os *frames* que não atenderam as especificações do limiar retornaram como posição e área do objeto valores intermediários dos últimos detectados, como por exemplo, os mostrados nas Figuras 6.6(a) e 6.6(f).

6.5 Conclusões

Os resultados obtidos mostram a validade dos algoritmos desenvolvidos e implementados neste trabalho. O próximo capítulo apresenta as conclusões da dissertação, mencionando suas contribuições acadêmicas e sociais, levando em conta sua característica interdisciplinar, e apresenta sugestões de trabalhos futuros que podem ser desenvolvidos a partir desta dissertação.

CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

7.1 Introdução

Esta dissertação apresenta todas as etapas de processamento de um sistema pupilométrico. De forma simples e eficiente, construiu-se um equipamento robusto capaz de processar sequências de vídeo e extrair as principais medidas propostas por pesquisas na área da pupilometria, como por exemplo, os parâmetros de Fotiou [7].

O controle da fonte de iluminação está integrado ao sistema e é feito por meio da modulação por comprimento de pulso (PWM) e da variação de seu comprimento de onda (λ). Dessa forma, todo o movimento pupilar, acomodação, PLR e hippus puderam ser sincronizados com a fonte em qualquer ciclo pupilar.

A transformada circular de Hough utilizada como método de extração de características de alto nível para o rastreamento de objetos em imagens digitais provou ser eficiente, mesmo tratando-se de uma aplicação em tempo de vídeo, onde o tempo de processamento é limitado e crítico. Foram alcançadas as propriedades de invariância, detectando o objeto com posições e tamanhos variáveis e, além disso, de forma automática os ruídos, as imagens redundantes e as oclusões foram tratados, melhorando a desempenho do sistema sem, no entanto aumentar o custo computacional.

Todas as variáveis medidas apresentaram o comportamento esperado quando compara-

das com seus respectivos modelos. As amplitudes, os tempos de latência e as velocidades adquiridas pelo sistema mostraram um comportamento onde as maiores amplitudes, maiores velocidades e menores latências ocorreram para os estímulos de intensidades intermediárias.

Com um grupo de oito voluntários o tempo de latência antes do disparo do degrau apresentou um desvio mínimo de 4,09 ou 1,24%, para a fonte de luz branca a 80% de sua intensidade máxima, quando comparado com o modelo de Ellis [40]. A máxima constrição foi alcançada pela fonte de menor frequência e atingiu 17% da área da íris.

O modelo de Moon e Spencer [37], para o diâmetro da pupila, foi o que mais se aproximou dos resultados obtidos, com desvio mínimo de 0,57 mm para 70% de intensidade da fonte verde e máximo de 1,65 mm para 20% de intensidade da fonte de luz azul.

Ficou comprovado a relação das velocidades de contração e dilatação com a variação da amplitude e consequentemente com o comprimento de onda. As máximas velocidades de contração e dilatação foram atingidas para a 60% da intensidade da fonte de luz verde, 3,6 mm/s e 1,474 mm/s , respectivamente.

7.2 Principais Contribuições

A principal contribuição deste trabalho é a avaliação da transformada circular de Hough em sistemas de rastreamento em tempo real. Seu baixo custo computacional comparado com outras transformadas, como por exemplo, Fourier e Wavelet, permite que cada *frame* da sequência de vídeo seja analisado sem interrupções ou travamentos do sistema, com alta imunidade à ruídos, interferências e oclusões.

Outra contribuição deste trabalho é o desenvolvimento de um protótipo simples e robusto que permite o rastreamento da pupila em tempo de vídeo. A integração do controle de iluminação com o sistema de rastreamento em tempo de vídeo fornece suporte a um dispositivo completo, portátil e reprogramável.

Finalmente, o desenvolvimento de um *know how* local de uma área da ciência relacio-

nada ao bem estar humano. O comportamento da pupila ligado ao sistema nervoso simpático e parassimpático é considerado uma janela para o estudo de várias doenças conforme comprovam estudos recentes, como por exemplo, os trabalhos de Fotiou que relacionam a pupilometria com doenças cardíacas [17], Alzheimer [44] e Parkinson [45].

7.3 Publicações

Publicados

- C. R. Bernadelli, A. C. P. Veiga, E. L. Flôres, M. B. Pereira. Iris Motion Tracking - Using Feature Extraction by Color and Shape Matching. XIII Symposium on Virtual and Augmented Reality (SVR-2011, CD-ROM). Maio de 2011. Uberlândia, MG – Brasil.

Aprovados

- C. R. Bernadelli, A. C. P. Veiga, E. L. Flôres, M. B. Pereira. Iris Motion Tracking - Using Feature Extraction by Color and Shape Matching. 18th International Conference on Systems, Signal and Image Processing (IWSSIP-2011). Junho de 2011. Sarajevo, Bosnia and Herzegovina.

7.4 Trabalhos futuros

O sistema desenvolvido neste trabalho atende a todas as expectativas iniciais, porém, ele oferece condições para que outros aspectos sejam explorados, como por exemplo:

- Podem ser utilizadas outras técnicas em cada uma das etapas de processamento. Técnicas mais precisas de segmentação ou ainda mais velozes;

- Um sistema de aquisição que permita maiores resoluções temporais e espaciais dos vídeos adquiridos;
- Embarcar todo o sistema de modo a oferecer maior conforto e praticidade tanto para o operador quanto para o paciente;
- Avaliar o sistema desenvolvido utilizando um banco de vídeos com um número maior de amostras;
- Implementar atalhos via software que auxiliem o operador na execução de diversos ciclos pupilares; e
- Investigar e comparar as saídas do sistema desenvolvido com grupos de pessoas saudáveis e não saudáveis, acometidas por alguma doença que afete o sistema nervoso ou visual.

Referências Bibliográficas

- [1] P. L. Kaufman and A. Alm. *Adler's Physiology of the Eye*. Elsevier, 2011.
- [2] P. Schor. *Óptica Fisiológica e Cirurgia Refrativa*. PhD thesis, Universidade Federal de São Paulo – Escola Paulista de Medicina, 2003.
- [3] P. Brodal. *The Central Nervous System: Structure and Function*. Oxford University Press, 2010.
- [4] J. Kim, K. Park, and G. Khang. A method for size estimation of amorphous pupil in 3-dimensional geometry. 1:1451 – 1454, 2004.
- [5] A. C. M. Gomes, V. Tuoni, E. J. V. Martinelli, V. Vitiello, and L. G. Ribeiro. Comparaçāo entre o orbscan ii e o pupilômetro colvard na medida do diâmetro pupilar mesópico.
- [6] L. A. V. de Carvalho and A. P. Júnior. Resultados preliminares de um sistema computadorizado e estereoscópico para pupilometria in vivo. *Arquivos Brasileiros de Oftalmologia*, 71(6):775 – 800, 2008.
- [7] F. Fotiou, K. N. Fountoulakis, A. Goulas, L. Alexopoulos, and A. Palikaras. Automated standardized pupillometry with optical method for purposes of clinical practice and research. *Clinical Physiology*, 20(5):336 – 347, 2000.

- [8] G. N. S. M. Leal. Desenvolvimento de um pupilómetro. Master's thesis, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa (FCT-UNL), 2008.
- [9] S. Rakshit and D. M. Monro. Pupil shape description using fourier series. In *Signal Processing Applications for Public Security and Forensics, 2007. SAFE'07. IEEE Workshop on*, pages 1 – 4. IET, 2007.
- [10] V. F. Pamplona and M. M. Oliveira. Photorealistic models for pupil light reflex and iridal pattern deformation. Master's thesis, Universidade Federal do Rio Grande do Sul, 2008.
- [11] R. C. Gonzalez and R. E. Woods. *Processamento Digital de Imagens*. Pearson Prentice Hall, 2010.
- [12] P. Trevor-Roper. *The Eye and Its Disorders*. Oxford: Blackwells, 1974.
- [13] W. H. Bates. *The Cure of Imperfect Sight by Treatment Without Glasses*. Health Research Books, 1978.
- [14] H. J. Burd, S. J. Judge, and M. J. Flavell. Mechanics of accommodation of the human eye. *Vision research*, 39(9):1591 – 1595, 1999.
- [15] D. J. Coleman. Unified model for accommodative mechanism. *American journal of ophthalmology*, 69(6):1063, 1970.
- [16] N. Link and L. Stark. Latency of the pupillary response. *Biomedical Engineering, IEEE Transactions on*, 35(3):214–218, 1988.
- [17] A. Keivanidou, D. Fotiou, C. Arnaoutoglou, M. Arnaoutoglou, F. Fotiou, and A. Karlovositou. Evaluation of autonomic imbalance in patients with heart failure: A preliminary study of pupillomotor function. *Cardiol J*, 17(1):65 – 72, 2010.
- [18] Google Imagens. Representation of Parasympathetic Pathways in the Pupillary Light Reflex. Disponível em: http://www.tedmontgomery.com/the_eye/reflex.html. Acesso em 20 novembro de 2011.

- [19] S. A. Smith and R. R. Dewhirst. A simple diagnostic test for pupillary abnormality in diabetic autonomic neuropathy. *Diabetic medicine*, 3(1):38 – 41, 1986.
- [20] A. C. Bovik. *The Essential Guide to Video Processing*. Academic Press, 2009.
- [21] E. O. Nunes and A. Conci. Segmentação por textura e localização do contorno de regiões em imagens multibandas. *IEEE Latin America Transactions*, 5(3):185–192, 2007.
- [22] R. Uppaluri, E. A. Hoffman, M. Sonka, P. G. Hartley, G. W. Hunninghake, and G. McLennan. Computer recognition of regional lung disease patterns. *American Journal of Respiratory and Critical Care Medicine*, 160(2):648, 1999.
- [23] C. R. Bernadelli and A. C. P. Veiga. Iris motion tracking using feature extraction by shape matching. *XIII Symposium on Virtual and Augmented Reality - SVR-2011, CD-ROM*, May/2011.
- [24] F. Dadgostar and A. Sarrafzadeh. An adaptive real-time skin detector based on hue thresholding: A comparison on two motion tracking methods. *Pattern recognition letters*, 27(12):1342 – 1352, 2006.
- [25] M. S. Nixon and A. S. Aguado. *Feature Extraction and Image Processing*. Academic Press, 2008.
- [26] R. A. de Bem. Uma abordagem livre de modelo para rastreamento de objetos em sequências de imagens. Universidade de São Paulo, São Paulo, Brasil, 2007.
- [27] C. Y. V. W. da Silva. Extração de características de imagens médicas utilizando wavelets para mineração de imagens e auxílio ao diagnóstico. Universidade de São Paulo, São Paulo, Brasil, 2007.
- [28] S. Ordas, L. Boisrobert, M. Huguet, and A. F. Frangi. Active shape models with invariant optimal features (iof-asm) application to cardiac mri segmentation. In *Computers in Cardiology, 2003*, pages 633 – 636. IEEE, 2003.

- [29] M. J. Vasconcelos and J. M. R. da Silva Tavares. Modelos pontuais de distribuição em visão computacional. In *6º Congresso Nacional Mecânica Experimental*, 2005.
- [30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Training models of shape from sets of examples. In *Proc. British Machine Vision Conference*, volume 9, page 18. Citeseer, 1992.
- [31] A. B. V. Graciano. Rastreamento de objetos baseado em reconhecimento estrutural de padrões. Universidade de São Paulo, São Paulo, Brasil, Março 2007.
- [32] A. B. de Oliveira. Filtro de partículas adaptativo para o tratamento de oclusões no rastreamento de objetos em vídeos. Universidade Federal do Rio Grande do Sul, Porto Alegre, Brasil, Maio 2008.
- [33] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, 2008.
- [34] G. M. Simas, G. P. Fickel, L. Novelo, S. S. C. Botelho, and R. A. de Bem. Using computer vision for 3d probabilistic reconstruction and motion tracking. In *Computational Modeling (MCSUL), 2009 Third Southern Conference on*, pages 119–124. IEEE, 2009.
- [35] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university Press, 2003.
- [36] S. T. Acton and N. Ray. *Biomedical image analysis: Tracking*. Morgan & Claypool, 2005.
- [37] P. Moon and D. E. Spencer. On the stiles-crawford effect. *J. Opt. Soc. Am*, 34(6):319 – 329, 1944.
- [38] S. G. De Groot and J. W. Gebhard. Pupil size as determined by adapting luminance. *JOSA*, 42(7):492 – 495, 1952.
- [39] J. Pokorny and V. C. Smith. How much light reaches the retina? *Documenta Ophthalmologica Proceedings Series*, 59:491 – 512, 1997.

- [40] C. J. Ellis. The pupillary light reflex in normal subjects. *British Journal of Ophthalmology*, 65(11):754 – 759, 1981.
- [41] M.C. Moreira, R.N. do Prado, and A. Campos. Application of high-brightness leds in tissue human and their therapeutic interaction. In *Industry Applications Society Annual Meeting, 2009. IAS 2009. IEEE*, pages 1 – 6. IEEE.
- [42] G. L. Ferrari. Pupilometria dinâmica: Aplicação na detecção e avaliação da neuropatia autonômica diabética e estudo da correlação entre a resposta temporal da pupila ao estímulo visual e a glicemia. Master's thesis, Universidade Tecnológica Federal do Paraná, 2010.
- [43] M. Alpern, D. W. McCready, and L. Barr. The dependence of the photopupil response on flash duration and intensity. *The Journal of General Physiology*, 47(2):265, 1963.
- [44] D. F. Fotiou, C. G. Brozou, A. B. Haidich, D. Tsipitsios, M. Nakou, A. Kabitsi, C. Giantselidis, and F. Fotiou. Pupil reaction to light in alzheimer's disease: Evaluation of pupil size changes and mobility. *Aging clinical and experimental research*, 19(5):364 – 371, 2007.
- [45] V. Stergiou, D. Fotiou, D. Tsipitsios, B. Haidich, M. Nakou, C. Giantselidis, and A. Karlovasitou. Pupillometric findings in patients with parkinson's disease and cognitive disorder. *International Journal of Psychophysiology*, 72(2):97 – 101, 2009.



Aquisição e processamento

A.1 Sistema de sincronismo

A.1.1 Rotina interna do micro controlador

Código A.1: *Controle PWM da fonte de iluminação.*

```
1 #include <18F4550.h>
2
3 #device adc=8
4
5 #FUSES NOWDT      //No Watch Dog Timer
6 #FUSES WDT128     //Watch Dog Timer uses 1:128 Postscale
7 #FUSES HSPLL      //Crystal/Resonator with PLL enabled
8 #FUSES NOPROTECT //Code not protected from reading
9 #FUSES BROWNOUT   //Reset when brownout detected
10 #FUSES BORV20    //Brownout reset at 2.0V
11 #FUSES PUT       //Power Up Timer
12 #FUSES NOCPD    //No EE protection
13 #FUSES STVREN   //Stack full/underflow will cause reset
```

```

#FUSES NODEBUG      //No Debug mode for ICD

13 #FUSES NOLVP      //No low voltage prgming , B3(PIC16) or
                     //B5(PIC18) used for I/O

15 #FUSES NOWRT      //Program memory not write protected

#FUSES NOWRID      //Data EEPROM not write protected

17 #FUSES IESO       //Internal External Switch Over mode enabled

#FUSES FCMEN       //Fail-safe clock monitor enabled

19 #FUSES PBADEN     //PORTB pins are configured as analog
                     //input channels on RESET

21 #FUSES NOWRTC     //configuration not registers write protected

#FUSES NOWRIB      //Boot block not write protected

23 #FUSES NOEBTR     //Memory not protected from table reads

#FUSES NOEBTRB     //Boot block not protected from table reads

25 #FUSES NOCPB      //No Boot Block code protection

#FUSES MCLR        //Master Clear pin enabled

27 #FUSES LPT1OSC    //Timer1 configured for low-power operation

#FUSES NOXINST     //Extended set extension and Indexed Addressing
                     // mode disabled (Legacy mode)

29
#FUSES PLL5        //No PLL PreScaler

31 #FUSES CPUDIV1    //System Clock by 4

#FUSES NOUSBDIV    //USB clock source comes from primary oscillator

33 #FUSES VREGEN     //USB voltage regulator enabled

#FUSES ICPRT

35

#use delay(clock=48000000)

37 #use rs232(baud=9600,parity=N,xmit=PIN_C6,recv=PIN_C7,bits=8)

```

```
#include <usb_cdc.h>

39 #include <stdio.h>

void main()

41 {

    long int x,y,c,d;

43    float a,b;

    setup_adc_ports(NO_ANALOGS|VSS_VDD);

45    setup_adc(ADC_OFF);

    setup_psp(PSP_DISABLED);

47    setup_spi(SPLSS_DISABLED);

    setup_wdt(WDT_OFF);

49    setup_timer_0(RTCC_INTERNAL);

    setup_timer_1(T1_DISABLED);

51    setup_timer_2(T2_DIV_BY_4,255,1);

    setup_comparator(NC_NC_NC_NC);

53    setup_vref(FALSE);

    setup_ccp1(CCP_PWM);

55    setup_ccp2(CCP_PWM);

// Setup_Oscillator parameter not selected from Intr

57        // Oscillator Config tab

// TODO: USER CODE!!

59    usb_init();

    usb_cdc_init();

61    while (!usb_enumerated()) {}

    while (true)

63    {
```

```

usb_task();

65      if (usb_cdc_kbhit())
{
67          if (usb_cdc_getc() == 101)
{
69              x = usb_cdc_getc();
71              a = x*10.23 ;
73              b = y*10.23 ;
75              set_pwm1_duty(c);
77              set_pwm2_duty(d);
79      }
}
}

```

A.1.2 Rotina interna da interface C Sharp

Principais classes.

Código A.2: Interface de controle para aquisição do vídeo.

```

int i = 0;
2 public webcam()
{

```

```
4     InitializeComponent();  
5 }  
6 ~webcam()  
7 {  
8     this.Stop();  
9 }  
10 #region Start and Stop Capture Functions  
11 // Ajusta o tamanho da imagem da WebCam  
12 public void ImageSize()  
13 {  
14     m_Width = imgwebcam.Size.Width;  
15     m_Height = imgwebcam.Size.Height;  
16 }  
17 // Iniciar a captura de telas da Webcam  
18 public void Start()  
19 {  
20     try  
21     {  
22 // Ajusta o tamanho da imagem  
23     ImageSize();  
24 // Por segurança, chamamos o método stop só para  
25 // garantirmos que não estamos rodando o código.  
26     this.Stop();  
27 // Criamos a janela de captura usando a  
28 // API "capCreateCaptureWindowA"  
29 mCapHwnd = capCreateCaptureWindowA("WebCap", 0, 0, 0, m_Width, m_Height,
```

```
30     this.Handle.ToInt32(), 0);  
31     // Liberamos recurso ao sistema operacional.  
32     Application.DoEvents();  
33     //Enviamos a mensagem através do SO para conectar  
34     //com o driver da WebCam  
35     SendMessage(mCapHwnd, WM_CAP_CONNECT, 0, 0);  
36     // Ajustamos o intervalo de captura da webcam  
37     this.timercam.Interval = 1;  
38     bStopped = false;  
39     this.timercam.Start();  
40     }  
41     catch (Exception excep)  
42     {  
43         MessageBox.Show("Ocorreu um erro na exibição da WebCam.  
44         Verifique se está tudo conectado.\r  
45         \n\n" + excep.Message);  
46     this.Stop();  
47     }  
48     }  
49     public void Stop()  
50     {  
51         try  
52         {  
53             bStopped = true;  
54             this.timercam.Stop();  
55             // Liberamos recurso ao sistema operacional.
```

```
56 Application.DoEvents();  
57 //Envia mensagem ao SO para desconectar a Webcam.  
58 SendMessage(mCapHwnd, WM_CAP_DISCONNECT, 0, 0);  
59 //Fecha a área de transferência.  
60 CloseClipboard();  
61 }  
62 catch (Exception excep)  
63 { // Não dispara nenhum erro.  
64 }  
65 }  
66 #endregion  
67 #region Código de Captura de vídeo.  
68 /// Captura os frames  
69 public void timercam_Tick(object sender, System.EventArgs e)  
70 {  
71     try  
72     {  
73         i++;  
74         // Pausa o temporizador  
75         this.timercam.Stop();  
76         // Ajusta o tamanho da imagem.  
77         ImageSize();  
78         // Envia ao SO a mensagem para capturar o próximo frame.  
79         SendMessage(mCapHwnd, WM_CAP_GET_FRAME, 0, 0);  
80         // copia o frame capturado para a área de transferência.  
81         SendMessage(mCapHwnd, WM_CAP_COPY, 0, 0);
```

```
82 //Abre a área de transferência.  
OpenClipboard(mCapHwnd);  
84 //Busca os dados da área de transferência, colocando os dados em  
//uma estrutura de ponteiro.  
86 IntPtr img = GetClipboardData(2);  
//Fecha a área de transferência.  
88 CloseClipboard();  
//Criamos aqui um objeto do tipo Bitmap.  
90 System.Drawing.Bitmap bmp = new System.Drawing.Bitmap(m_Width, m_Height);  
//Criamos um objeto gráfico para manipular nossa imagem.  
92 System.Drawing.Graphics g = System.Drawing.Graphics.FromImage(bmp);  
//Ajustamos algumas propriedades do nosso objeto gráfico.  
94 //No caso abaixo, estou tentanto otimizar ao máximo a velocidade.  
//Mas também a possível ajustar para a qualidade da imagem.  
96 g.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighSpeed;  
g.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.Low;  
98 g.PixelOffsetMode = System.Drawing.Drawing2D.PixelOffsetMode.HighSpeed;  
g.CompositingQuality = System.Drawing.Drawing2D.CompositingQuality.HighSpeed;  
100 //Pegamos a imagem que está na estrutura do ponteiro que buscamos da  
//área de transferência e carregamos dentro do nosso bitmap.  
102 g.DrawImage(Image.FromHbitmap(img), 0, 0, m_Width, m_Height);  
//Exibimos o frame da Webcam no controle que adicionamos no formulário  
104 //o frame foi salvo na variável do tipo Bitmap  
//e será exibido no controle System.Windows.Forms.PictureBox abaixo.  
106 imgwebcam.Image = bmp;  
String bmpFile = @"d:\Imagens\Frame_" + i + "_Led_" +
```

```
108     Class2.c.get_string(1) + "_Infravermelho_" +  
109     Class2.c.get_string(2) + ".jpg";  
110 // salva o arquivo de imagem  
111     bmp.Save(bmpFile);  
112     //fazemos um refresh na imagem.  
113     imgwebcam.Refresh();  
114 // Liberamos recurso ao sistema operacional.  
115     Application.DoEvents();  
116 if (!bStopped)  
117     this.timercam.Start();  
118 }  
119 catch (Exception excep)  
120 {  
121     MessageBox.Show("Ocorreu um erro na exibição da WebCam. Verifique o  
122                     status da conexão.\r\n  
123                     n\n" + excep.Message);  
124 this.Stop(); // stop the process  
125 }  
126 }  
127 #endregion  
128 }  
129 }  
130 }
```

A.2 Processamento do *video stream*

Principais funções.

Código A.3: *Interface de controle para o rastreamento.*

```

function varargout = teste(varargin)
2 clc
gui_Singleton = 1;
4 gui_State = struct('gui_Name',          mfilename, ...
    'gui_Singleton',    gui_Singleton, ...
6    'gui_OpeningFcn', @teste_OpeningFcn, ...
    'gui_OutputFcn',   @teste_OutputFcn, ...
8    'gui_LayoutFcn',  [], ...
    'gui_Callback',    []);
10 if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
12 end
14 if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
16 else
    gui_mainfcn(gui_State, varargin{:});
end
18 function teste_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.output = hObject;
20 guidata(hObject, handles);
function varargout = teste_OutputFcn(hObject, eventdata, handles)

```

```
22 varargout{1} = handles.output;  
  
active.Enable='off';  
  
24 set(findobj('Tag','pushbutton1'),active);  
  
set(findobj('Tag','pushbutton2'),active);  
  
26 set(findobj('Tag','pushbutton4'),active);  
  
set(findobj('Tag','pushbutton5'),active);  
  
28 set(findobj('Tag','pushbutton6'),active);  
  
set(findobj('Tag','pushbutton7'),active);  
  
30 set(findobj('Tag','pushbutton8'),active);  
  
set(handles.uipanel2,'Visible','on');  
  
32 set(handles.uipanel3,'Visible','off');  
  
set(handles.uipanel4,'Visible','on');  
  
34 set(handles.uipanel6,'Visible','off');  
  
set(handles.text5,'String',12);  
  
36 set(handles.slider1,'Value',12);  
  
set(handles.text6,'String',12);  
  
38 set(handles.slider2,'Value',12);  
  
set(handles.text7,'String',0.4);  
  
40 set(handles.slider3,'Value',40);  
  
set(handles.text8,'String',8);  
  
42 set(handles.slider4,'Value',8);  
  
function pushbutton3_Callback(hObject, eventdata, handles)  
  
44 active.Enable='on';  
  
set(findobj('Tag','pushbutton4'),active);  
  
46 set(findobj('Tag','pushbutton5'),active);  
  
active.Enable='off';
```

```
48 set(findobj('Tag','pushbutton3'),active);

axes(handles.axes1)

50 vid=videoinput('winvideo',1,'RGB24_160x120');

vidRes = get(vid, 'VideoResolution');

52 nBands = get(vid, 'NumberOfBands');

hImage = image( zeros(vidRes(2), vidRes(1), nBands) );

54 preview(vid, hImage);

handles.video=vid;

56 guidata(hObject, handles)

function pushbutton4_Callback(hObject, eventdata, handles)

58 vid=handles.video;

closepreview(vid);

59 stop(vid);

% delete(vid);

60 active.Enable='on';

62 active.Enable='on';

set(findobj('Tag','pushbutton3'),active);

64 active.Enable='off';

set(findobj('Tag','pushbutton4'),active);

66 set(findobj('Tag','pushbutton5'),active);

function pushbutton5_Callback(hObject, eventdata, handles)

68 active.Enable='off';

set(findobj('Tag','pushbutton3'),active);

70 set(findobj('Tag','pushbutton4'),active);

set(findobj('Tag','pushbutton5'),active);

72 active.Enable='on';

set(findobj('Tag','pushbutton6'),active);
```

```
74 vid=handles.video;
%closepreview(vid);
76 for frames=1:10
    minR=fix(get(handles.slider1,'Value'));
78    maxR=fix(get(handles.slider2,'Value'));
    thresh = get(handles.slider3,'Value')/100;
80    delta = get(handles.slider4,'Value');
    if minR<0 || maxR<0 || minR>maxR || thresh<0 || thresh>1 || delta<0
82    errordlg('Condições de entrada: 0<minR, 0<maxR,
               minR<=maxR, 0<thresh<=1, 0<delta');
84    return;
end
86 %axes(handles.axes1)
% vidRes=get(vid,'VideoResolution');
88 % nBands=get(vid,'NumberOfBands');
% im=image(zeros(vidRes(1),vidRes(2),nBands));
90 im=getsnapshot(vid);
%
92 % if teste1==1
%     im=getsnapshot(vid);
94 % elseif teste2==1
%     im=frame2im(getframe(handles.tela1));
96 % end
origim = im;
98 if length(size(im))>2
    im = rgb2gray(im);
```

```

100 end

101 fid=fopen( 'dados.dat' , 'a' );

102 maxR2 = 2*maxR;

103 hough = zeros( size(im,1)+maxR2, size(im,2)+maxR2, maxR-minR+1);

104 [X Y] = meshgrid( 0:maxR2, 0:maxR2 );

105 Rmap = round( sqrt( (X-maxR).^2 + (Y-maxR).^2 ) );

106 Rmap(Rmap<minR | Rmap>maxR) = 0;

107 edgeim = edge(im, 'canny', [0.15 0.2]);

108 [Ey Ex] = find( edgeim );

109 [Cy Cx R] = find( Rmap );

110 for i = 1:length(Ex);

111     Index = sub2ind( size(hough), Cy+Ey(i)-1, Cx+Ex(i)-1, R-minR+1);

112     hough(Index) = hough(Index)+1;

113 end

114 twoPi = 0.9*2*pi;

115 circles = zeros(0,4);

116 for radius = minR:maxR

117     slice = hough(:,:,radius-minR+1);

118     twoPiR = twoPi*radius;

119     slice(slice<twoPiR*thresh) = 0;

120     [y x count] = find( slice );

121     circles = [ circles; [x-maxR, y-maxR, radius*ones(1,3), count/twoPiR] ];

122
123
124 end

125 circles = sortrows(circles, -4);

```

```
126 i = 1;
127 while i<size(circles,1)
128     j = i+1;
129     while j<=size(circles,1)
130         if sum(abs(circles(i,1:3)-circles(j,1:3))) <= delta
131             circles(j,:)= [];
132         else
133             j = j+1;
134         end
135     end
136     i = i+1;
137 end
138 if nargout==0
139     % image([0 120],[0 160],origim), hold on;
140     imshow(origim), hold on;
141     for i = 1:size(circles,1)
142         x = circles(i,1)-circles(i,3);
143         y = circles(i,2)-circles(i,3);
144         w = 2*circles(i,3);
145         rectangle('Position', [x y w w], 'EdgeColor', 'r',
146                     'Curvature', [1 1], 'LineWidth', 2);
147     end
148 end
149 hold off;
150 fprintf(fid,'%i\t',circles);%%
151 fclose(fid);%%
```

```
152 end
153
154 active.Enable='off';
155 set(findobj('Tag','pushbutton6'),active);
156 active.Enable='on';
157 set(findobj('Tag','pushbutton3'),active);
158 set(findobj('Tag','pushbutton5'),active);
159
160 function pushbutton6_Callback(hObject, eventdata, handles)
161 function pushbutton7_Callback(hObject, eventdata, handles)
162 function pushbutton8_Callback(hObject, eventdata, handles)
163 function uipanel1_SelectionFcn(hObject, eventdata, handles)
164 if hObject==handles.op1
165     set(handles.uipanel2,'Visible','on');
166     set(handles.uipanel3,'Visible','off');
167     set(handles.uipanel4,'Visible','on');
168     set(handles.uipanel6,'Visible','off');
169 %     vid=videoinput('winvideo',1,'RGB24_160x120');
170 %     handles.video=vid;
171 %     guidata(hObject, handles)
172 elseif hObject==handles.op2
173     set(handles.uipanel2,'Visible','off');
174     set(handles.uipanel3,'Visible','on');
175     set(handles.uipanel4,'Visible','off');
176     set(handles.uipanel6,'Visible','on');
177
178 end
179
180 function slider1_Callback(hObject, eventdata, handles)
```

```
178 minR=get(handles.slider1,'Value');

set(handles.text5,'String',fix(minR));

180 active.Enable='on';

guidata(hObject, handles)

182 function slider1_CreateFcn(hObject, eventdata, handles)

if isequal(get(hObject,'BackgroundColor'),

184 get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor',[.9 .9 .9]);

186 end

function slider2_Callback(hObject, eventdata, handles)

188 maxR=get(handles.slider2,'Value');

set(handles.text6,'String',fix(maxR));

190 guidata(hObject, handles)

function slider2_CreateFcn(hObject, eventdata, handles)

192 if isequal(get(hObject,'BackgroundColor'),

194 get(0,'defaultUicontrolBackgroundColor'))

    set(hObject,'BackgroundColor',[.9 .9 .9]);

196 end

function slider3_Callback(hObject, eventdata, handles)

198 thresh=get(handles.slider3,'Value');

200 thresh=thresh/100;

set(handles.text7,'String',thresh);

202 guidata(hObject, handles)

function slider3_CreateFcn(hObject, eventdata, handles)

204 if isequal(get(hObject,'BackgroundColor'),

206 get(0,'defaultUicontrolBackgroundColor'))
```

```
204      set(hObject, 'BackgroundColor', [.9 .9 .9]);  
end  
  
206 function slider4_Callback(hObject, eventdata, handles)  
delta=get(handles.slider4, 'Value');  
  
208 set(handles.text8, 'String', delta);  
  
guidata(hObject, handles)  
  
210 function slider4_CreateFcn(hObject, eventdata, handles)  
if isequal(get(hObject, 'BackgroundColor'),  
212     get(0, 'defaultUicontrolBackgroundColor'))  
set(hObject, 'BackgroundColor', [.9 .9 .9]);  
  
214 end
```



Descrição dos equipamentos utilizados

B.1 Webcam

Requisitos do Sistema

- Requisitos Gerais:
 - Windows® XP (SP2 ou superior), Windows Vista® ou Windows® 7 (32-bit ou 64-bit)
- Requisitos Básicos:
 - 1 GHz
 - 512 MB RAM ou mais
 - 200 MB de espaço em disco rígido
 - Conexão com Internet
 - Porta USB 1.1 (2.0 recomendado)
- Para vídeo chamadas HD 720p e gravação de vídeo HD 720p:
 - 2.4 GHz Intel® Core™2 Duo

- 2 GB RAM
- 200 MB de espaço em disco rígido
- Porta USB 2.0
- 1 Mbps de velocidade de upload ou mais
- 1280 x 720 resolução de tela

Número de Referência

- PN 960-000715

Especificações Técnicas

- HD vídeo chamada (1280 x 720 pixels) com sistema recomendado*
- HD vídeo captura: Até 1280 x 720 pixels*
- Tecnologia Logitech Fluid Crystal™
- Tecnologia Logitech More HD
- Autofoco
- Fotos: até 8 megapixels (software avançado)
- Microfone embutido com tecnologia RightSound™
- Hi-Speed USB 2.0 certified (recomendado)
- Encaixe universal para laptops, LCD ou monitores CRT

Software Logitech webcam :

- Logitech VidTM HD
- Logitech RightLightTM 2 technology
- Captura de vídeo e foto
- MAGIXTM software editor de vídeo e foto*
- 1-click Facebook®, TwitterTM and YouTubeTM HD upload (Registro obrigatório)
- Logitech Video EffectsTM: filtros divertidos, avatares, acessórios faciais, máscaras de vídeo e confecção de máscara**



Figura B.1: *Webcam Logitech.*

B.2 LEDs

Foram utilizados LEDs nas cores azul, verde, vermelho e branco, todos com especificações similares e do mesmo fabricante. A seguir, encontra-se o *data sheet* do LED de cor branca utilizado, onde também é possível encontrar o endereço eletrônico do fabricante, caso seja necessário outras especificações.



Cree® 5-mm Round LED Model # LC503FWH1-15P-A0 Data Sheet

15-degree, 5-mm round LED lamp in white color with water-transparent lens and stopper

Applications

- Flashlight
- LCD Backlight
- Illuminations

Absolute Maximum Ratings ($T_A = 25^\circ\text{C}$)

Items	Symbol	Absolute Maximum Rating	Unit
Forward Current	I_F	25	mA
Peak Forward Current <small>Note</small>	I_{FP}	100	mA
Reverse Voltage	V_R	5	V
Power Dissipation	P_D	100	mW
Operation Temperature	T_{opr}	-40 ~ +95	°C
Storage Temperature	T_{stg}	-40 ~ +100	°C
Lead Soldering Temperature	T_{sol}	Max. 260°C for 3 sec. max. (3 mm from the base of the epoxy bulb)	

Note: Pulse width ≤ 0.1 msec, duty $\leq 1/10$.

Typical Electrical & Optical Characteristics ($T_A = 25^\circ\text{C}$)

Characteristics	Symbol	Condition	Unit	Minimum	Typical	Maximum
Forward Voltage	V_F	$I_F = 20$ mA	V		3.2	4.0
Reverse Current	I_R	$V_R = 5$ V	μA			100
Luminous Intensity	I_V	$I_F = 20$ mA	mcd	16800	24000	
Chromaticity Coordinates	x	$I_F = 20$ mA			0.2877	
	y	$I_F = 20$ mA			0.2831	
50% Power Angle	$20\frac{1}{2}\text{H-H}$	$I_F = 20$ mA	deg		15	

Subject to change without notice.
www.cree.com/ledlamps



Rank		B12				B13				B14			
Chromaticity	x	0.2895	0.2998	0.3045	0.2960	0.2950	0.3070	0.3100	0.300	0.3000	0.3100	0.3130	0.3050
Coordinates	y	0.2905	0.3028	0.2865	0.2760	0.3210	0.3370	0.3150	0.3030	0.3030	0.3150	0.2970	0.2870

Rank		B21				B22				B23			
Chromaticity	x	0.3070	0.3190	0.3200	0.3100	0.3100	0.3200	0.3220	0.3130	0.3190	0.3300	0.3300	0.3200
Coordinates	y	0.3370	0.3490	0.3270	0.3150	0.3150	0.3270	0.3080	0.2970	0.3490	0.3600	0.3390	0.3270

Rank		B24			
Chromaticity	x	0.3200	0.3300	0.3300	0.3220
Coordinates	y	0.3270	0.3390	0.3180	0.3080

Forward Voltage (VF)

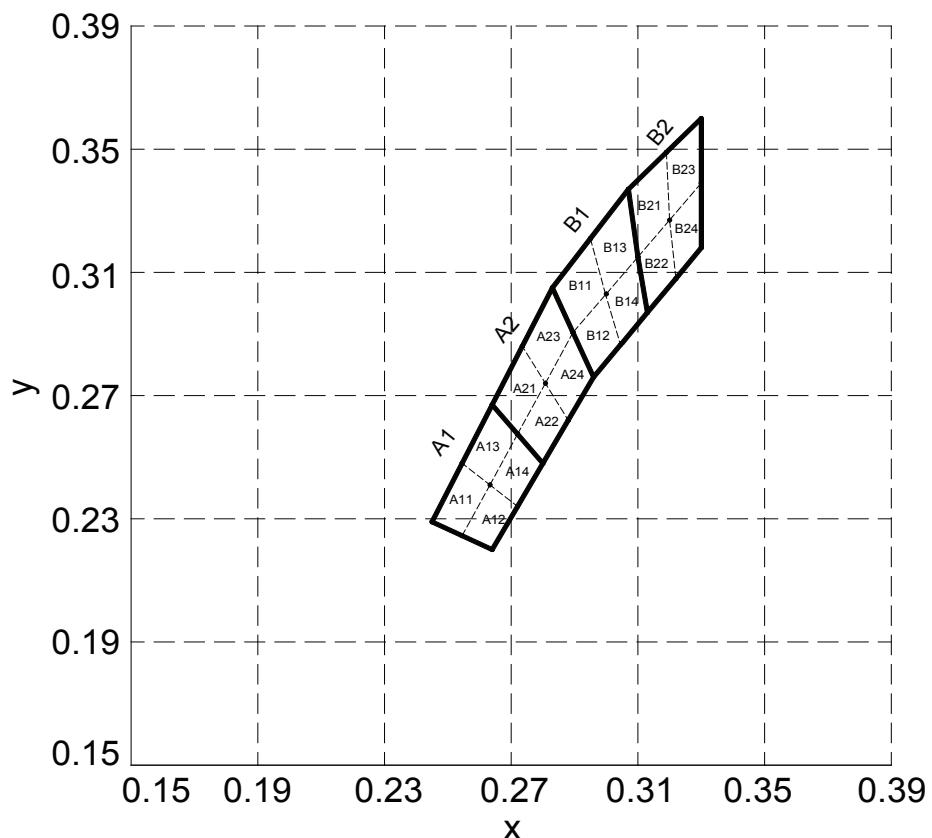
Rank	V7	V8	V9	V10	V11	V12
Voltage	2.8-3.0V	3.0-3.2V	3.2-3.4V	3.4-3.6V	3.6-3.8V	3.8-4.0V

Important Notes:

1. All ranks will be included per delivery; rank ratio will be based on the dice distribution.
2. Pb content <1000 ppm.
3. Tolerance of measurement of luminous intensity is $\pm 15\%$.
4. Tolerance of measurement of the color coordinates is ± 0.01 .
5. Tolerance of measurement of V_F is ± 0.05 V.
6. Packaging methods are available for selection; please refer to the "Cree LED Lamp Packaging Standard" document.
7. Please refer to the "Cree LED Lamp Reliability Test Standards" document for reliability test conditions.
8. Please refer to the "Cree LED Lamp Soldering & Handling" document for information about how to use this LED product safely.



CIE Chromaticity Diagram





Graphs

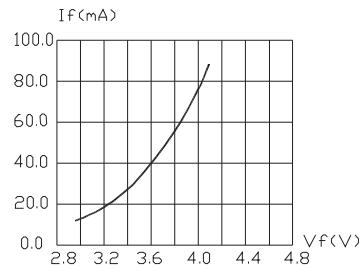


FIG.1 FORWARD CURRENT VS. FORWARD VOLTAGE

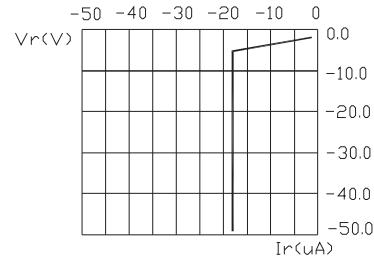


FIG.2 REVERSE CURRENT VS. REVERSE VOLTAGE

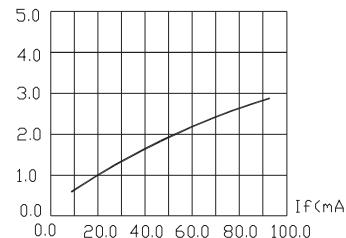


FIG.3 RELATIVE LUMINOUS INTENSITY VS. FORWARD CURRENT

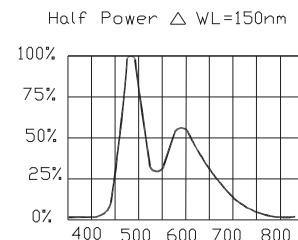


FIG.4 RELATIVE LUMINOUS INTENSITY VS. WAVELENGTH

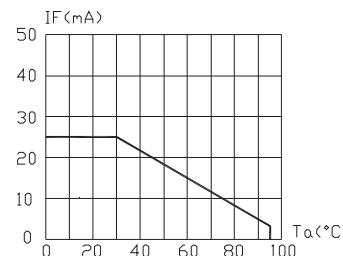
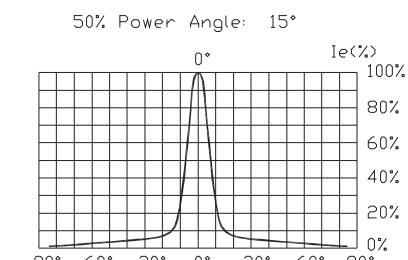
FIG.5 MAXIMUM FORWARD CURRENT VS. AMBIENT TEMPERATURE (T_{jmax}=105°C)

FIG.6 FAR FIELD PATTERN

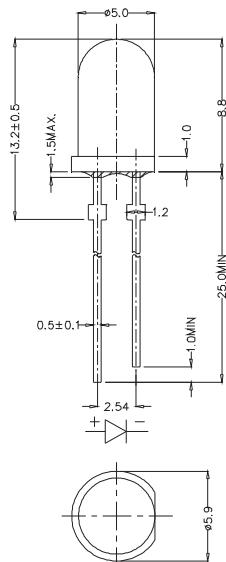


Mechanical Dimensions

All dimensions are in mm. Tolerance is ± 0.25 mm unless otherwise noted.

An epoxy meniscus may extend about 1.5 mm down the leads.

Burr around bottom of epoxy may be 0.5 mm max.



Notes

RoHS Compliance

The levels of environmentally sensitive, persistent biologically toxic (PBT), persistent organic pollutants (POP), or otherwise restricted materials in this product are below the maximum concentration values (also referred to as the threshold limits) permitted for such substances, or are used in an exempted application, in accordance with EU Directive 2002/95/EC on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS), as amended through April 21, 2006.

Vision Advisory Claim

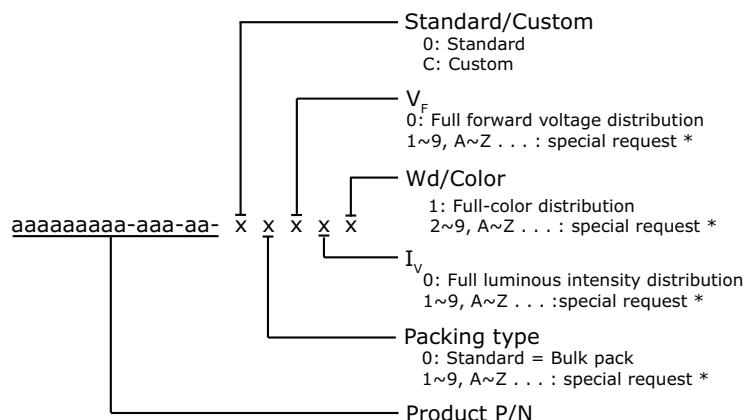
Users should be cautioned not to stare at the light of this LED product. The bright light can damage the eye.



Kit Number System

Cree LED lamps are tested and sorted into performance bins. A bin is specified by ranges of color, forward voltage, and brightness. Sorted LEDs are packaged for shipping in various convenient options. Please refer to the "Cree LED Lamp Packaging Standard" document for more information about shipping and packaging options.

Cree LEDs are sold by order codes in combinations of bins called kits. Order codes are configured in the following manner:



* Contact your Cree sales representative for ordering information.

Standard Available Kits*

Kit Number	Description
LC503FWH1-15P-A0-00001	5mm LED Lamp, FULL RANK. Bulk Pack

* Please contact your Cree representative about the availability of non-standard kits.

