

**Instituto Tecnológico de Aeronáutica – ITA**

**Divisão de Ciência da Computação – IEC**

**Programa de Pós-Graduação em Engenharia Eletrônica e Computação**

**Área de Informática – PG/EEC-I**

**Mecanismos de Interação Ocular Baseados  
em Imagens Voltados à Inclusão Digital de**

**Portadores de Necessidades Especiais**

Dissertação de Mestrado – Versão Final

por

**Fabricio da Silva Soares**

**Prof. Dr. Carlos Henrique Quartucci Forster**

Orientador

Campo Montenegro  
São José dos Campos, SP – Brasil  
2008

Tese apresentada à Pró-Reitoria de Pós-Graduação e Pesquisa do Instituto Tecnológico de Aeronáutica, como parte dos requisitos para obtenção do título de Mestre em Ciências no Curso de Engenharia Eletrônica e Computação, área de Informática.

**Fabricio da Silva Soares**

# **Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais**

Tese aprovada em sua versão final pelos abaixo assinados:

**Prof. Dr. Carlos Henrique Quartucci Forster**  
Orientador

**Prof. Dr. Celso Massaki Hirata**  
Pró-Reitor de Pós-Graduação e Pesquisa

Campo Montenegro  
São José dos Campos, SP – Brasil  
2008

## Dados Internacionais de Catalogação-na-Publicação (CIP)

### Divisão de Informação e Documentação

Soares, Fabricio da Silva

Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais / Fabricio da Silva Soares.

São José dos Campos, 2008.

140f.

Tese de Mestrado – Curso Engenharia Eletrônica e Computação. Área de Informática – Instituto Tecnológico de Aeronáutica, 2008. Orientador: Carlos Henrique Quartucci Forster.

1. Rastreamento Ocular. 2. Dispositivos Rastreadores. 3. Processamento Digital de Imagens. 4. Transformada de Hough. 5. Portadores de Necessidades Especiais. I. Comando-Geral de Tecnologia Aeroespacial. Instituto Tecnológico de Aeronáutica. Divisão de Ciência da Computação. II. Título.

## REFERÊNCIA BIBLIOGRÁFICA

SOARES, Fabricio da Silva. **Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais**. 2008. 140f. Tese de Mestrado – Instituto Tecnológico de Aeronáutica, São José dos Campos.

## CESSÃO DE DIREITOS

NOME DO AUTOR: Fabricio da Silva Soares

TÍTULO DO TRABALHO: Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais.

TIPO DO TRABALHO/ANO: Tese / 2008

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias desta tese e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta tese pode ser reproduzida sem a autorização do autor.

---

Fabricio da Silva Soares

Avenida Apolônio Sales, 500 – Centro

CEP: 48.608-100 – Paulo Afonso – BA

# **Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais**

**Fabricio da Silva Soares**

Composição da Banca Examinadora:

Prof. Dr. Sérgio Roberto Matiello Pellegrino	Presidente	–	ITA
Prof. Dr. Carlos Henrique Quartucci Forster	Orientador	–	ITA
Prof. Dr. Felipe Afonso de Almeida	Membro	–	ITA
Prof. Dra. Leila Maria Garcia Fonseca	Membro Externo	–	INPE

Escrever aqui uma pequena dedicatória!

De duas linhas no máximo

---

# ***Agradecimentos***

---

Escrever uma página de agradecimentos para todos que participaram diretamente e indiretamente desse trabalho.

*“Existe uma lenda acerca de um pássaro que só canta uma vez na vida, com mais suavidade que qualquer outra criatura sobre a terra. A partir do momento em que deixa o ninho, começa a procurar um espinheiro-alvar e só descansa quando o encontra. Depois, cantando entre os galhos selvagens, empala-se no acúleo mais agudo e mais comprido. E, morrendo, sublima a própria agonia e despede um canto mais belo que o da cotovia e o do rouxinol. Um canto superlativo, cujo preço é a existência. Mas o mundo inteiro pára para ouvi-lo, e Deus sorri no céu. Pois o melhor só se adquire à custa de um grande sofrimento...  
Pelo menos é o que diz a lenda”*

– Colleen McCullough

---

# Resumo

---

Essa dissertação apresenta o desenvolvimento de um protótipo de Dispositivo Rastreador (*Eye Tracker*) que permite a interação entre o usuário e o computador através dos movimentos oculares. Através de técnicas de Processamento Digital de Imagens procuramos detectar o Ponto de Interesse (*Point of Regard*) do usuário na tela do computador. O nosso protótipo foi desenvolvido com base nas técnicas de Foto-Oculografia e Vídeo-Oculografia para detectar a íris do usuário em imagens e vídeos capturados em tempo real. O principal problema na utilização da íris para detectar o Ponto de Interesse do usuário, é obter um bom grau de precisão vertical mesmo nas imagens capturadas com a íris parcialmente coberta pelas pálpebras. A solução adotada foi a utilização dos algoritmos de detecção de círculos baseados na Transformada de Hough, onde além de detectar a íris com bom grau de precisão, foi possível obter uma taxa de processamento de 93 milissegundos por imagem analisada. O *hardware* do Dispositivo Rastreador foi montado artesanalmente, fixando o Dispositivo de Captura de Vídeo no visor direito de um óculos de proteção industrial. Em conjunto com o *hardware*, foram desenvolvidos *softwares* para permitir que Portadores de Necessidades Especiais emulem o uso dos dispositivos de entrada padrão do computador (mouse e teclado) através dos movimentos oculares. Os principais sistemas criados foram: o “Mouse Óptico Ocular” e o “Teclado Óptico Ocular”. O Mouse Óptico Ocular posiciona o cursor do mouse na região observada pelo usuário na tela do computador e o Teclado Óptico Ocular digita seqüências de caracteres através do mapeamento dos movimentos oculares do usuário. Ao final dessa dissertação, apresentamos os resultados de alguns testes para mostrar a eficácia dos modelos adotados no desenvolvimento do nosso protótipo e a possibilidade real do Dispositivo Rastreador auxiliar na Inclusão Digital de indivíduos sem a mobilidade dos seus membros superiores.

---

# Abstract

---

This dissertation presents the development of an prototype for a Eye Tracker Device that allows the interaction between the user and the computer by means of eye movements. The Eye Tracking is the main topic of this work. By using Digital Imaging Processing techniques we aim to detect the user's point of regard in the computer screen. Our prototype was developed based on the techniques provided by photo- and video-oculography, for to detect of the iris of the user from images and real-time video. The main drawback caused by the use of iris detection to estimate the user's point of regard is the low vertical accuracy that captured images provided when the iris is partially covered by eyelids. The solution adopted in this work to mitigate this problem was the use of algorithms that detect circles based on the Hough Transform, which not only detect the user's iris with a high accuracy but also reached a processing time rate of 98 miliseconds per analyzed image. The homemade hardware of the Eye Tracker prototype was assembled using a video capture device that was mounted in front of the right eyeglass lens of a plastic safety glasses. Among with the hardware, this works also presents some software that emulate standard computer input devices (mouse and keyboard) that allows individuals with physical disabilities to use the computer through eye movements. The main applications created were the "Optical Ocular Mouse" and the "Optical Ocular Keyboard". The Optical Ocular Mouse places the mouse pointer in the region of the computer screen that the user is looking at and the Optical Ocular Keyboard inserts a sequence of characters that were mapped by the user's eyes movements. At the end of this dissertation we presented the results of some tests to show the efficacy of the adopted models for the development of our prototype and a concrete opportunity for the Eye Tracker Device to help the digital inclusion of individual that does not possess motion capabilities on the whole body.

**Keywords:** Eye Tracker Devices, Eye Tracking, Digital Imaging Processing, Hough Transform, Individuals with Physical Disabilities, Digital Inclusion

---

# Sumário

---

<b>LISTA DE FIGURAS.....</b>	<b>XIII</b>
<b>LISTA DE TABELAS .....</b>	<b>XVII</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>XVIII</b>
<b>CAPÍTULO 1: INTRODUÇÃO .....</b>	<b>19</b>
1.1 – Problema .....	21
1.2 – Proposta de Solução .....	21
1.3 – Fases do Projeto .....	22
1.4 – Organização da Dissertação .....	22
<b>CAPÍTULO 2: RASTREIO OCULAR .....</b>	<b>24</b>
2.1 – Sistema Visual Humano .....	24
2.1.1 – O Olho .....	24
2.1.2 – Anatomia Ocular e Fisiologia da Visão.....	25
2.1.3 – Ângulo Visual.....	27
2.1.4 – Movimentos Oculares.....	27
2.2 – Técnicas de Rastreio Ocular .....	28
2.2.1 – Eletro-Oculografia.....	28
2.2.2 – Lentes de Contatos Magnéticas.....	29
2.2.3 – Foto-Oculografia e Vídeo-Oculografia.....	30
2.2.4 – Análise Pela Reflexão da Córnea/Pupila .....	30
2.3 – Dispositivos Rastreadores.....	32
2.4 – Calibração dos Dispositivos Rastreadores.....	33
2.5 – Principais Produtos de Rastreio Ocular.....	36

2.5.1 – MAGIC Pointing.....	36
2.5.2 – Tobii Technology .....	37
2.6 – Considerações Finais .....	38
<b>CAPÍTULO 3: TRANSFORMADA DE HOUGH .....</b>	<b>39</b>
3.1 – Teoria da Transformada de Hough.....	39
3.1.1 – Representação do Espaço de Parâmetros.....	40
3.2 – Exemplos de Detecção de Formas Geométricas .....	40
3.2.1 – Detecção de Linhas .....	41
3.2.2 – Detecção de Círculos .....	46
3.3 – Aprimoramentos da Transformada de Hough Para Detecção de Círculos.....	47
3.3.1 – Parametrização do algoritmo de Duda e Hart .....	47
3.3.2 – Parametrização do algoritmo de Kimme, Ballard e Sklansky.....	49
3.4 – Variações da Transformada de Hough .....	50
3.4.1 – Transformada Adaptativa de Hough.....	50
3.4.2 – Transformada Combinatorial de Hough .....	52
3.4.3 – Transformada Generalizada de Hough.....	52
3.4.4 – Transformada Hierárquica de Hough .....	54
3.4.5 – Transformada Probabilística de Hough.....	55
3.4.6 – Transformada Randômica de Hough.....	56
3.5 – Considerações Finais .....	57
<b>CAPÍTULO 4: MOUSE ÓPTICO OCULAR.....</b>	<b>58</b>
4.1 – Hardware do Mouse Óptico Ocular.....	59
4.1.1 – Montagem do Dispositivo Rastreador .....	60
4.2 – Processo de Calibração.....	61
4.3 – Emulação dos Botões do Mouse .....	62
4.4 – Sistemas Auxiliares.....	63
4.4.1 – Sistema de Detecção de Círculos em Imagens Digitais.....	63
4.4.2 – Sistema de Atalhos .....	64
4.5 – Teclado Óptico Ocular.....	64
4.6 – Considerações Finais .....	66
<b>CAPÍTULO 5: DESENVOLVIMENTO .....</b>	<b>67</b>
5.1 – Viabilidade do Uso da Transformada de Hough.....	67
5.1.1 – Bibliotecas de Processamento Digital de Imagens.....	67
5.1.2 – Implementação dos Algoritmos de Detecção de Círculos.....	69

5.1.3 – Criação da Matriz Acumuladora em Memória.....	70
5.1.4 – Implementação do Algoritmo de Duda e Hart.....	71
5.1.5 – Implementação do Algoritmo de Kimme, Ballard e Sklansky .....	73
5.1.6 – Implementação da Transformada Probabilística de Hough.....	75
5.1.7 – Testes da Viabilidade do Uso da Transformada de Hough.....	75
5.1.8 – Conclusão do Estudo da Viabilidade da Transformada de Hough.....	79
5.2 – Sistema de Detecção de Círculos em Imagens .....	80
5.3 – Sistema de Detecção de Círculos em Vídeos .....	82
5.3.1 – Windows Driver Model (WDM).....	83
5.3.2 – GraphEdit.....	84
5.3.3 – Dispositivo de Captura de Vídeo.....	85
5.4 – Sistema de Calibração do Dispositivo Rastreador.....	87
5.4.1 – Método dos Mínimos Quadrados.....	88
5.4.2 – Cálculo do Ponto Observado Pelo Usuário .....	90
5.5 – Sistema do Mouse Óptico Ocular .....	91
5.6 – Sistema do Teclado Óptico Ocular .....	93
5.7 – Considerações Finais .....	95
<b>CAPÍTULO 6: TESTES E RESULTADOS .....</b>	<b>97</b>
6.1 – Diretivas Para o Uso do Protótipo.....	98
6.2 – Análise da Precisão do Dispositivo Rastreador .....	99
6.3 – Análise do Funcionamento do Mouse Óptico Ocular .....	103
6.4 – Análise de Desempenho do Teclado Óptico Ocular.....	104
6.5 – Considerações Finais .....	107
<b>CAPÍTULO 7: CONCLUSÕES.....</b>	<b>108</b>
7.1 – Trabalhos Futuros.....	110
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>112</b>
<b>ANEXO I: IMAGENS PROCESSADAS NO TRABALHO.....</b>	<b>116</b>
<b>ANEXO II: AUTORIZAÇÕES.....</b>	<b>131</b>
<b>ANEXO III: TABELAS DE TESTES.....</b>	<b>139</b>

---

# Lista de Figuras

---

Figura 1: Reflexão da imagem capturada na retina do globo ocular.....	25
Figura 2: Anatomia de um olho humano (Wikimedia Commons – domínio público) [38].....	26
Figura 3: Exemplo de fixações e movimentos sacádicos sobre o texto (Wikimedia Commons – domínio público) [22].....	28
Figura 4: Imagens processadas através da técnica de Foto-Oculografia [42].....	30
Figura 5: Exemplo do efeito da Reflexão de Purkinje sob a pupila de um indivíduo .....	31
Figura 6: Uma relação entre os LEDs infravermelhos do dispositivo e o polígono refletivo no olho do usuário [45] .....	32
Figura 7: Exemplo do efeito da iluminação com quatro LEDs e o polígono criado na pupila do indivíduo .....	32
Figura 8: Modelo baseado em mapeamento linear sugerido por Daunys e Ramanauskas [7][36] .	35
Figura 9: Segmento de reta com quatro pontos no espaço da imagem .....	41
Figura 10: Resultado no espaço de parâmetros do processo de detecção de linhas .....	42
Figura 11: Segmento de reta detectado através da Transforma de Hough.....	43
Figura 12: Linha vertical no espaço da imagem.....	43
Figura 13: Resultado no espaço de parâmetros do processo de detecção de linhas verticais.....	44
Figura 14: Representação de uma linha utilizando coordenadas polares.....	45
Figura 15: Resultado no espaço de parâmetros utilizando coordenada polar.....	46
Figura 16: Circunferência de raio (r) e centro (a, b).....	47

Figura 17: Incremento de possíveis centros de círculos.....	48
Figura 18: Processo de detecção de círculos do algoritmo de Duda e Hart [10].....	48
Figura 19: Processo de detecção de círculos do algoritmo de Kimme, Ballard e Sklansky [18].....	50
Figura 20: Fluxograma teórico da Transformada Adaptativa de Hough.....	51
Figura 21: Geometria utilizada na Transformada Generalizada de Hough.....	53
Figura 22: Pirâmide da Transformada Hierárquica de Hough .....	55
Figura 23: Processo de detecção de círculos com a Transformada Randômica de Hough .....	57
Figura 24: Montagem do protótipo do Dispositivo Rastreador (óculos e micro-câmera de captura de vídeo).....	60
Figura 25: Instalação do receptor de vídeo e áudio ( <i>wireless</i> ) com a placa de captura de vídeo ( <i>framegrabber</i> ).....	60
Figura 26: Dispositivo Rastreador instalado em um computador do tipo <i>notebook</i> .....	61
Figura 27: Exemplo de calibração em um monitor com resolução de 1280x800 .....	61
Figura 28: Criação do espaço de parâmetros tridimensional segundo o algoritmo de Dorini e Rocha [8].....	70
Figura 29: Criação do espaço de parâmetros segundo os algoritmos de Powell [34] e Pistori <i>et al</i> [32]. .....	70
Figura 30: Processo de liberação dos recursos alocados na representação do espaço de parâmetros na memória principal.....	71
Figura 31: Detecção de círculos segundo a parametrização de Duda e Hart [10].....	71
Figura 32: Imagem analisada com o algoritmo da Figura 31 e seu respectivo espaço de parâmetros bidimensional .....	72
Figura 33: Visualização tridimensional do espaço de parâmetros da Figura 32 .....	72
Figura 34: Imagem analisada com o algoritmo da Figura 37 e seu respectivo espaço de parâmetros bidimensional .....	73
Figura 35: Visualização tridimensional do espaço de parâmetros da Figura 34 .....	73
Figura 36: Detecção de círculos segundo a parametrização de Kimme, Ballard e Sklansky [18]... ..	74

Figura 37: Incrementos do conjunto $C_x$ entre um ângulo de $45^\circ$ e o ângulo oposto de $-45^\circ$ .....	74
Figura 38: Conjunto de elementos incrementados com a parametrização de Kimme, Ballard e Sklansky [18].....	75
Figura 39: Seleção randômica dos elementos de imagens analisados no processo da Transformada Probabilística de Hough .....	75
Figura 40: Amostra das imagens utilizadas no processo de detecção da circunferência da íris em imagens digitais .....	76
Figura 41: (a) 30.000 pixels analisados na imagem; (b) 8.400 pixels analisados na imagem.....	77
Figura 42: Número de círculos identificados nas imagens analisadas .....	77
Figura 43: Número de incrementos na matriz acumuladora .....	78
Figura 44: Tempo médio de processamento dos algoritmos da Transformada de Hough .....	79
Figura 45: (a) imagem original (b) imagem em tons de cinza (c) imagem com Filtro de Sobel (d) resultado final .....	80
Figura 46: Janela com as estatísticas do processamento da imagem de entrada.....	81
Figura 47: Visualização bidimensional do espaço de parâmetros (a) raio = 39 (b) raio = 40 e (c) raio = 41.....	81
Figura 48: Visualização tridimensional do processamento de uma imagem com a parametrização de Duda e Hart.....	82
Figura 49: Diagrama de processamento de vídeos com filtros DirectShow .....	83
Figura 50: Ambiente de trabalho do GraphEdit com o filtro da Transformada de Hough.....	84
Figura 51: Renderizador do GraphEdit.....	84
Figura 52: Seleção do dispositivo de captura de vídeo .....	85
Figura 53: Detecção de bola de massagem terapêutica com raio entre 50 e 53 pixels.....	85
Figura 54: Detecção da íris em imagens da região do olho do usuário capturadas em tempo real.....	86
Figura 55: Duas situações de Rastreo Ocular: (a) com má iluminação ambiente e (b) com boa iluminação ambiente.....	87
Figura 56: Criação da matriz S no Método dos Mínimos Quadrados.....	89

---

Figura 57: Criação da matriz pseudo-inversa de S .....	89
Figura 58: Cálculo das incógnitas utilizadas no Método dos Mínimos Quadrados e a criação da matriz T .....	90
Figura 59: Estimativa do ponto observado pelo usuário .....	91
Figura 60: Emulação do duplo clique do botão esquerdo do mouse.....	92
Figura 61: Sistema de atalhos para os principais programas instalados no Sistema Operacional ..	93
Figura 62: Exemplo dos quatro mapeamentos utilizados pelo Teclado Óptico Ocular .....	94
Figura 63: Visualização das nove regiões virtuais utilizadas pelo sistema do Teclado Óptico Ocular .....	95
Figura 64: Padrão de posicionamento entre o usuário e o monitor .....	98
Figura 65: Regiões virtuais dos monitores utilizados nos testes de precisão do Dispositivo Rastreador.....	100

---

# Lista de Tabelas

---

Tabela 1: Cálculo do parâmetro (b) no intervalo de (a) e a indicação da célula incrementada na matriz (A).....	42
Tabela 2: Cálculo dos parâmetros (a) e (b) no processo de detecção de linhas verticais com a equação <i>slope-intercept</i> .....	44
Tabela 3: Resultados obtidos para cada voluntário nos testes da precisão do Dispositivo Rastreador.....	101
Tabela 4: Diferença da precisão com suaves movimentos sob o eixo x.....	102
Tabela 5: Diferença da precisão com suaves movimentos sob o eixo y.....	102
Tabela 6: Diferença da precisão com suaves movimentos sob o eixo z.....	103
Tabela 7: Informações adquiridas durante o teste de funcionamento do Mouse Óptico Ocular	104
Tabela 8: Tempo médio (tm) e tempo gasto (mm:ss) nas tentativas de escrever as palavras “casa” e “faqueiro” .....	105
Tabela 9: Tempo médio (tm) e tempo gasto (mm:ss) nas tentativas de escrever as frases “quero água” e “hoje o dia está lindo” .....	106

---

## ***Lista de Abreviaturas e Siglas***

---

ATI	Assistive Technology Inc.
CI	Calibração Interpolada
CL	Calibração Linear
CMNL	Calibração com Mapeamento Não-Linear
CSO	Calibração de Segunda Ordem
DCV	Dispositivo de Captura de Vídeo
EH	Espaço de Hough
EOG	Eletro-Oculografia
FOG	Foto-Oculografia
FPF	Fundação Desembargador Paulo Feitoza
IHC	Interação Humano-Computador
INPE	Instituto Nacional de Pesquisas Espaciais
PDA	Assistente Pessoal Digital
PDI	Processamento Digital de Imagens
PNE	Portadores de Necessidades Especiais
MMQ	Método dos Mínimos Quadrados
MOO	Mouse Óptico Ocular
RCA	Radio Corporation of America
RP	Reflexão de Purkinje
SO	Sistema Operacional
SVH	Sistema Visual Humano
TCH	Transformada Clássica de Hough
TCBH	Transformada Combinatorial de Hough
TGH	Transformada Generalizada de Hough
TH	Transformada de Hough
THH	Transformada Hierárquica de Hough
TOO	Teclado Óptico Ocular
TPH	Transformada Probabilística de Hough
TRH	Transformada Randômica de Hough
USP	Universidade de São Paulo
VOG	Video-Oculografia
WDM	Windows Driver Model
WIMP	Windows, Icons, Menus and Pointing

---

# Introdução

---

**A** Interação Humano-Computador (IHC) estuda o processo de comunicação entre o usuário e o computador. O processo de interação é realizado através de uma interface, composta por *hardware* e *software* de naturezas diversas: como monitores sensíveis ao toque, luvas sensoriais, capacetes de realidade virtual, equipamentos avançados de acústica, dispositivos hápticos, interfaces por voz, teclados e mouses adaptados entre outros. Segundo Moran [23], a interface de usuário deve ser entendida como sendo a parte de um sistema computacional com a qual a pessoa entra em contato – física, perceptiva ou conceitualmente.

Usuários podem apresentar diferentes aptidões quanto à utilização de sistemas computacionais. Devido a isso, é necessário durante o projeto da interface levar em consideração as dificuldades que ele venha a ter ao utilizá-la. Uma interface adaptável permite ajustar certas funcionalidades do sistema às necessidades específicas dos usuários.

Uma interface exerce um papel fundamental no processo de acessibilidade, fornecendo meios para que portadores de algum tipo de necessidade especial (visual, neurológica, auditiva ou física) possam usufruir dos recursos computacionais de modo mais natural. Conhecendo algumas limitações humanas e restrições das tecnologias existentes, pode-se oferecer aos usuários um meio alternativo de interação com computadores, como através de Dispositivos Rastreadores aplicados à Portadores de Necessidades Especiais (PNE).

O Rastreo Ocular (*Eye Tracking*) é um processo utilizado para estimar a posição observada por um indivíduo através do monitoramento dos movimentos dos seus olhos, com o auxílio de um Dispositivo Rastreador (*Eye Tracker*). Tais dispositivos podem ser utilizados em diferentes áreas de atuação, como: 1. pesquisas do sistema visual; 2. diagnósticos neurológicos; 3. *Marketing* e Propaganda, 4. *Design* de produtos, 5. Psicologia, 6. Jornalismo Digital e 7. interfaces.

Os estudos do sistema visual envolvem desde pesquisas básicas da visão até estudos complexos como os movimentos oculares involuntários. Em Neurociência e Psicologia, as pesquisas englobam estudos do comportamento, percepção e busca visual. Nas áreas de *Design* de produtos; e *Marketing* e Propaganda, são analisados a aceitação pelos consumidores de novos produtos, embalagens e anúncios antes de serem lançados no mercado. No Jornalismo Digital são avaliadas as diferenças entre leitura impressa em papel e as notícias publicadas em *websites*. Na área de Interação Humano-Computador, os movimentos oculares são usados como alternativa de interação em sistemas computacionais.

As técnicas de Rastreo Ocular estão baseadas em três vertentes [9]: Eletro-Oculografia (EOG); lentes de contato magnéticas; e na análise de imagens. Os motivos por considerar a análise de imagens mais interessante para a criação do nosso protótipo são: o baixo custo dos equipamentos de aquisição de imagens; os atuais computadores permitirem o desenvolvimento de sistemas com análise de imagens em tempo real; evitar a utilização dispositivos fixos no corpo do indivíduo que emitam energia eletromagnética; e, por ser a técnica mais desafiadora no desenvolvimento de Dispositivos Rastreadores com alto índice de precisão e baixa taxa de erros.

No processo de análise de imagens, é realizada a captura de imagens de um ou de ambos os olhos do usuário através de um Dispositivo de Captura de Vídeo (DCV) que segmenta determinadas regiões de interesse na imagem. O processo pode ser baseado na detecção da circunferência da íris (utilizando a diferença no contraste entre a esclera e a íris) ou da pupila do indivíduo (com iluminação infravermelha para criar uma reflexão na córnea).

Segundo Young e Sheena [47] o rastreo é baseado no cálculo do Ponto de Interesse (*Point of Regard*) com base na posição do olho no espaço, na direção do olhar (*Gaze Tracking*) e no alvo observado pela pessoa. O Rastreo Ocular calcula uma estimativa da projeção do ponto observado pelo usuário no monitor, que vem a servir de entrada de dados no computador permitindo a realização de tarefas que simulam algumas funções do mouse e do teclado através de aplicativos específicos.

Pessoas Portadoras de Necessidades Especiais com deficiências físicas, que por alguma razão perderam a mobilidade dos membros superiores, apresentam algumas dificuldades ao usar computadores pessoais convencionais. Na arquitetura de muitos dispositivos tecnológicos como Assistentes Pessoais Digitais (PDA), câmeras digitais, celulares, computadores, entre outros, não se disponibilizam meios de interação especiais que atendam as dificuldades dos Portadores de Necessidades Especiais.

Na literatura e no mercado podem-se encontrar diversos dispositivos de Rastreo Ocular, porém, em sua maioria, são soluções proprietárias e voltadas para áreas de atuação diferentes de IHC. Os dispositivos mais encontrados são utilizados para identificar diferenças no comportamento visual, quando os usuários são submetidos a testes de avaliação de *layout* de *websites*, propagandas, jornais e revistas e de *design* de produtos.

Conjectura-se que Portadores de Necessidades Especiais poderão ser beneficiados pelo uso de um dispositivo de Rastreo Ocular atuando como interface no uso de computadores domésticos, permitindo a realização de tarefas cotidianas pessoais e até mesmo profissionais com aplicativos de qualquer modalidade que tenham convencionalmente como entrada de dados o teclado ou mouse, gerando a possibilidade de integrar digitalmente o PNE ao meio social.

## 1.1 – Problema

Parte das pesquisas de Rastreo Ocular aplicadas a IHC que usam técnicas de Processamento Digital de Imagens (PDI) optam pelo rastreo da pupila. A pupila não é coberta pela pálpebra durante os movimentos oculares, deixando a sua circunferência na maioria dos movimentos completamente visível. Diferentemente da pupila, a íris é coberta parcialmente pela pálpebra durante os movimentos oculares, dificultando a precisão vertical do Rastreo Ocular.

As técnicas de Rastreo Ocular através do monitoramento da pupila geralmente usam iluminação infravermelha para criar imagens iluminadas da região da córnea [24][25][26][45]. A fim de eliminar a exposição contínua do usuário à iluminação infravermelha, a proposta é o desenvolvimento de um Dispositivo Rastreador com monitoramento da íris. O problema consiste em como proceder para detectar e localizar a íris, com bom grau de precisão e alta performance computacional, mesmo nos casos em que ocorra a cobertura parcial pelas pálpebras.

## 1.2 – Proposta de Solução

De acordo com Gonzales e Woods [13] uma das técnicas de Processamento Digital de Imagens mais eficazes na localização de formas geométricas é a Transformada de Hough (TH). Pesquisas mostram que a Transformada de Hough apresenta desempenho satisfatório na detecção de formas geométricas mesmo em imagens com altos índices de ruídos [1][3][10][11].

A presente dissertação é centrada em uma proposta para desenvolver um protótipo de Dispositivo Rastreador utilizando variações da Transformada de Hough para detecção da

circunferência da íris. É esperado que o dispositivo permita, num futuro próximo, servir como interface para auxiliar Portadores de Necessidades Especiais no uso de computadores, que através dos movimentos oculares poderá simular as principais funções do mouse e do teclado.

### 1.3 – Fases do Projeto

Para o desenvolvimento do trabalho as seguintes tarefas foram realizadas:

- ✓ Estudo da viabilidade do uso da Transformada de Hough para detecção da circunferência da íris no desenvolvimento do Dispositivo Rastreador;
- ✓ Implementação da parametrização de detecção de círculos da Transformada de Hough que melhor se adaptou ao processo de detecção da íris nas imagens capturadas em tempo real;
- ✓ Projeto e construção do dispositivo físico (*hardware*) utilizado no processo de aquisição das imagens dos olhos do usuário;
- ✓ Projeto e implementação do aplicativo lógico (*software*) usados na emulação de algumas funções do mouse e do teclado através do Dispositivo Rastreador;
- ✓ Implementação do processo inicial de calibração que obtém dados válidos dos movimentos oculares, e recalcula durante o uso do Dispositivo Rastreador o ponto aproximadamente observado pelo usuário no monitor;
- ✓ Avaliação do protótipo desenvolvido e a apresentação das conclusões e considerações de desempenho do protótipo criado.

### 1.4 – Organização da Dissertação

Esta dissertação encontra-se organizada em sete capítulos, descritos a seguir:

Capítulo 2 – São introduzidos conceitos sobre Rastreo Ocular, desde o estudo da estrutura ocular humana, as principais técnicas de rastreo, alguns dos Dispositivos Rastreadores existentes, o processo de calibração e a metodologia de análise dos movimentos oculares.

Capítulo 3 – São descritas as principais parametrizações da Transformada de Hough, mostrando detalhadamente os passos do mapeamento do espaço bidimensional da imagem para o espaço  $n$ -dimensional de Hough, bem como os principais métodos de detecção de linhas e círculos com seus respectivos algoritmos de aprimoramento.

Capítulo 4 – É ilustrado em detalhes o processo de montagem da estrutura física do Dispositivo de Captura de Vídeo. Além disso, são abordados os modelos adotados no desenvolvimento dos sistemas “Mouse Óptico Ocular” (MOO) e “Teclado Óptico Ocular” (TOO). A escolha das técnicas e métodos utilizados na etapa de desenvolvimento foram definidas com base nas informações reportadas por outros pesquisadores no desenvolvimento dos seus respectivos Dispositivos Rastreadores.

Capítulo 5 – Descreve o estudo da viabilidade do uso da Transformada de Hough aplicada na detecção da íris em imagens. São apresentadas as etapas do desenvolvimento do protótipo, seguido das implementações dos algoritmos de detecção de círculos aplicados em imagens capturadas em tempo real. É mostrado em detalhes o desenvolvimento do sistema de calibração do Dispositivo Rastreador, finalizando com o desenvolvimento dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular.

Capítulo 6 – São apresentados alguns resultados e aprimoramentos dos algoritmos implementados, acompanhados das análises que permitiram conhecer as falhas e acertos do modelo adotado, são demonstrados a porcentagem de acerto dos algoritmos para análise de imagens e os parâmetros adotados para execução dos algoritmos desenvolvidos.

Capítulo 7 – São reportadas contribuições, dificuldades, conclusões sobre os resultados obtidos, propostas de trabalhos futuros e considerações finais dessa dissertação.

---

# Rastreio Ocular

---

**O**s primeiros estudos sobre Rastreio Ocular datam do século XIX, onde a análise dos movimentos oculares eram realizadas através de observações diretas nos olhos dos indivíduos. De acordo com Paulson [30], no princípio tinha-se a impressão que ao ler um determinado parágrafo os olhos percorriam o texto continuamente do início ao final da linha. Porém, ao contrário do que se imaginava, a leitura é composta de pequenas paradas chamadas fixações e rápidos deslocamentos chamados movimentos sacádicos. Somente no século XX surgiram os primeiros Dispositivos Rastreadores que responderam a uma série de questões levantadas como as regressões sacádicas (Figura 3).

Para dar continuidade na discussão sobre Rastreio Ocular é necessário um estudo detalhado sobre a estrutura ocular humana. Os tópicos a seguir apresentam em detalhes as principais partes da anatomia ocular humana, obtidas a partir de evidências neurofisiológicas e psicofísicas do seu funcionamento.

## 2.1 – Sistema Visual Humano

De acordo com Duchowski [9], o complexo Sistema Visual Humano (SVH) inicia-se nos olhos e se estende por várias estruturas internas cerebrais até as regiões do córtex. É um dos principais sistemas sensoriais que permite aos seres humanos agirem adequadamente ao meio em que vivem, envolvendo diversas funções como localizar, reconhecer e entender objetos, comportamentos, ações ou eventos no ambiente. O primeiro órgão do Sistema Visual Humano é o olho, responsável pela captura de imagens ao redor do ambiente que o indivíduo se encontra.

### 2.1.1 – O Olho

O olho humano tem aproximadamente 25 milímetros de diâmetro e 7 gramas de peso.

É o responsável pela captação da luz refletida pelos objetos do ambiente em que o indivíduo se encontra. Segundo Duchowski [9], a luz penetra o globo ocular através da córnea, atravessando o humor aquoso até atingir a pupila e o cristalino. O cristalino funciona como uma lente de focalização que converge os raios luminosos para um ponto focal sobre a retina. Cerca de cem milhões de células fotossensíveis na retina transformam a luz em impulsos eletroquímicos, que através do nervo óptico são transmitidas até o cérebro. No córtex visual ocorre o processamento das imagens recebidas pelos olhos gerando a sensação visual humana.

As imagens projetadas na retina consistem em imagens inversas do objeto visualizado, igualmente a uma câmera fotográfica conforme ilustra na Figura . De acordo com Gonzales e Woods [13], isto ocorre com os sistemas ópticos quando os alvos estão dispostos além das suas distâncias focais. No Sistema Visual Humano, o cérebro aprendeu a interpretar corretamente a imagem visualizada invertendo a imagem do objeto para a sua posição original, dando a sensação que a imagem é projetada na retina em sua posição normal.

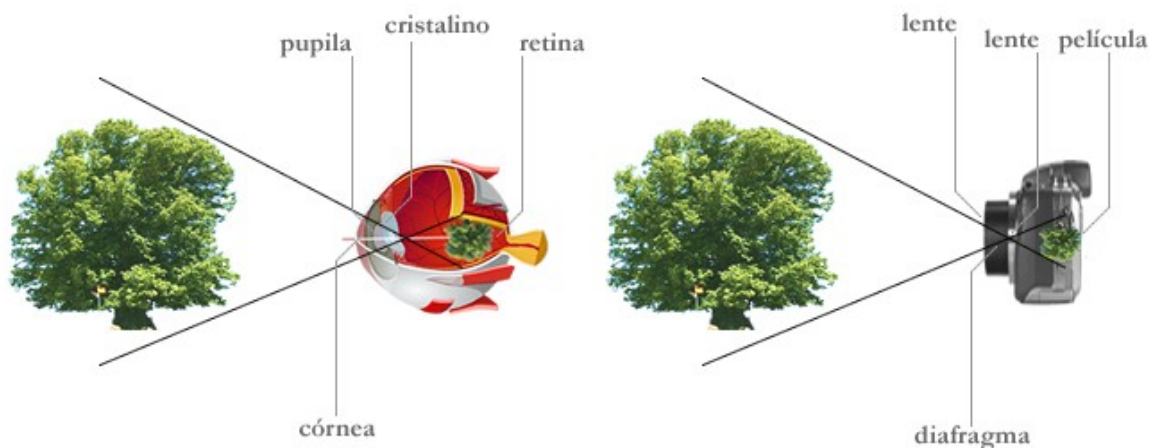


Figura 1: Reflexão da imagem capturada na retina do globo ocular

Os raios luminosos projetados na retina encontram-se precisamente na fóvea central, que é circundada pela mácula, proporcionando uma visão nítida dos objetos focalizados. Por ser um instrumento óptico, o propósito principal do olho humano é formar uma imagem na retina de forma nítida. Alguns indivíduos apresentam ametropia, que ocasiona alguns defeitos na focagem das imagens na retina, como miopia, hipermetropia e astigmatismo. As imagens refletidas na retina que apresentam borramentos ou distorções podem ser corrigidas com o uso de lentes oftálmicas para compensar tais deficiências visuais.

### 2.1.2 – Anatomia Ocular e Fisiologia da Visão

Duchowski [9] mostra que o globo ocular, ilustrado na Figura 2, é composto por três

camadas: externa, média e interna. A camada externa, conhecida como esclera, é a responsável pela sustentação e proteção da estrutura ocular. A esclera tem uma coloração branca em quase sua totalidade e na região da córnea uma coloração transparente. Na região da camada externa encontram-se os músculos extra-oculares responsáveis pelos movimentos dos globos oculares. A camada média apresenta uma estrutura vascular pigmentada, onde se encontram a íris e a pupila. Chamada de coróide é a responsável pela nutrição das estruturas internas do globo ocular.

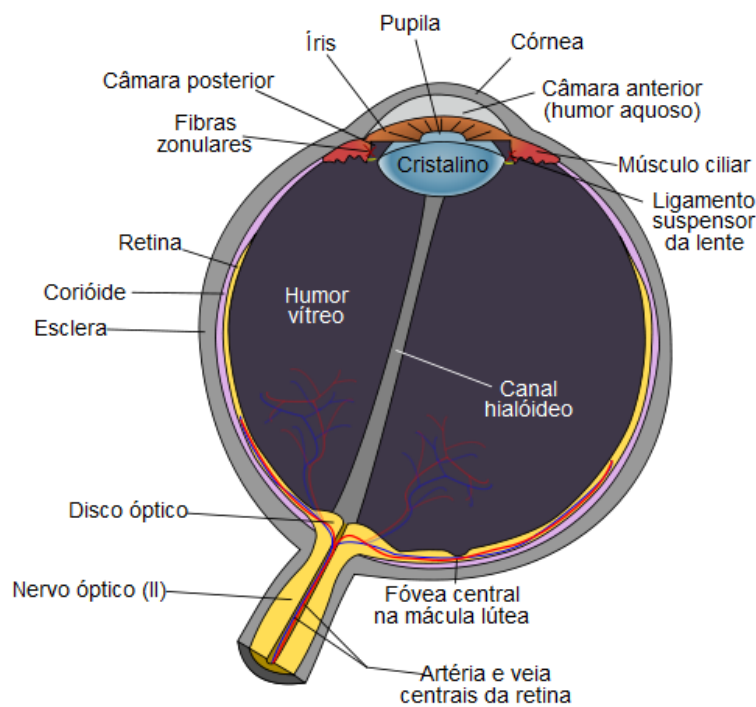


Figura 2: Anatomia de um olho humano (Wikimedia Commons – domínio público) [38]

A camada interna é a camada sensorial responsável pela visão, formada principalmente pela retina. A retina é composta de várias células fotossensíveis, que quando excitadas pela energia luminosa estimulam células nervosas adjacentes que geram impulsos nervosos transmitidos pelo nervo óptico ao cérebro. Essas células são classificadas em dois tipos: cones e bastonetes. Os cones são capazes de distinguir cores, através de cones específicos que excitam cada uma das cores do sistema de percepção tricromática. As imagens fornecidas pelos cones são mais nítidas e ricas em detalhes.

Apesar de apresentar imagens com menor resolução visual, os bastonetes são mais sensíveis à luz do que os cones. Nas situações de baixa luminosidade a visão humana passa a depender exclusivamente dos bastonetes, a chamada visão de penumbra ou visão noturna. Ainda na camada interna encontra-se o ponto cego, que recebe esse nome por ser insensível a luz por

não ter nenhum cone ou bastonete. Qualquer imagem que se forme nesse ponto não será visível pelo indivíduo. O ponto cego contém a saída do nervo óptico e os vasos sanguíneos da retina.

### 2.1.3 – Ângulo Visual

A fóvea é uma das regiões mais importantes do globo ocular. É a área de maior acuidade visual da retina onde se concentram o maior número de cones da estrutura ocular. A imagem do objeto focalizado pelo indivíduo é projetada na região da fóvea em alta resolução. Essa região contém apenas cones e a luz atinge essas células fotossensíveis sem passar pelas demais camadas da retina. A fóvea não está localizada exatamente no eixo óptico do olho, definida pela semi-reta que intercepta o centro do globo ocular e da pupila. O eixo que define a atenção visual do indivíduo é definido pela semi-reta que passa pelo centro da fóvea e da pupila.

Segundo Barcelos [2], o processo de Rastreamento Ocular apresenta uma importante limitação na tentativa de identificar o ponto exato observado pelo indivíduo. A fóvea cobre uma área de aproximadamente 1 grau no ângulo visual. Isso significa que Dispositivos Rastreadores que usam técnicas de Processamento Digital de Imagens terão uma imprecisão correspondente ao tamanho do polegar do indivíduo visto por ele com seu braço estendido. Os movimentos microsacádicos é outro fator que poderá ocasionar ruídos no processo de rastreamento.

### 2.1.4 – Movimentos Oculares

Seis músculos ligados à superfície externa do globo ocular são os responsáveis pelos movimentos oculares. Com a contração e o relaxamento desses músculos, os olhos podem seguir objetos que se deslocam no ambiente, bem como explorar um determinado campo visual. Cada movimento ocular exige que os seis músculos trabalhem em conjunto, embora cada caso exija que determinados músculos desempenhem um papel mais importante que os demais.

Quando um dos músculos se contrai, o globo ocular gira em sua direção. Os olhos podem se mover em qualquer direção, sendo em média 50° para cima, 35° para baixo, 45° para o lado externo e 50° para o lado interno em direção ao nariz [41]. Os dois olhos se movem simultaneamente controlados pelo cérebro, que emite instruções precisas para dois ou três músculos se contraírem na medida exata, realizando o movimento apropriado.

Os movimentos oculares são compostos de fixações e movimentos sacádicos, sendo que a maioria das informações adquiridas pelos olhos é através das fixações. Salvucci e Goldberg [40] mostram que o tempo de duração das fixações geralmente é menor que 100 milissegundos e

os movimentos sacádicos são maiores que 300 milissegundos. A Figura 3 mostra a análise dos movimentos oculares sobre um texto de idioma sueco, com círculos representando fixações e as linhas demonstrando as seqüências de movimentos sacádicos.

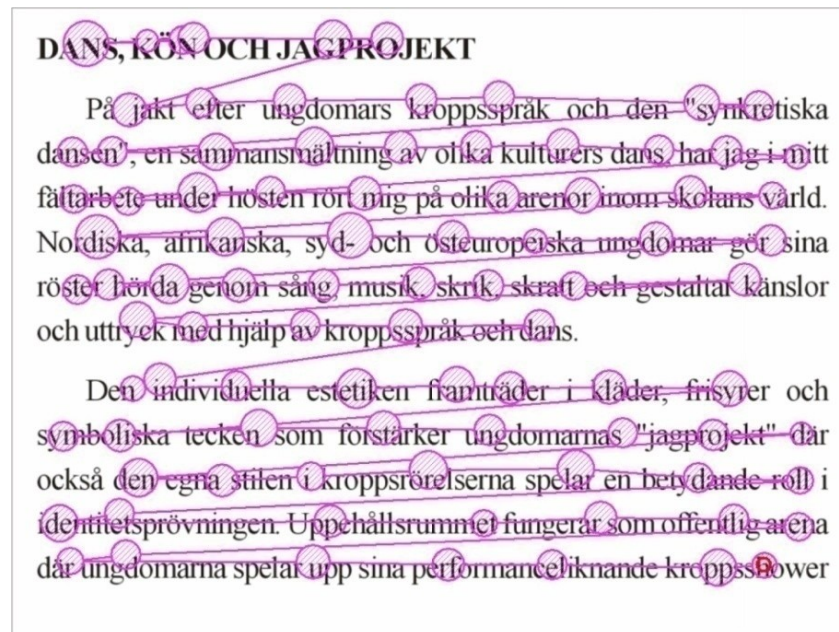


Figura 3: Exemplo de fixações e movimentos sacádicos sobre o texto (Wikimedia Commons – domínio público) [22]

## 2.2 – Técnicas de Rastreo Ocular

Os Dispositivos Rastreadores são desenvolvidos para monitoramento dos movimentos oculares com base em técnicas distintas. Barcelos [2] descreve o agrupamento das técnicas de Rastreo Ocular em duas categorias: intrusivas e não-intrusivas. As técnicas intrusivas usam algum tipo de instrumento de medição, como elétrodos, em contato físico com corpo do indivíduo. E as técnicas não-intrusivas são aquelas trabalham com análise de imagens ou vídeos dos olhos para estimar a posição observada pelo indivíduo.

Duchowski [9] apresenta uma proposta para divisão das técnicas de Rastreo Ocular em quatro categorias, definidas pelo procedimento usado para realizar a análise ocular do indivíduo: Eletro-Oculografia, lentes de contato magnéticas, Foto-Oculografia (FOG) ou Vídeo-Oculografia (VOG), e reflexão infravermelha da córnea. A seguir serão apresentadas algumas das principais pesquisas de Rastreo Ocular seguindo a classificação de Duchowski [9].

### 2.2.1 – Eletro-Oculografia

A Eletro-Oculografia é uma técnica que utiliza elétrodos fixos na região dos olhos para

identificar a posição observada através da variação do potencial elétrico da pele. Ao mover os olhos, uma pessoa consegue gerar potenciais elétricos de até  $200 \mu V$  nos músculos faciais. Essa técnica analisa os movimentos oculares em relação à cabeça. Geralmente não é a mais indicada para estimar o ponto observado pelo indivíduo, a não ser que seja utilizado um rastreador de cabeça em segundo plano.

Uma equipe da Fundação Desembargador Paulo Feitoza (FPF) liderada pelo engenheiro eletricitista Manuel Cardoso criou um Dispositivo Rastreador com técnicas de Eletro-Oculografia [5]. Através de 5 eletrodos fixos na face, o usuário consegue manusear o computador por meio da captura e codificação dos movimentos dos músculos da face e da íris. O dispositivo permite que o usuário mova o cursor com os movimentos oculares, escreva textos com o auxílio do teclado virtual do Sistema Operacional (SO) e simule os cliques dos botões com o piscar dos olhos.

O dispositivo foi lançado como produto de mercado durante a III Feira Internacional da Amazônia em 2006 e encontra-se à venda com valores médios entre R\$ 150,00 e R\$ 200,00. A estudante de enfermagem Luciana Gonçalves Novaes, que no ano de 2003 ficou tetraplégica vítima de uma bala perdida na Universidade Estácio de Sá, foi uma das primeiras usuárias Portadoras de Necessidades Especiais a testar o dispositivo criado pela FPF, que proporcionou tanto a sua inclusão social, quanto a digital.

### 2.2.2 – Lentes de Contatos Magnéticas

Uma das técnicas mais precisas de Rastreamento Ocular envolve o uso de lentes de contato magnéticas. Antigamente, a medida ocular era calculada por um objeto de referência mecânico ou óptico, fixos nas lentes de contato que eram utilizadas pelo indivíduo nos seus olhos. A técnica evoluiu para uma lente de contato mais moderna que trabalha com a captação de energia eletromagnética. Geralmente as lentes de contato rastreadoras são grandes, em comparação as lentes de contato oftálmicas, sendo usadas na região da córnea e esclera. Algumas lentes de contato antigas usavam anéis metálicos para auxiliar no Rastreamento Ocular.

Um bom exemplo dessa técnica é apresentado por Robinson [39]. Utilizando uma lente de contato especial, determina a linha de visão com uma ótima precisão através da reação a campos eletromagnéticos de alta frequência gerados ao redor do usuário. A precisão do Dispositivo Rastreador apresentado por Robinson [39] é de 0,08 graus do ângulo visual. A principal desvantagem dessa técnica é a sua característica invasiva, onde a geração de campos eletromagnéticos ao redor do usuário pode vir a prejudicar a sua saúde com o uso constante.

### 2.2.3 – Foto-Oculografia e Vídeo-Oculografia

As técnicas de Foto-Oculografia e Vídeo-Oculografia trabalham respectivamente com o processamento de imagens e vídeos digitais, analisando determinadas características oculares como a circunferência da pupila, a direção do *limbus* (região de contorno entre a íris e a esclera), e reflexão da córnea através de iluminação infravermelha. Essas técnicas permitem que a análise ocular seja realizada de dois modos: no momento da aquisição das imagens ou posteriormente com imagens guardadas em um dispositivo de armazenamento.

A análise da região do *limbus* permite uma maior facilidade na detecção do centro da íris, dado o alto contraste entre a íris e a esclera. Contudo, nesse caso, a precisão vertical do rastreo é limitada nos momentos em que a íris é parcialmente coberta pela pálpebra. Na análise da circunferência da pupila a limitação vertical é menor, tendo como principal dificuldade o baixo contraste entre a íris e a pupila nos indivíduos de olhos escuros. O ideal é criar imagens com a pupila iluminada através de iluminação infravermelha da córnea.

O dispositivo desenvolvido por Souza *et al.* [42] utiliza técnicas de Vídeo-Oculografia para detectar a posição atual da pupila em cada imagem processada (Figura 4a). Através de morfologia matemática, utiliza a seqüência de limiarização, erosão e dilatação para ter como resultado final uma imagem com um círculo preenchido em preto correspondente a pupila (Figura 4b). A Figura 4 ilustra as imagens registradas do olho esquerdo de um voluntário dos testes de Souza *et al.* [42], e os respectivos resultados do processamento das imagens capturadas.

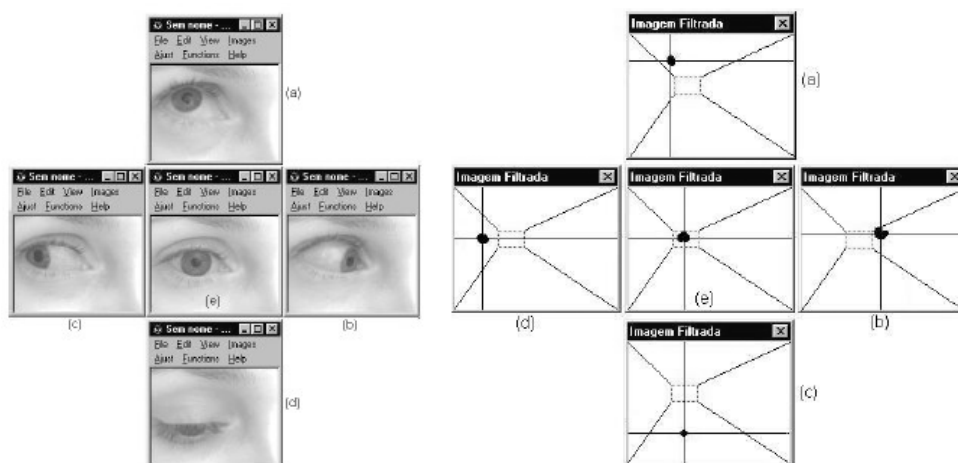


Figura 4: Imagens processadas através da técnica de Foto-Oculografia [42]

### 2.2.4 – Análise Pela Reflexão da Córnea/Pupila

Conhecida como Reflexão de Purkinje (RP), atualmente a técnica de Rastreo Ocular a

partir da reflexão da córnea é uma das mais utilizadas no desenvolvimento de novos Dispositivos Rastreadores. Utilizando iluminação infravermelha alinhada com o eixo óptico da câmera de aquisição, uma imagem iluminada da pupila é criada da mesma forma que o efeito de “olho vermelho” em determinadas fotografias.

A Reflexão de Purkinje pode ser formada em quatro regiões do globo ocular, através das reflexões que ocorrem nos limites do cristalino e da córnea. A diferença da posição entre o centro da pupila e a reflexão da córnea muda simplesmente com os movimentos oculares. O intervalo sobre a reflexão da córnea varia entre  $12^{\circ}\sim 15^{\circ}$ , sendo necessárias correções geométricas para ajustar reflexões que ocorram fora da parte esférica da córnea.

Morimoto *et al.* [24][25][26] desenvolveram um Dispositivo Rastreador para detecção da pupila utilizando uma câmera de aquisição de imagens para computador (*webcam*), integrada a dois jogos de LEDs infravermelho. A luz infravermelha é imperceptível a olho nu, porém o dispositivo de captura é sensível ao comprimento de onda usado pelos LEDs. Um dos jogos de LEDs é ajustado ao eixo óptico da câmera de aquisição para gerar a imagem iluminada da pupila, e o outro é adequado ao anel exterior da câmera gerando uma imagem escura da pupila com um brilho ajustado.

O vídeo adquirido pela câmera de aquisição é composto de quadros intercalados de pupilas iluminadas e escuras, conforme ilustrado na Figura 5. Quando uma imagem com a pupila iluminada é capturada, usa-se a imagem seguinte com a pupila escura aplicando a técnica de subtração. Uma terceira imagem binária é criada com o resultado da diferença das duas primeiras. Com isso, a terceira imagem irá apresentar apenas a circunferência da pupila, finalizando o processo de detecção com o cálculo do centro dessa circunferência.

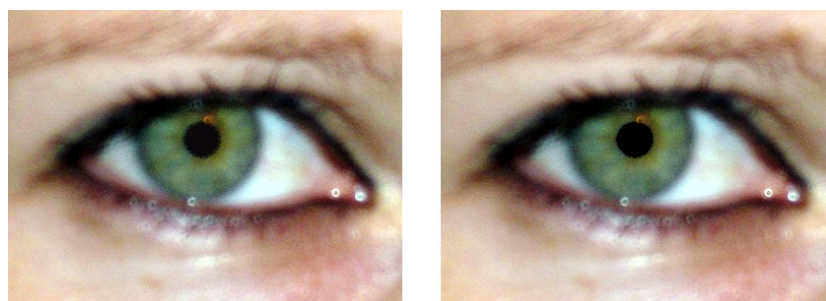


Figura 5: Exemplo do efeito da Reflexão de Purkinje sob a pupila de um indivíduo

Outro sistema desenvolvido com a técnica de Reflexão de Purkinje é descrito por Yoo *et al.* [45], sendo formado por cinco LEDs infravermelho e uma câmera de aquisição. Quatro LEDs estão localizados nos cantos do monitor, iluminando a superfície da córnea. O quinto LED está

localizado no eixo da câmara de aquisição com objetivo de criar uma imagem com a pupila iluminada. Na Figura 6, os pontos  $A$ ,  $B$ ,  $C$  e  $D$  são os reflexos na pupila do usuário provocados respectivamente pelos LEDs  $A'$ ,  $B'$ ,  $C'$  e  $D'$  do monitor.

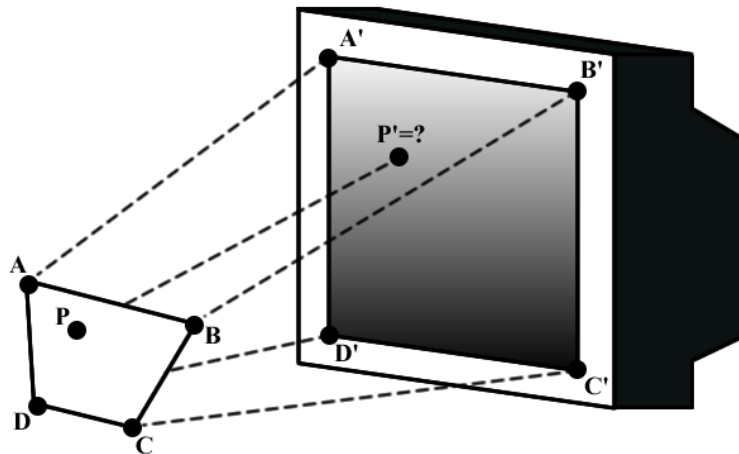


Figura 6: Uma relação entre os LEDs infravermelhos do dispositivo e o polígono refletido no olho do usuário [45]

O ponto  $P$  é o centro da pupila e sempre se encontra dentro do polígono refletido pelo conjunto de LEDs externos. Quando o usuário está posicionado em frente ao monitor, o polígono iluminado é similar ao ilustrado na Figura 7. A direção do olhar do usuário, representado ponto  $P'$ , é computada sem nenhuma relação geométrica das distâncias entre o olho, a câmara e o monitor no espaço tridimensional. Nesse sistema a precisão do rastreo é de aproximadamente  $2.0^\circ$  do ângulo de visão.



Figura 7: Exemplo do efeito da iluminação com quatro LEDs e o polígono criado na pupila do indivíduo

### 2.3 – Dispositivos Rastreadores

Vários Dispositivos Rastreadores são desenvolvidos baseados nas técnicas apresentadas no item 2.2. As vantagens dos dispositivos variam do embate entre a precisão de rastreo versus a taxa de processamento. Embora cada um tenha seus benefícios e suas desvantagens, de acordo com Duchowski [9] as técnicas de Vídeo-Oculografia, Foto-Oculografia e Reflexão de Purkinje

são as mais utilizadas no desenvolvimento de novos dispositivos.

A principal vantagem dos dispositivos baseados em análise de vídeos em relação aos demais dispositivos é que sua estrutura é relativamente não-invasiva, ou seja, o seu uso a longo prazo não prejudica a estrutura ocular do usuário. A precisão do rastreamento desses dispositivos é de aproximadamente 1° do ângulo visual sobre 30° de intervalo de visualização. A limitação é a frequência da taxa de vídeo de aproximadamente 60 Hz, mas isso é compensado com a latência entre a captura da imagem e o seu processamento, tendo pelo menos 16ms para que isso ocorra.

Os atuais Dispositivos Rastreadores usam câmeras de aquisição de baixo custo em conjunto com avançadas técnicas de Processamento Digital de Imagens. Duchowski [9] classifica os Dispositivos Rastreadores em dois tipos: Rastreadores Fixos na Mesa e Rastreadores Fixos na Cabeça. A ótica de ambos os dispositivos são essencialmente as mesmas, tendo uma relativa diferença nos seus tamanhos.

Alguns Rastreadores Fixos na Mesa podem ser confundidos com monitores de tela plana, tendo como distinção um conjunto de câmeras e LEDs infravermelhos acoplados na sua estrutura. Em alguns casos isso é intencional como, por exemplo, os estudos de Jornalismo Digital que visam analisar basicamente a diferença entre a leitura de notícias impressas e digitais. O objetivo é avaliar como leitores escolhem as notícias que mais os agradam, seus principais interesses na leitura, a sua reação ao *layout* e às cores da publicação, entre outros.

Os Rastreadores Fixos na Cabeça tem uma estrutura similar aos óculos utilizados em Realidade Aumentada. Nesse tipo de dispositivo, a câmera de aquisição das imagens está localizada estrategicamente na estrutura do rastreador, analisando os movimentos oculares em relação à cabeça do usuário. Para evitar falhas no processo de Rastreamento Ocular, o usuário deverá permanecer com a cabeça imóvel durante o uso desse tipo de dispositivo.

## 2.4 – Calibração dos Dispositivos Rastreadores

Uma das etapas mais importantes do processo de Rastreamento Ocular é o procedimento de calibração. A calibração tem por objetivo estabelecer relações entre um conjunto de pontos observados pelo usuário e toda a área do monitor que possa a vir observar enquanto usa o Dispositivo Rastreador. É um processo de configuração que ocorre antes da utilização do dispositivo, e em geral exige que o usuário observe entre 5 e 25 pontos. Ohno *et al.* [28] explicam que a maioria dos dispositivos precisam de algum tipo de calibração devido as seguintes razões:

- ✓ Diferentes tamanhos de globos oculares nos indivíduos em idade adulta (a diferença no raio pode chegar até 10%);
- ✓ Dificuldade de medir a posição da fóvea através da técnica de Foto-Oculografia.

Ramanauskas [37] realizou um estudo com objetivo de comparar as quatro principais técnicas de calibração usadas em Dispositivos Rastreadores: Calibração Linear (CL), Calibração de Segunda Ordem (CSO), Calibração Interpolada (CI) e Calibração com Mapeamento Não-Linear (CMNL). A técnica mais simples é a Calibração Linear, que realiza o mapeamento em 5 pontos no monitor (os quatro cantos e um ponto central). O mapeamento entre as medidas oculares nas imagens analisadas e as coordenadas do monitor é realizado usando:

$$\begin{cases} s_x = a_0 + a_1 \cdot x \\ s_y = b_0 + b_1 \cdot x \end{cases} \quad (2.1)$$

onde, as variáveis  $(s_x, s_y)$  correspondem as coordenadas do monitor e  $(x, y)$  ao centro da circunferência analisada no globo ocular. Os coeficientes  $a_0$ ,  $a_1$ ,  $b_0$  e  $b_1$  não são conhecidos e podem ser encontrados através do Método dos Mínimos Quadrados (MMQ). Já a Calibração de Segunda Ordem pode ser usada para o mapeamento do conjunto de 9 ou 25 pontos. A equação polinomial usada nesse caso é definida por:

$$\begin{cases} s_x = a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot xy + a_4 \cdot x^2 + a_5 \cdot y^2 \\ s_y = b_0 + b_1 \cdot x + b_2 \cdot y + b_3 \cdot xy + b_4 \cdot x^2 + b_5 \cdot y^2 \end{cases} \quad (2.2)$$

onde, os coeficientes  $a_0$ ,  $a_1$ ,  $a_2$ ,  $a_3$ ,  $a_4$ ,  $a_5$ ,  $b_0$ ,  $b_1$ ,  $b_2$ ,  $b_3$ ,  $b_4$  e  $b_5$  também não são conhecidos, sendo necessário o MMQ para encontrar os seus respectivos valores. A Calibração Interpolada foi desenvolvida por Zhu e Yang [50] em 2002. Seu conceito é o mapeamento de uma matriz bidimensional entre o canto interno do olho e o centro da íris como ângulo visual. Depois do processo de calibração, o Rastreo Ocular é computado por interpolação.

Por exemplo, suponha que a coordenada do monitor é representada pelo ponto  $P_1$  e o elemento de referência no olho do indivíduo pelo ponto  $P_2$ . Os valores de  $P_1$  e  $P_2$  são respectivamente  $\{(s_{x1}, s_{y1}), (x_1, y_1)\}$  e  $\{(s_{x2}, s_{y2}), (x_2, y_2)\}$ . Ao realizar a medida da distância entre o canto do olho e o centro da íris, representados por  $(x, y)$ , a coordenada do monitor é calculada através de:

$$\begin{cases} s_x = s_{x1} + \frac{x - x_1}{x_2 - x_1} (s_{x2} - s_{x1}) \\ s_y = s_{y1} + \frac{y - y_1}{y_2 - y_1} (s_{y2} - s_{y1}) \end{cases} \quad (2.3)$$

A Calibração com Mapeamento Não-Linear, desenvolvido por Daunys e Ramanauskas

[7][36], constrói um modelo matemático do olho do usuário durante a primeira etapa de interação com o sistema. Essa técnica é adaptável para possíveis movimentos da cabeça, recalculando os valores dos parâmetros quando necessário. Sua aplicação é voltada para Dispositivos Rastreadores Fixos na Mesa, conforme ilustra a Figura 8.

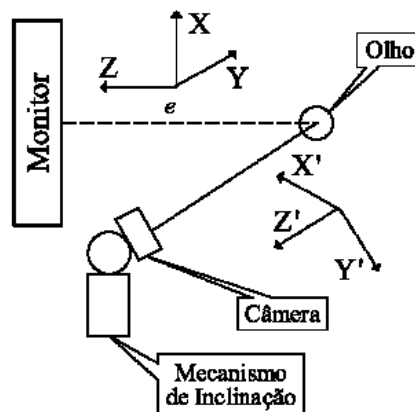


Figura 8: Modelo baseado em mapeamento linear sugerido por Daunys e Ramanauskas [7][36]

O modelo matemático do canto interno do olho ao centro da íris é usado como parâmetro principal na Calibração com Mapeamento Não-Linear. Os parâmetros  $X$  e  $Y$  são coordenadas do ambiente em que o usuário se encontra, sendo orientadas pelo eixo  $Z$  que é perpendicular ao monitor. O sistema de coordenadas  $X'Y'Z'$  é alinhado ao Dispositivo de Captura de Vídeo. Os movimentos oculares são descritos pelo sistema de coordenadas do ambiente, e as imagens são capturadas por projeção linear da estrutura ocular para o plano  $X'Y'$ .

Um vetor  $e$  é alinhado ao eixo óptico, e sua projeção é proporcional no plano  $X'Y'$  para a medida do centro da pupila. A orientação do sistema  $X'Y'Z'$  é definida por dois ângulos. Um dos seus parâmetros é descrito entre o final do vetor  $e$  e os movimentos oculares em relação ao Dispositivo de Captura de Vídeo. As coordenadas do rastreamento em relação ao monitor dependem da distância dos olhos para a tela. Nesse caso deve ser considerado que o eixo óptico não passa pela fóvea, sendo outro parâmetro definido entre os eixos óptico e o de visão. Nessa técnica o número mínimo de parâmetros são seis.

Para calcular a taxa de erros de cada técnica de calibração analisada por Ramanauskas [37], foi utilizada a equação 2.4:

$$error = \sqrt{(x_e - x_m)^2 - (y_e - y_m)^2} \quad (2.4)$$

onde  $(x_e, y_e)$  representam a coordenada do ponto observado pelo usuário no monitor e  $(x_m, y_m)$  a coordenada mapeada através da análise ocular do indivíduo.

A conclusão de Ramanauskas [37] é que não existem diferenças significativas entre a técnica de Calibração com Mapeamento Não-Linear [7] e os modelos de calibração polinomiais. As técnicas de calibração polinomiais não permitem movimentos extras da cabeça, sendo necessária uma nova etapa de calibração caso o movimento exceda um determinado limite dado ao usuário. Esse problema não é encontrado no modelo criado por Daunys e Ramanauskas [7].

O mapeamento da taxa de erros mostrou que a precisão do Dispositivo Rastreador variou entre 1.5° para a Calibração Linear com 5 pontos, e 1° para a Calibração de Segunda Ordem com 25 pontos. Ao final da sua pesquisa, Ramanauskas [37] classificou as técnicas de calibração na seguinte ordem, de acordo com a melhor precisão:

1. Calibração de Segunda Ordem usando um conjunto de 25 pontos;
2. Calibração com Mapeamento Não-Linear usando um conjunto de 25 pontos;
3. Calibração de Segunda Ordem usando um conjunto de 9 pontos;
4. Calibração Interpolada usando um conjunto de 25 pontos;
5. Calibração Linear usando um conjunto de 5 pontos;

O dispositivo desenvolvido por Morimoto *et al.* [24][25][26] usa a técnica de Calibração de Segunda Ordem com 9 pontos. Nesse caso, a precisão varia entre 1.5°~2.0° do ângulo visual. Para o desenvolvimento do Mouse Óptico Ocular optamos por seguir a pesquisa de Ramanauskas [37]. Foi utilizada a técnica de Calibração de Segunda Ordem com 25 pontos no processo inicial de calibração do protótipo criado. No item 4.2 são demonstrados maiores detalhes sobre o processo de calibração do nosso Dispositivo Rastreador.

## 2.5 – Principais Produtos de Rastreo Ocular

O avanço nos estudos de Rastreo Ocular permitiu que ao longo dos anos, Dispositivos Rastreadores fossem facilmente encontrados no comércio. Boa parte desses produtos tem um preço acessível e seu uso está direcionado para diversas áreas de atuação. Dois sistemas de Rastreo Ocular são apresentados nesse tópico, ilustrando algumas características e aspectos como funcionalidade, usabilidade e aprimoramentos.

### 2.5.1 – MAGIC Pointing

O MAGIC Pointing é um dos modernos sistemas de Rastreo Ocular existentes no mercado. Foi desenvolvido por Zhai *et al.* [49] através de uma parceria entre o IBM Almaden Research Center com a Universidade de São Paulo (USP). Utiliza a técnica de reflexão da pupila

para auxiliar o usuário na tarefa de apontamento do cursor do mouse através de Rastreo Ocular. A tarefa é realizada através de uma estimativa da direção do olhar do usuário num determinado ponto do monitor, considerando certo grau de erro.

A análise ocular é realizada quando a atenção visual do usuário muda para um alvo relativamente distante da posição atual do cursor do mouse. De acordo com Barcelos [2], o MAGIC Pointing executa o reposicionamento do cursor do mouse de duas maneiras: através da *estratégia conservadora*, reposicionando o cursor a partir do uso inicial do mouse pelo usuário através de um movimento com as mãos (movimento motor); e com a *estratégia liberal*, reposicionando automaticamente o cursor sem a necessidade da intervenção do usuário.

Vários autores aprimoraram o sistema MAGIC Pointing com objetivo de adicionar novas funcionalidades ao seu uso. Wang, Zhai e Su [48] desenvolveram uma aplicação para digitação de ideogramas do idioma chinês através dos movimentos oculares. Os caracteres selecionados pelo usuário são apresentados em botões como um teclado virtual e a seleção dos botões é feita através do MAGIC Pointing. Para atenuar a precisão limitada do Rastreo Ocular a distribuição dos botões está em um formato de ‘W’, maximizando a distância entre os alvos.

Barcelos [2] desenvolveu uma versão para Linux do Dispositivo Rastreador proposto por Morimoto *et al.* [25]. Esse dispositivo, em conjunto com o MAGIC Pointing permitiu o estudo do contexto de interação entre usuários e sistemas que usam o paradigma “Windows, Icons, Menus and Pointing” (WIMP). Barcelos [2] adicionou uma nova funcionalidade ao MAGIC Pointing, um sistema de reposicionamento do cursor do mouse para a janela mais próxima observada pelo usuário.

### 2.5.2 – Tobii Technology

A Tobii Technology é uma empresa líder no mercado mundial em soluções de dispositivos e sistemas de Rastreo Ocular. Seus produtos são intensamente usados em pesquisas científicas e de propagandas, em aplicações comerciais e também por Portadores de Necessidades Especiais como um meio alternativo de comunicação. Uma parceria com a Assistive Technology Inc. (ATI) tornou a Tobii a principal fornecedora no mundo, de soluções para sistemas voltados a Inclusão Digital de Pessoas Portadoras de Necessidades Especiais.

O produto MyTobii P10 é um Dispositivo Rastreador portátil exclusivo para comunicação através de movimentos oculares. O produto inclui o Dispositivo Rastreador e o computador integrado a um monitor LCD 15”. Em conjunto com o MyTobii Applications, os

Portadores de Necessidades Especiais podem editar e navegar em documentos, receber e enviar *e-mail's*, se comunicar através de *chats*, navegar na *web* e até manipular *games* exclusivos.

Os Dispositivos Rastreadores da Tobii têm auxiliado laboratórios de pesquisas em vários países. As pesquisas envolvem domínios como Psicologia do desenvolvimento humano, IHC, Desenho Industrial, Neuropsicológica e estudos da leitura. Atualmente a Tobii direciona o desenvolvimento de seus produtos para atender necessidades específicas da indústria. Com a evolução das técnicas de Rastreo Ocular nos últimos anos, é possível a partir de agora integrar a comunicação visual do usuário com a indústria automobilística, naval e aeroviária, siderúrgica, setor petrolífero entre outros.

## 2.6 – Considerações Finais

Neste capítulo foram abordados os principais conceitos sobre Rastreo Ocular. O estudo começou com uma pesquisa sobre o Sistema Visual Humano, onde foi possível detectar o problema do objeto refletido na região da fóvea do indivíduo. As técnicas de Rastreo Ocular foram apresentadas de acordo com a classificação sugerida por Duchowski [9]. Também foram apresentados tipos distintos de Dispositivos Rastreadores, algumas técnicas de calibração e produtos de Rastreo Ocular já desenvolvidos e encontrados à venda no mercado.

O próximo capítulo apresenta um estudo teórico sobre a Transformada de Hough, uma alternativa no processo de detecção da íris para o desenvolvimento do sistema de Rastreo Ocular. As principais parametrizações e aprimoramentos da Transformada de Hough são apresentadas, enfocando principalmente o processo de detecção de círculos em imagens digitais.

---

## ***Transformada de Hough***

---

**A** Transformada de Hough é um modelo matemático que detecta formas geométricas em imagens digitais, sendo utilizada em Análise de Imagens, Visão Computacional e Processamento Digital de Imagens. A Transformada Clássica de Hough (TCH) foi desenvolvida por Paul Hough no ano de 1962 e patenteada pela IBM [15]. No início era utilizada para identificar linhas nas imagens analisadas, mas com o passar do tempo foi estendida para detecção de curvas, como círculos e elipses.

Segundo Gonzales e Woods [13], a Transformada de Hough é uma das técnicas mais eficazes para localizar formas geométricas em imagens digitais, detectando objetos mesmo em imagens muito ruidosas. É possível detectar qualquer forma geométrica desde que seja conhecida uma equação matemática simples e de pequena dimensão para representá-la, como a equação geral da reta  $a.x + b.y + c = 0$ .

### **3.1 – Teoria da Transformada de Hough**

Um problema comum no processamento de imagens é a detecção de formas simples como segmentos de retas, círculos e elipses em imagens digitais. De forma geral, é possível segmentar essas formas através de um detector de extremidade para obter os pontos que estão na curva desejada no espaço da imagem. Devido a ruídos na imagem, pode haver divergências entre a forma geométrica na imagem e os pontos de extremidade ruidosos obtidos.

Por estas razões, esse processo de detecção não é trivial para agrupar todas as características de extremidades das formas a serem detectadas na imagem analisada. O propósito da Transformada de Hough é solucionar esse problema, tornando possível o agrupamento dos pontos de extremidade através de um processo de votação, através da parametrização da forma

geométrica a ser detectada na imagem.

O conceito principal da Transformada de Hough é o mapeamento do espaço bidimensional da imagem para uma representação no espaço de parâmetros. O espaço de parâmetros é dividido em células que representam cada elemento da imagem. Com a equação da forma a ser detectada, é realizado o mapeamento através de incrementos nas respectivas células. No final do processo, as células incrementadas mais vezes apresentam os picos relacionados aos valores mais prováveis dos parâmetros da forma detectada na imagem.

### 3.1.1 – Representação do Espaço de Parâmetros

O espaço de parâmetros, também conhecido como Espaço de Hough (EH), é representado na forma de uma matriz acumuladora de números inteiros, onde cada elemento da matriz representa um determinado intervalo no espaço “real” dos parâmetros. A matriz acumuladora é criada em memória explicitamente pelo algoritmo de detecção, da forma a ser detectada, para computar a Transformada de Hough na imagem analisada.

A dimensão da matriz acumuladora é definida pela quantidade de parâmetros da forma geométrica a ser detectada. Por exemplo, na detecção de formas circulares, o círculo é representado por três parâmetros  $(x, y, r)$ , onde  $(x, y)$  representa a posição do centro do círculo no espaço bidimensional e  $(r)$  o seu raio. Nesse caso, para armazenar os valores quantizados dos parâmetros  $x$ ,  $y$  e  $r$  é utilizada uma matriz acumuladora tridimensional.

Já o número de linhas e de colunas da matriz acumuladora é respectivamente igual à altura e à largura da imagem analisada. Após a criação da matriz acumuladora, todas as suas células recebem valor igual a zero. Ao analisar um ponto localizado por um filtro de detecção de bordas na imagem, é realizado o mapeamento para o espaço de parâmetros que define quais células da matriz acumuladora devem ser incrementadas por um.

Ao final do processo, as células da matriz acumuladora com maiores valores acumulados (picos) indicam os parâmetros correspondentes a possíveis formas detectadas na imagem. Nesse processo pode ser usado um “limiar” para estabelecer um valor mínimo de pontos colineares, que define se o valor acumulado na célula corresponde ou não a uma forma geométrica detectada.

## 3.2 – Exemplos de Detecção de Formas Geométricas

Os algoritmos utilizados para a detecção de diferentes formas geométricas com a

Transformada de Hough são similares, sendo necessárias apenas algumas modificações na estrutura do espaço de parâmetros. A forma geométrica mais simples de ser detectada é um segmento de reta, por causa de sua equação e pelo número de parâmetros utilizados.

### 3.2.1 – Detecção de Linhas

Para detecção de segmentos de retas em imagens digitais, Hough [15] utilizou a equação analítica da reta  $y = a.x + b$ , para representação paramétrica de linhas. Ao usar a equação na forma  $y = a.x + b$ , um número infinito de linhas que passam pelo ponto  $(x,y)$  satisfazem a equação para diferentes valores de  $a$  e  $b$ . Uma solução é definir a mesma equação na forma  $b = -a.x + y$ . Desse modo, tem-se uma equação para representar uma linha que passa pelo ponto  $(x,y)$ , onde  $a$  representa a inclinação (*slope*) e  $b$  o ponto de intersecção (*intercept*) com o eixo  $y$ .

No caso do algoritmo descrito por Hough [15], é necessário um acumulador bidimensional ( $A$ ) que corresponde aos valores quantizados de  $a$  e  $b$ . É definido um intervalo válido para  $a$ , entre  $a_{min}$  e  $a_{max}$ . Para cada ponto localizado pelo filtro de detecção de bordas, varre-se o intervalo em  $a$  realizando o cálculo na equação  $b = -a.x + y$ . Se para cada  $a_i$  resulta em uma solução  $b_i$ , a célula  $A(a_i, b_i)$  na matriz acumuladora é incrementada em um. Ao analisar vários pontos que pertencem à mesma linha no espaço da imagem, os seus resultados irão se interceptar no mesmo ponto  $(a', b')$  no espaço de parâmetros.

A Figura 9 ilustra o processo de detecção de linhas descrito por Hough [15] aplicado nos 4 pontos pertencentes ao segmento da imagem. Nesse caso, foi definido para o parâmetro  $a$  um intervalo entre  $[-2, 3]$ . Substituindo-se os valores de  $x$  e  $y$  do ponto analisado na equação  $b = -a.x + y$ , retorna-se para cada valor do intervalo de  $a$  o seu respectivo  $b$ . Os valores de  $a$  e  $b$  são utilizados para incrementar a célula  $A(a, b)$  e traçar um segmento no espaço de parâmetros.

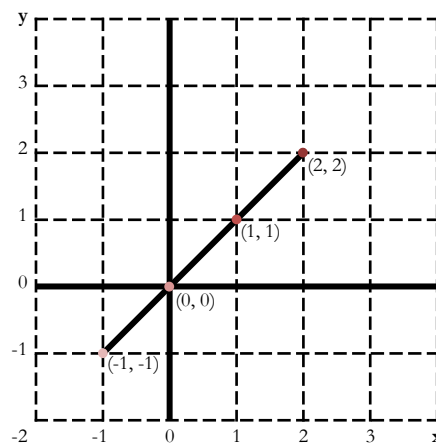


Figura 9: Segmento de reta com quatro pontos no espaço da imagem

A Tabela 1 mostra os valores utilizados para cada ponto analisado na Figura 9 e os parâmetros obtidos através do algoritmo de detecção de linhas criado por Hough [15].

Tabela 1: Cálculo do parâmetro (b) no intervalo de (a) e a indicação da célula incrementada na matriz (A)

X	y	a	b=-a.x+y	A(a, b)
-1	-1	-2	-3	A(-2, -3)
-1	-1	-1	-2	A(-1, -2)
-1	-1	0	-1	A(0, -1)
-1	-1	1	0	<b>A(1, 0)</b>
-1	-1	2	1	A(2, 1)
-1	-1	3	2	A(3, 2)
0	0	-2	0	A(-2, 0)
0	0	-1	0	A(-1, 0)
0	0	0	0	A(0, 0)
0	0	1	0	<b>A(1, 0)</b>
0	0	2	0	A(2, 0)
0	0	3	0	A(3, 0)
1	1	-2	3	A(-2, 3)
1	1	-1	2	A(-1, 2)
1	1	0	1	A(0, 1)
1	1	1	0	<b>A(1, 0)</b>
1	1	2	-1	A(2, -1)
1	1	3	-2	A(3, -2)
2	2	-2	6	A(-2, 6)
2	2	-1	4	A(-1, 4)
2	2	0	2	A(0, 2)
2	2	1	0	<b>A(1, 0)</b>
2	2	2	-2	A(2, -2)
2	2	3	-4	A(3, -4)

Ao final do processo de votação, observa-se na matriz acumuladora que a célula por mais vezes incrementada é referenciada por  $a = 1$  e  $b = 0$ . Substituindo esses valores na equação *slope-intercept*, ou seja,  $y = 1.x + 0$ , significa que a equação que melhor define um segmento de reta que passa pelos pontos  $(-1, -1)$ ,  $(0, 0)$ ,  $(1, 1)$ ,  $(2, 2)$  na imagem analisada é  $y = x$ . A Figura 10 ilustra o espaço “real” dos parâmetros após o processo de votação aplicado na Figura 9.

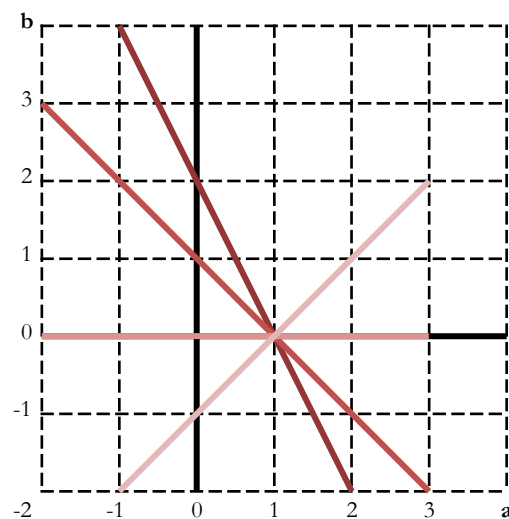


Figura 10: Resultado no espaço de parâmetros do processo de detecção de linhas

A Transformada de Hough não consegue determinar o início e o final do segmento de reta encontrado, dessa forma é necessário utilizar outra técnica de PDI para determinar onde começa e termina a reta. A Figura 11 ilustra através da linha tracejada, o local estimado pela Transformada de Hough do segmento de reta detectado no espaço da imagem.

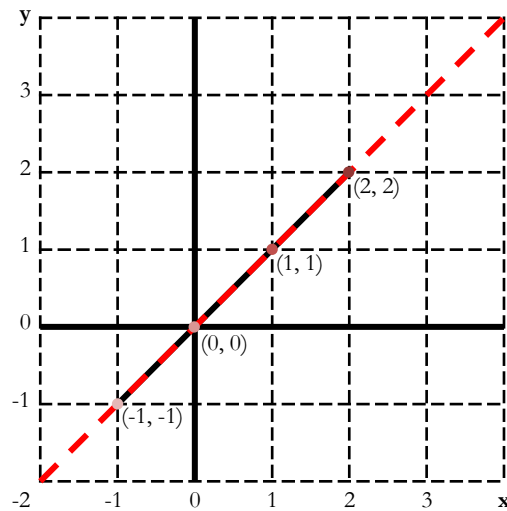


Figura 11: Segmento de reta detectado através da Transformada de Hough

O problema ao se utilizar a equação da reta na forma de *slope-intercept* é que tanto a inclinação quanto a intersecção são ilimitados quando a reta a ser detectada está na posição vertical. Segundo Gonzales e Woods [13] e Duda e Hart [10] isso ocorre, pois a equação  $y = a.x + b$ , quando usada em uma reta vertical, os valores dos parâmetros tende ao infinito. Por exemplo, esse problema pode ser observado na Figura 12 quando a Transformada de Hough é aplicada.

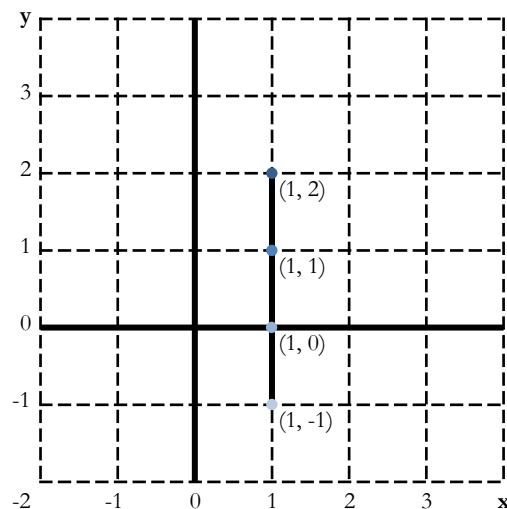


Figura 12: Linha vertical no espaço da imagem

A Tabela 2 mostra os valores das células quando executamos o processo de detecção de linhas usando a equação *slope-intercept*, para cada um dos quatro pontos do segmento de reta da Figura 12. A Figura 13 mostra que ao final do processo de votação, não existe, no espaço de parâmetros, nenhum ponto de intersecção relacionado aos valores mais prováveis dos parâmetros correspondentes ao segmento de reta existente na imagem analisada.

Tabela 2: Cálculo dos parâmetros (a) e (b) no processo de detecção de linhas verticais com a equação *slope-intercept*

x	y	a	b=-a.x+y	A(a, b)
1	-1	-2	1	A(-2, 1)
1	-1	-1	0	A(-1, 0)
1	-1	0	-1	A(0, -1)
1	-1	1	-2	A(1, -2)
1	-1	2	-3	A(2, -3)
1	-1	3	-4	A(3, -4)
1	0	-2	2	A(-2, 2)
1	0	-1	1	A(-1, 1)
1	0	0	0	A(0, 0)
1	0	1	-1	A(1, -1)
1	0	2	-2	A(2, -2)
1	0	3	-3	A(3, -3)
1	1	-2	3	A(-2, 3)
1	1	-1	2	A(-1, 2)
1	1	0	1	A(0, 1)
1	1	1	0	A(1, 0)
1	1	2	-1	A(2, -1)
1	1	3	-2	A(3, -2)
1	2	-2	4	A(-2, 4)
1	2	-1	3	A(-1, 3)
1	2	0	2	A(0, 2)
1	2	1	1	A(1, 1)
1	2	2	0	A(2, 0)
1	2	3	-1	A(3, -1)

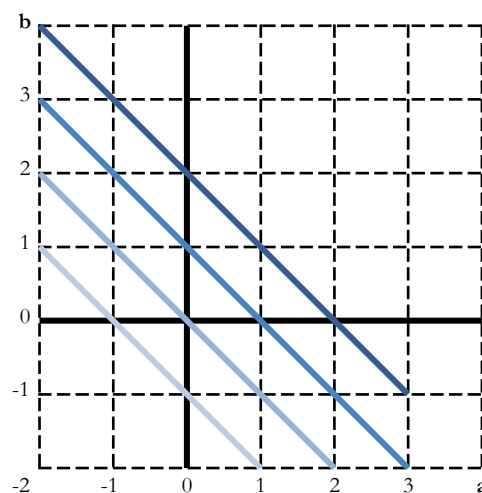


Figura 13: Resultado no espaço de parâmetros do processo de detecção de linhas verticais

Para solucionar o problema de detecção de linhas verticais com a técnica de Hough [15], Duda e Hart [10] sugerem uma parametrização alternativa, através da representação normal de uma linha utilizando coordenadas polares. As linhas são parametrizadas utilizando o

comprimento  $\rho$  e a orientação  $\theta$  das suas normais, conforme a Figura 14. Dessa forma, todos os pontos  $(x_i, y_i)$  da linha satisfazem a equação  $\rho = x \cdot \cos \theta + y \cdot \sin \theta$ . Se o valor de  $\theta$  for restrito ao intervalo de  $\theta[0, \Pi]$ , os parâmetros normais para uma linha são únicos. Dessa forma, cada linha no espaço da imagem corresponde a um ponto no espaço de parâmetros.

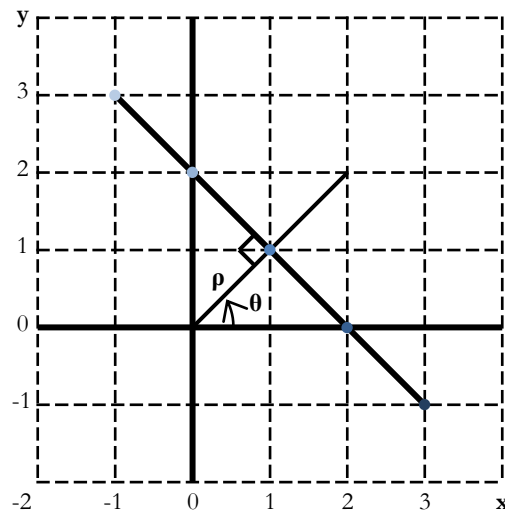


Figura 14: Representação de uma linha utilizando coordenadas polares

O algoritmo descrito por Duda e Hart [10] considera um acumulador bidimensional ( $A$ ) correspondendo dessa vez aos valores quantizados de  $\rho$  e  $\theta$ . Cada ponto no espaço da imagem produz uma linha senoidal no espaço de parâmetros. As curvas no espaço de parâmetros correspondentes a pontos de uma mesma linha no espaço da imagem têm um ponto de intersecção em comum. Um exemplo do processo de detecção de linhas utilizando o algoritmo de Duda e Hart [10] é ilustrado na Figura 15. Esse é o resultado gerado no espaço de parâmetros ao aplicar esse algoritmo sob o segmento de reta ilustrado na Figura 9.

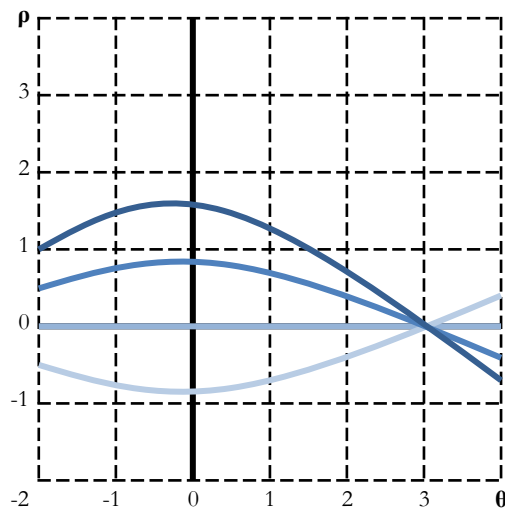


Figura 15: Resultado no espaço de parâmetros utilizando coordenada polar

As propriedades da parametrização proposta por Duda e Hart [10] são sumarizadas nos quatro itens a seguir:

- ✓ **1º item:** um ponto no espaço da imagem corresponde a uma curva senoidal no espaço de parâmetros;
- ✓ **2º item:** um ponto no espaço de parâmetros corresponde a uma linha reta no espaço da imagem;
- ✓ **3º item:** pontos sobre a mesma linha reta no espaço da imagem correspondem a curvas intercedendo um ponto em comum no espaço de parâmetros;
- ✓ **4º item:** pontos sobre a mesma curva no espaço de parâmetros correspondem a curvas intercedendo o mesmo ponto no espaço da imagem.

A representação na matriz acumuladora torna-se complexa quando a imagem analisada contiver muitos segmentos de retas. De acordo com Jamundá [17], para resolver esse problema deve-se iterativamente descobrir a maior linha no espaço da imagem, remover sua contribuição na matriz acumuladora e repetir o processo para achar sucessivamente as maiores linhas restantes.

### 3.2.2 – Detecção de Círculos

O processo de detecção de circunferências em imagens digitais é similar ao processo de detecção de linhas retas. A diferença básica entre as duas técnicas são as formas paramétricas para representação de um círculo e de uma linha. O principal problema do processo de detecção de círculos consiste em determinar quais pontos no espaço da imagem pertencem a uma mesma circunferência de raio  $r$ .

Esse problema pode ser solucionado utilizando um espaço de parâmetros tridimensional para incrementar valores dos parâmetros  $(a, b)$  que representam possíveis pontos centrais das circunferências na imagem. A Figura 16 ilustra os três parâmetros de um círculo, onde  $(a, b)$  especifica a sua localização no espaço bidimensional e  $(r)$  o tamanho do seu raio.

A equação geral para calcular raios de circunferências é definida por  $r = \sqrt{(x - a)^2 + (y - b)^2}$ , porém geralmente essa equação não é utilizada. A parametrização da equação geral de circunferência em função de  $a$  e  $b$  não produz resultados com alta performance computacional. Uma alternativa é a utilização dos parâmetros  $a$  e  $b$  em coordenadas polares, definidas por duas equações:  $a = x - r \cdot \cos(\theta)$  e  $b = y - r \cdot \sin(\theta)$ .

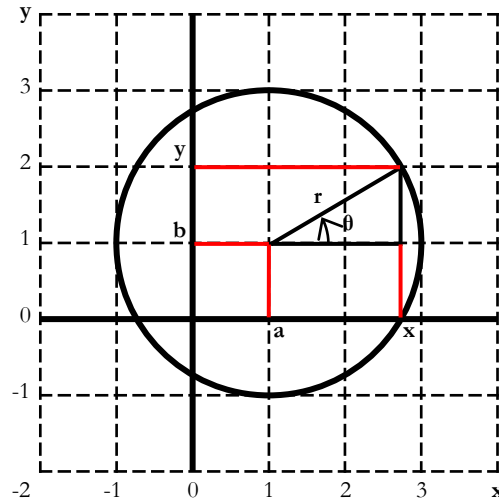


Figura 16: Circunferência de raio ( $r$ ) e centro ( $a, b$ )

### 3.3 – Aprimoramentos da Transformada de Hough Para Detecção de Círculos

Desde a criação da Transformada de Hough em 1962, surgiram várias parametrizações alternativas com objetivo de melhorar o desempenho computacional da Transformada Clássica de Hough e detectar outras formas geométricas além de retas. Atualmente a maioria das pesquisas envolvendo a TH utiliza a parametrização criada por Duda e Hart [10] em 1972.

#### 3.3.1 – Parametrização do algoritmo de Duda e Hart

O algoritmo de detecção de círculos criado por Duda e Hart [10] usa uma matriz acumuladora tridimensional para indexar os parâmetros que especificam a localização  $a = (a_i, a_j)$  e o raio  $r$  do círculo. Percorre o espaço de parâmetros através do ângulo  $\theta$  num intervalo de  $[0, 2\pi]$ , incrementando possíveis centros de círculos existentes na imagem analisada. O incremento  $\Delta\theta$  baseia-se no tamanho do raio que se deseja encontrar, variando aproximadamente em  $1/r$ . A Figura 17 ilustra o processo de incremento dos possíveis centros de círculos na imagem com a parametrização descrita por Duda e Hart [10].

Seja  $p(x)$  a imagem a ser analisada e já previamente pré-processada com um filtro de detecção de bordas e  $x$  um ponto na imagem  $p(x)$  sob as coordenadas  $(x_1, x_2)$ . Para cada ponto  $x$ , existe um conjunto  $C_x$  de círculos que interceptam  $x$ . Seja  $X_p$  o conjunto de pontos  $\{x \mid p(x) \neq 0\}$ . A parametrização sugerida por Duda e Hart [10] procura o centro  $a(x)$  e o raio  $r(x)$  para cada membro do conjunto  $C_x$ , tal que  $x \in X_p$ . A representação paramétrica utilizada por esse algoritmo é a equação geral de círculos.

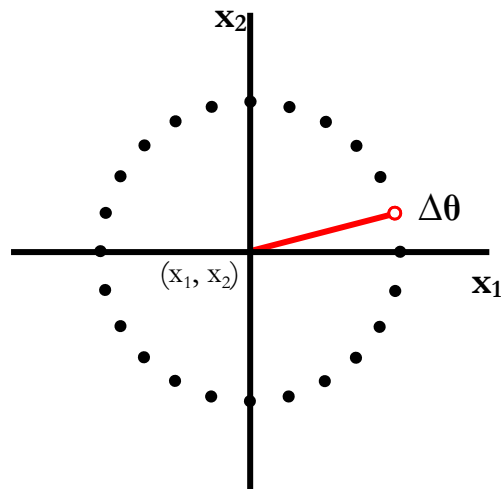


Figura 17: Incremento de possíveis centros de círculos

O conjunto resultante dos centros de raios detectados na imagem  $p(x)$  é denotado por  $\{(a, r) \mid x \in X_p\}$ . Para cada membro de  $\{(a, r) \mid x \in X_p\}$ , uma célula acumuladora em  $(a(x), r(x))$  no espaço de parâmetros é incrementada em um. Depois que todos os membros do conjunto  $X_p$  forem processados, a matriz acumuladora em  $(a(x), r(x))$  conterá o número de elementos de  $X_p$  que estão sobre o círculo com centro  $a(x)$  e o raio  $r(x)$ .

A Figura 18 ilustra o processo de detecção de círculos de Duda e Hart [10], onde seis círculos pertencentes ao conjunto  $C_x$  interceptam o ponto  $(x_i, y_i)$ . Ao realizar o processo de incremento, os centros dos seis círculos da Figura 18 serão incrementados em um. Para cada ponto processado pertencente à mesma circunferência, será incrementado em um o mesmo ponto central do círculo. Quando todos os pontos da imagem forem processados, os centros dos seis círculos irão apresentar as máximas locais da matriz acumuladora.

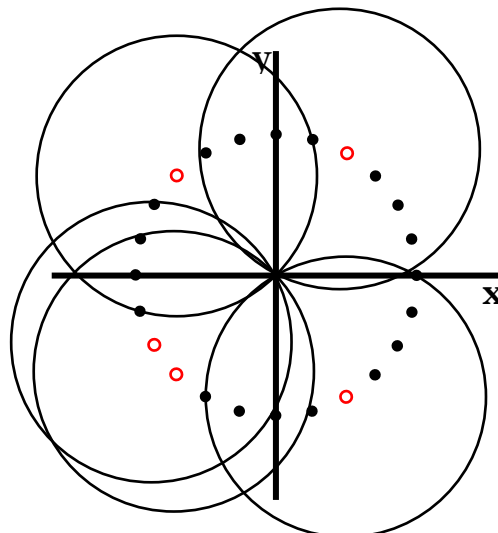


Figura 18: Processo de detecção de círculos do algoritmo de Duda e Hart [10]

### 3.3.2 – Parametrização do algoritmo de Kimme, Ballard e Sklansky

O algoritmo de detecção de círculos sugerido por Kimme, Ballard e Sklansky [18] é uma extensão do algoritmo de Hough [15] e um aprimoramento do algoritmo de Duda e Hart [10]. Baseia-se na idéia de O’Gorman e Clowes [27] na qual o ângulo  $\theta$ , usado como entrada de valores no espaço de parâmetros, é obtido a partir da direção do gradiente de um contorno. É um algoritmo de detecção de círculos computacionalmente mais eficiente que o de Duda e Hart [10], pois realiza um procedimento para gerar os membros digitalizados de  $C_x$  e usa a direção do gradiente para eliminar porções de cada membro do conjunto  $C_x$ .

Para gerar os membros do conjunto  $C_x$ , é calculada a digitalização de  $(x_i, y_i)$  para um raio fixo  $r$ , sendo  $x_i = r \cdot \cos(\theta)$  e  $y_i = r \cdot \sin(\theta)$ . Enquanto isso o valor de  $\theta$  cresce em incrementos de  $\Delta\theta$ , onde  $\Delta\theta$  varia em aproximadamente  $1/r$ . Da mesma forma que a parametrização de Duda e Hart [10], esse algoritmo gera o conjunto dos possíveis centros de círculos que interceptam o mesmo ponto  $x$ .

A diferença básica entre os dois algoritmos é que ao invés de guardar no espaço de parâmetros as informações sobre todos os possíveis centros de círculos que interceptam  $x$ , utiliza-se a direção do gradiente do ponto  $x$  para selecionar apenas uma porção do conjunto  $C_x$  que provavelmente são centros de círculos.

Seja  $f(x)$  o nível de cinza do ponto  $x$  analisado,  $g(x)$  a digitalização do gradiente de  $f(x)$  e  $\phi(x)$  a direção de  $g(x)$ . Para cada ponto analisado na imagem, realiza-se o processo de detecção de bordas para aquisição dos componentes horizontais  $g_1(x)$  e verticais  $g_2(x)$  do gradiente  $g(x)$ . O primeiro passo é obter os valores dos componentes  $g_1(x)$  e  $g_2(x)$ , através das equações  $g_1(x) = \frac{1}{2s} [f(x + (s, 0)) - f(x - (s, 0))]$  e  $g_2(x) = \frac{1}{2s} [f(x + (0, s)) - f(x - (0, s))]$ , onde  $s$  é igual a um número inteiro positivo.

O gradiente  $g(x)$  é calculado por  $\|g(x)\| = \sqrt{(g_1)^2 + (g_2)^2}$  e sua direção é obtida através da equação de arco-tangente  $\phi(x) = \tan^{-1} \left( \frac{g_2(x)}{g_1(x)} \right)$ . A partir do resultado de  $\phi(x)$ , seleciona-se apenas a porção de cada membro do conjunto  $C_x$ , tal que o vetor  $g(x)$  aponta para  $\{(a, r) \mid x \in X_p\}$ . Dessa forma, utilizando a representação paramétrica  $a = x - r \cdot \cos(\theta)$  e  $b = y - r \cdot \sin(\theta)$ , incrementa-se em um a célula  $(a, b, r)$  na matriz acumuladora.

A Figura 19 mostra que os pontos centrais incrementados são os pontos definidos pelas direções do gradiente  $g(x)$ . Na maioria das vezes, o centro da circunferência encontra-se na direção determinada pelo gradiente  $g(x)$ . Por eliminar um número considerável de incrementos a

cada ponto analisado, a parametrização de Kimme, Ballard e Sklansky [18] tem um desempenho computacional superior em comparação com a parametrização de Duda e Hart [10].

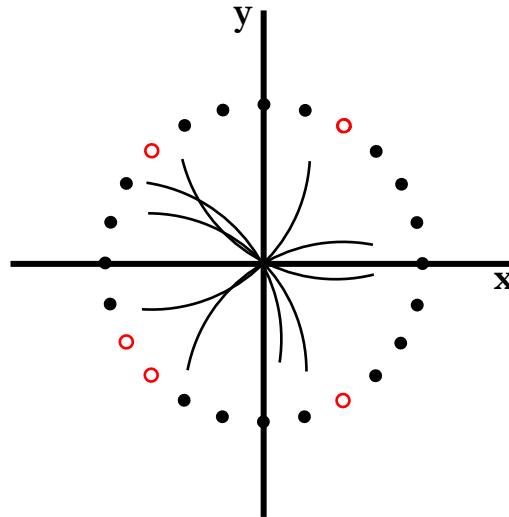


Figura 19: Processo de detecção de círculos do algoritmo de Kimme, Ballard e Sklansky [18]

### 3.4 – Variações da Transformada de Hough

A Transformada Clássica de Hough é uma das maneiras mais elegantes e eficazes de extrair formas primitivas de imagens digitais. Porém, as exigências computacionais e o uso demasiado da memória muitas vezes impedem que a Transformada de Hough seja usada em tarefas básicas de Processamento Digital de Imagens e Visão Computacional. Vários algoritmos baseados na Transformada de Hough foram criados com objetivo de diminuir esses esforços e tornar possível a extração de formas abstratas em imagens digitais.

#### 3.4.1 – Transformada Adaptativa de Hough

Illingworth e Kittler [16] criaram em 1986 a Transformada Adaptativa de Hough (TAH), que consiste em calcular a Transformada de Hough em um acumulador de menor tamanho e usar essa informação de modo inteligente para redefinir os parâmetros a partir da investigação de determinadas áreas de interesse com um maior nível de detalhamento. Nesse processo, a região do espaço de parâmetros onde o pico do acumulador irá se apresentar não é conhecida, precisando calcular a Transformada Adaptativa de Hough um determinado número de vezes até achar uma solução.

O primeiro passo é criar uma matriz acumuladora com um número pequeno de células, produzindo uma eficiência computacional significativa. O objetivo é procurar o pico no

acumulador verificando se ele satisfaz uma condição pré-determinada de precisão. Caso essa condição não seja satisfatória, uma nova matriz acumuladora é criada, recursivamente, limitada pelas coordenadas da célula onde o último pico foi encontrado. Caso contrário, os parâmetros do objeto são identificados. A Figura 20 ilustra um fluxograma com uma formalização da Transformada Adaptativa de Hough.

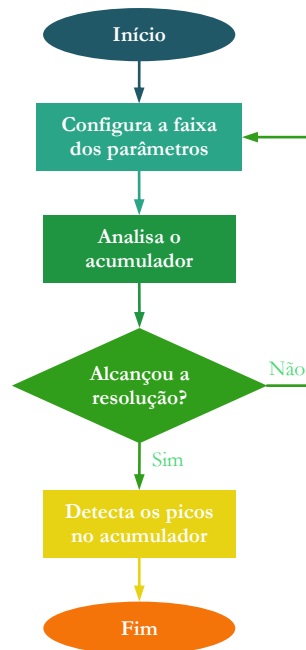


Figura 20: Fluxograma teórico da Transformada Adaptativa de Hough

A primeira decisão que deve ser tomada ao utilizar a Transformada Adaptativa de Hough é definir o tamanho inicial da matriz acumuladora. Inicialmente esse tamanho pode ser definido em uma matriz de tamanho  $2 \times 1$  ou  $2 \times 2$ , porém são necessárias muitas amostras do espaço de parâmetros para que a Transformada Adaptativa de Hough possa ser interpretada confiantemente. De acordo com Illingworth e Kittler [16], uma matriz acumuladora de  $9 \times 9$  é de bom tamanho para contemplar essas considerações.

O processo de recursão sobre as células onde se apresentam os picos das matrizes analisadas aumenta a precisão da busca a cada nível de recursão. Um dos aspectos mais importantes desse algoritmo é a sua habilidade de redefinir os parâmetros independentemente do nível atual de recursão. Imagens que contém múltiplas ocorrências da mesma forma a ser detectada serão processadas até que a matriz acumuladora não contenha nenhuma estrutura significativa para representar os objetos no espaço de parâmetros. Desse modo, a quantidade de informação processada nas matrizes da Transformada Adaptativa de Hough é bem menor que a processada pela Transformada Clássica de Hough.

### 3.4.2 – Transformada Combinatorial de Hough

A Transformada Combinatorial de Hough (TCBH) foi criada por Ben-Tzvi e Sandler [3] em 1989 como o meio mais eficiente de calcular a Transformada de Hough. A idéia desse algoritmo é usar a informação definida por um conjunto de características de pontos na imagem ao invés de calcular  $\rho$  sob os valores de  $\theta$ . Cada dois pontos na imagem indicam uma possível colinearidade. Os parâmetros que descrevem a linha que une os dois pontos analisados são descritos por  $(\rho_0, \theta_0)$ , esses parâmetros podem ser derivados da equação paramétrica da reta em coordenadas polares  $\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta)$ .

Para cada combinação de dois pontos na imagem, computam-se os parâmetros  $(\rho_0, \theta_0)$  que representa a linha que une esses pontos. Dado o par de pontos  $(x_1, y_1)$  e  $(x_2, y_2)$  têm-se as seguintes equações:  $\rho_0 = x_1 \cdot \cos(\theta_0) + y_1 \cdot \sin(\theta_0)$  e  $\rho_0 = x_2 \cdot \cos(\theta_0) + y_2 \cdot \sin(\theta_0)$ . Subtraindo as duas equações, temos como resultado  $\theta_0 = \arctan\left(\frac{\Delta x}{\Delta y}\right)$ . O parâmetro  $\theta_0$  pode ser usado para determinar o  $\rho_0$  através da substituição da equação paramétrica. Se três ou mais pontos são colineares, as suas respectivas senóides interceptam um ponto em comum, sendo que todos os pares de intersecção têm um mesmo valor em comum de  $(\rho, \theta)$ .

Para determinar se qualquer parte de um conjunto de  $m$  pontos são colineares, todas as possíveis combinações de segmentos de reta existentes entre dois pontos precisam ser calculadas. Cada cálculo dos pares processados contribui no incremento em um na célula  $(\rho, \theta)$  da matriz acumuladora. Nesse processo, o número de ciclos de computação é de  $c_m^2 = \frac{m!}{2!(m-2)!}$  crescendo nitidamente com o aumento de  $m$ , o número de pontos processados na imagem. Se o número de pontos processados for relativamente grande, o número de computações da Transformada Combinatorial de Hough será maior em comparação com a Transformada Clássica de Hough.

### 3.4.3 – Transformada Generalizada de Hough

A Transformada Generalizada de Hough (TGH) é usada quando a forma a ser detectada na imagem não possui uma equação analítica simples para representá-la. A idéia central dessa parametrização é que, ao invés de usar a equação matemática da curva, usa-se uma tabela paramétrica ( $R$ -table) para definir uma relação entre os limites e a orientação da curva com os parâmetros de Hough. Durante a fase treinamento, a tabela paramétrica é preenchida com os valores representativos de um protótipo usado no processo de detecção da forma na imagem.

O primeiro estudo apresentado sobre a Transformada Generalizada de Hough foi

realizado em 1981, por Ballard [1]. Seu algoritmo supõe que a forma e a orientação do objeto a ser detectado na imagem são conhecidas, como o exemplo ilustrado na Figura 21. O início do processo se dá pela escolha de um ponto de referência arbitrário  $(x_{ref}, y_{ref})$  no objeto. Após essa etapa é definida a forma arbitrária em termos de distância e ângulo, da borda do objeto para o ponto de referência escolhido.

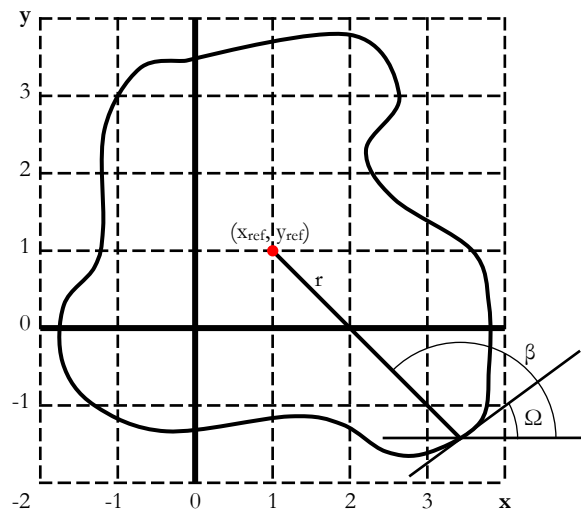


Figura 21: Geometria utilizada na Transformada Generalizada de Hough

Ao analisar cada ponto pertencente à borda do objeto, um segmento é traçado até o ponto de referência  $(x_{ref}, y_{ref})$  calculando a orientação  $\Omega_i$  e preenchendo tabela paramétrica  $R$ -table com os valores de direção e distância do ponto analisado para o ponto de referência. Nesse caso o valor de  $\Omega_i$  é usado como índice na tabela paramétrica. É necessário fazer a operação para mais de um par de direções e distâncias se levado em consideração que provavelmente haverá mais de uma ocorrência da orientação  $\Omega_i$  ao percorrer toda a borda do objeto.

O espaço de parâmetros usado na Transformada Generalizada de Hough corresponde aos valores quantizados de  $x_{ref}$  e  $y_{ref}$ , que define as possíveis posições da forma arbitrária na imagem. A transformada é executada sobre a imagem no ponto  $(x_{ref}, y_{ref})$ , na distância  $r$  e no ângulo  $\beta$ , definido por duas equações:  $x_{ref} = x + r \cos \beta$  e  $y_{ref} = y + r \sin \beta$ . Os valores de  $r$  e  $\beta$  são derivados da  $R$ -table, usando a orientação  $\Omega$  como índice da tabela paramétrica, enquanto são lidos todos os pares  $(r, \beta)$ . Seguindo as regras do processo de votação, a célula  $(x_{ref}, y_{ref})$  na matriz acumuladora é incrementada em um.

Até aqui foi assumido que a orientação da forma abstrata na imagem era conhecida. No caso em que a orientação não é conhecida a matriz acumuladora irá incorporar mais um parâmetro indexador, o  $\phi$  utilizado para indexar os valores da orientação. Desse modo, a matriz

acumuladora tridimensional é indexada por  $(\phi, x_{ref}, y_{ref})$  cujos valores são calculados a partir das seguintes equações:  $x_{ref} = x + r \cos(\beta + \phi)$  e  $y_{ref} = y + r \sin(\beta + \phi)$ .

As vantagens atribuídas a Transformada Generalizada de Hough são: a robustez na detecção de formas parciais ou ligeiramente deformadas; o reconhecimento com a presença de estruturas adicionais na imagem (retas, círculos, elipses, etc.) e a possibilidade de encontrar múltiplas ocorrências da mesma forma arbitrária durante uma única fase de detecção. Porém, a sua principal desvantagem é o uso extensivo de memória e processamento.

### 3.4.4 – Transformada Hierárquica de Hough

A Transformada Hierárquica de Hough (THH) foi proposta por Princen *et al.* [33] em 1989, combinando uma estrutura piramidal com a Transformada Clássica de Hough. Essa estrutura piramidal é baseada em um modelo de camadas que divide a imagem em sub-imagens. Normalmente a pirâmide é constituída com  $P + 1$  níveis, sendo que a cada nível  $l$  a imagem é dividida em  $2^l - 1 \times 2^l - 1$  sub-imagens. O topo da pirâmide  $P$  representa a imagem original e no nível inferior segmentos de retas são detectados através da Transformada de Hough aplicada em uma pequena sub- imagem.

O problema em diferenciar um segmento de reta “real” e pontos casualmente alinhados é reduzido através da análise de sub-imagens. O método mostra-se relativamente eficiente, utilizando apenas um pequeno vetor acumulador para representar todas as possíveis linhas que passam sob uma sub-imagem. Cada sub-imagem contém no máximo duas linhas e seus resultados são distribuídos através da pirâmide. A cada nível acima da pirâmide, uma representação global dos segmentos de reta é obtida formando agrupamento de linhas da sub-imagem anterior.

No nível processado, uma matriz de tamanho  $4 \times 4$  é transformada em uma sub-imagem na qual inclui uma vizinhança central de  $2 \times 2$  e os seus vizinhos mais próximos, como ilustrado na Figura 22. Algumas linhas da imagem são unidas para serem propagadas ao próximo nível da hierarquia, repetindo esse processo até que no nível mais alto da pirâmide exista apenas um nó que descreva a imagem por completo. Por causa de efeitos como leves curvaturas na linha a ser detectada, deve haver certa tolerância no processo de união de determinadas linhas na imagem.

Princen *et al.* [33] notaram que no processo de união de linhas na imagem, uma linha que contém um leve encurvamento pode ser classificada como duas linhas sobrepostas. Não é necessário que as linhas sejam exatamente colineares, realiza-se apenas o calculo da média para representar os valores das linhas que serão unidas pela Transformada Hierárquica de Hough.

Quanto mais alto o nível da pirâmide menor a possibilidade de erros de colinearidade, tornando mais precisas as estimativas dos parâmetros da linha processada.

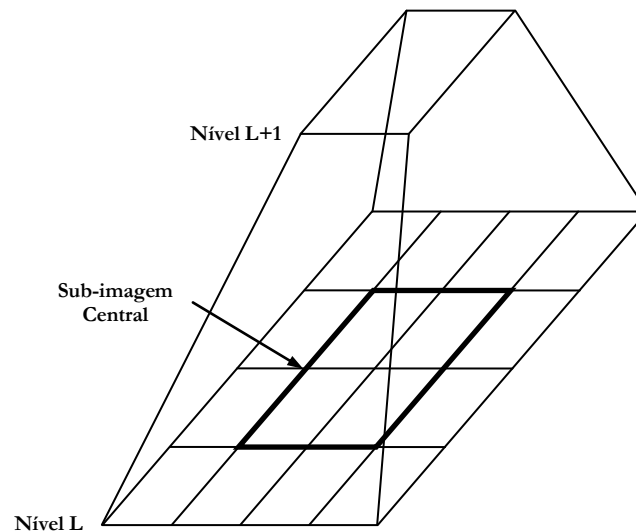


Figura 22: Pirâmide da Transformada Hierárquica de Hough

### 3.4.5 – Transformada Probabilística de Hough

A Transformada Probabilística de Hough (TPH) foi criada em 1991 pelos pesquisadores Kiryati, Eldar e Bruckstein [19]. Com esse algoritmo é possível detectar formas geométricas em imagens digitais computando apenas uma porção dos pontos da imagem analisada. Kiryati, Eldar e Bruckstein [19] mostram que é possível detectar formas geométricas em imagens computando a Transformada de Hough em uma proporção de ( $0% < \alpha < 100%$ ) de pontos na imagem, conduzindo dessa forma a uma economia computacional significativa.

Os pontos processados são escolhidos randomicamente de acordo com uma função de densidade de probabilidade uniforme, definida sob o tamanho da imagem analisada. Para reduzir a taxa de erros é necessário pré-ajustar o número de pontos escolhidos a um valor que seja pouco maior do que o necessário para as circunstâncias médias. A amostra dos pontos processados pode ser definida através de um valor fixo para todas as imagens analisadas ou ainda utilizar uma técnica que defina automaticamente esse valor.

A pesquisa de Ylä-Jääski e Kiryati [44] mostra um método de ajuste adaptável para o número de pontos escolhidos randomicamente para a Transformada Probabilística de Hough. Em muitos casos o número de objetos a serem detectados é desconhecido, e a detecção de objetos em uma imagem ruidosa pode dificultar o processo. O algoritmo desenvolvido por Ylä-Jääski e Kiryati [44] realiza uma parada adaptável que termina o processo de incrementos na

matriz acumuladora assim que o número de objetos a ser detectado parecer confiável.

Nos experimentos de Ylä-Jääski e Kiryati [44] foram usados os algoritmos de detecção de círculos com informações sobre o tamanho do raio a ser detectado, além do processo de direção do gradiente do ponto analisado. A pesquisa conclui que, com a análise de cerca de 15% dos pontos da imagem é possível detectar formas geométricas em imagens digitais com taxa de erro igual à zero e conduzindo a uma significativa economia computacional.

### 3.4.6 – Transformada Randômica de Hough

Kultanen, Xu e Oja [20] criaram a Transformada Randômica de Hough (TRH) para realizar o cálculo da Transformada de Hough a partir um conjunto de pontos randomicamente escolhidos na imagem analisada. Através desse conjunto é realizado o mapeamento da forma geométrica a ser detectada, do espaço da imagem para o espaço de parâmetros. O tamanho do conjunto de pontos depende exclusivamente da complexidade da forma geométrica. No caso de detecção de segmentos de retas, a célula acumuladora no espaço de parâmetros pode ser obtida a partir de um par de pontos pertencentes à borda da imagem analisada.

Os pares de pontos  $(p_i, p_j)$  são escolhidos randomicamente e os parâmetros da linha que passam através desses pontos são computados. O resultado é registrado no espaço de parâmetros incrementando em um a célula  $A(p_i, p_j)$ . Esse procedimento é repetido um número pré-determinado de vezes, sendo bem menor que a quantidade total de pares de pontos da imagem. Ao final do processo, a Transformada Randômica de Hough é executada para descobrir os valores no espaço de parâmetros que são superiores a um determinado limiar.

A parametrização para detecção de círculos é mais complexa que a de detecção de retas. Diferentemente da detecção de retas, dois pontos são insuficientes para detectar círculos com o cálculo da Transformada Randômica de Hough. Sendo assim, três ou quatro pontos é o mínimo necessário para realizar a busca por círculos na imagem analisada. A diferença na detecção de círculos entre a Transformada Clássica de Hough e a Transformada Randômica de Hough é que apenas um pequeno conjunto de pontos da borda do círculo é usado para o mapeamento do espaço de parâmetros e a Transformada Randômica de Hough não usa cálculos como a arco-tangente ou da direção do gradiente.

Falcon [11] apresenta um sistema de detecção de objetos circulares em imagens digitais através da Transformada Randômica de Hough. O algoritmo de Falcon [11] seleciona pares de pontos randomicamente na imagem e traça segmentos entre cada dois pares selecionados. Em

cada segmento traçado, o algoritmo encontra o ponto médio do segmento e traça uma reta perpendicular a partir de cada segmento. As retas perpendiculares interceptam o mesmo ponto central do círculo que representa a parametrização da máxima local na matriz acumuladora. O algoritmo desenvolvido por Falcon [11] é ilustrado na Figura 23.

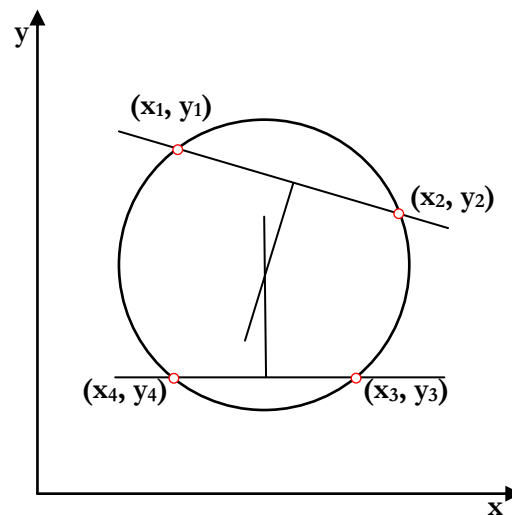


Figura 23: Processo de detecção de círculos com a Transformada Randômica de Hough

### 3.5 – Considerações Finais

Neste capítulo foram abordados os conceitos básicos da Transformada de Hough, as principais parametrizações do processo de detecção de circunferências, bem como as variações existentes como alternativa para um melhor desempenho computacional da transformada.

Decidimos por um estudo sobre a viabilidade do uso da Transformada de Hough no processo de detecção da íris para o desenvolvimento do sistema do Dispositivo Rastreador. O estudo foi restrito aos algoritmos de detecção de círculos parametrizados por Duda e Hart [10] e Kimme, Ballard e Sklansky [18], e uma extensão desses algoritmos adicionando aos dois casos a Transformada Probabilística de Hough [19][44].

O próximo capítulo apresenta as idéias que surgiram durante o desenvolvimento desse trabalho, com o objetivo de melhorar a funcionalidade do Dispositivo Rastreador criado e, também, de auxiliar Portadores de Necessidades Especiais com um conjunto de técnicas de acessibilidade implementadas no sistema do rastreador.

---

# Mouse Óptico Ocular

---

**M**ouse Óptico Ocular é o nome dado ao Dispositivo Rastreador desenvolvido durante essa dissertação. O dispositivo foi batizado como esse nome por prover um meio alternativo de interação de usuários com sistemas computacionais utilizando apenas os movimentos oculares. Apesar do nome ser específico ao dispositivo de entrada “mouse”, ele pode, de forma análoga, atuar como outros dispositivos de entrada computacional, por exemplo, o teclado e leitores biométricos.

Com os movimentos oculares é possível simular os movimentos do cursor do mouse. Ao invés do usuário utilizar um mouse convencional para mover o cursor com movimentos motores, o cursor do mouse será posicionado na tela com técnicas de Rastreamento Ocular. O posicionamento do cursor do mouse é calculado a partir de uma estimativa da direção do olhar sobre a tela. Esse mapeamento é descrito numa forma paramétrica e calibrado através do Método dos Mínimos Quadrados.

O ideal é utilizar os movimentos oculares para permitir que o usuário efetue tarefas rotineiras com o mouse e também com o teclado. Além de apontar para objetos visuais do Sistema Operacional, o Rastreamento Ocular permite que o usuário escreva através de uma comunicação visual. Para possibilitar a interação do usuário com o computador através dos movimentos dos olhos, além do movimento do cursor desenvolvemos sistemas que permitem a emulação do clique dos botões do mouse, a resposta de ações do sistema através da observação de regiões sensíveis e a escrita de palavras com o mapeamento dos movimentos oculares.

No Capítulo 2, item 2.4, identificamos que é possível recalibrar automaticamente o Dispositivo Rastreador durante o seu uso. Porém, nessa dissertação implementamos apenas o processo inicial de calibração para o nosso Dispositivo Rastreador.

## 4.1 – Hardware do Mouse Óptico Ocular

O ponto chave no início da montagem do dispositivo físico foi a escolha por um Rastreador Fixo na Mesa ou um Rastreador Fixo na Cabeça. Durante o uso do Rastreador Fixo na Cabeça, o usuário deve permanecer com a cabeça imóvel para não perder a precisão obtida na etapa de calibração. Como o foco do nosso dispositivo são usuários Portadores de Necessidades Especiais, o Rastreador Fixo na Cabeça atende as principais necessidades e requisitos de uso.

Portadores de Tetraplegia além da paralisia dos membros inferiores e superiores podem sofrer de distúrbios na musculatura do tronco e nos movimentos do pescoço. Esses usuários irão utilizar o Rastreador Fixo na Cabeça sem a necessidade de um detector de movimentos da cabeça em segundo plano. Para os demais usuários, o uso do Mouse Óptico Ocular será limitado a manter a cabeça imóvel durante o uso do dispositivo.

Do mesmo modo que o dispositivo criado por Souza *et al.* [42], o protótipo desenvolvido nessa dissertação foi adaptado em um óculos de proteção industrial, cujo material usado na armação é em acrílico. O protótipo é composto por uma micro-câmera de aquisição de vídeo colorido e áudio estéreo com transmissão sem fio (*wireless*), em conjunto com uma placa de aquisição de sinal de vídeo (*framegrabber*) e um microcomputador.

A vantagem do uso de transmissão *wireless* dos vídeos capturados é que o usuário utiliza o dispositivo da mesma forma que um óculos com lentes oftálmicas. Como a armação dos óculos é transparente, o usuário tem uma visão total da sua área de trabalho. Os vídeos podem ser transmitidos em um alcance máximo de 200 metros sem obstáculos, permitindo ao usuário utilizar o Dispositivo Rastreador em conjunto com um projetor multimídia. A micro-câmera possui uma lente de 3,5 mm e captura imagens no formato NTSC com resolução de 380 linhas.

As imagens são capturadas pelo receptor de vídeo que acompanha o kit da câmera *wireless*. Essas imagens são transmitidas através de uma saída RCA (*Radio Corporation of America*) para a placa de captura de vídeo. A *framegrabber* é a responsável pela digitalização das imagens que são processadas pelos algoritmos de detecção da circunferência da íris. Para o desenvolvimento do protótipo, foram adquiridas duas *framegrabbers*, uma com interface PCMCIA (utilizada em um computador do tipo *notebook*) e outra PCI (utilizada em computador do tipo *Desktop*).

Atualmente, os monitores computacionais são comercializados em formatos distintos de telas: normais (formato 4:3) e *widescreen* (formato 16:9). A escolha por se trabalhar com dois tipos de computadores foi em razão ao formato das telas de ambos. Na etapa de testes do protótipo, a

precisão do Rastreo Ocular e o processo de calibração nos dois formatos de tela foram avaliados.

#### 4.1.1 – Montagem do Dispositivo Rastreador

A micro-câmera foi incorporada aos óculos de proteção industrial de forma artesanal, como ilustrado na Figura 24. A viseira frontal direita foi perfurada para possibilitar a fixação da câmera nos óculos. Ela está estrategicamente posicionada na região frontal dos olhos para capturar diretamente as imagens da íris do usuário. Devido à localização da câmera, o usuário observa a área de trabalho um pouco ofuscada.

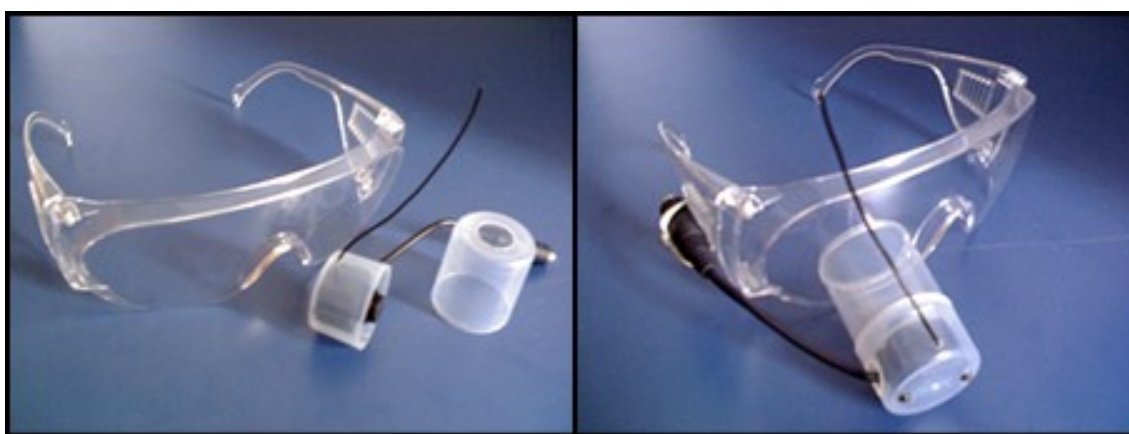


Figura 24: Montagem do protótipo do Dispositivo Rastreador (óculos e micro-câmera de captura de vídeo)

O receptor *wireless*, ilustrado na Figura 25, é o responsável pela captura do sinal emitido pela micro-câmera. O sinal é transmitido para a *Framegrabber* que trabalha como um integrador entre o Dispositivo Rastreador “físico” e o sistema rastreador “lógico”. O processamento das imagens digitalizadas pela *Framegrabber* é realizado através do modelo *Windows Driver Model* (WDM), onde as imagens são capturadas diretamente do *driver* da *Framegrabber* e armazenadas temporariamente na memória principal do computador.



Figura 25: Instalação do receptor de vídeo e áudio (*wireless*) com a placa de captura de vídeo (*framegrabber*)

Para cada quadro transmitido pela micro-câmera, é realizado o processamento da imagem diretamente na memória principal do computador. Após calcular e estimar o Ponto de Interesse, a imagem processada é liberada da memória. Como o sistema foi desenvolvido para ser executado em dois tipos diferentes de computadores, ele detecta automaticamente se o formato do monitor utilizado é *widescreen* ou normal. A Figura 26 ilustra a montagem do Dispositivo Rastreador em um computador do tipo *notebook*.

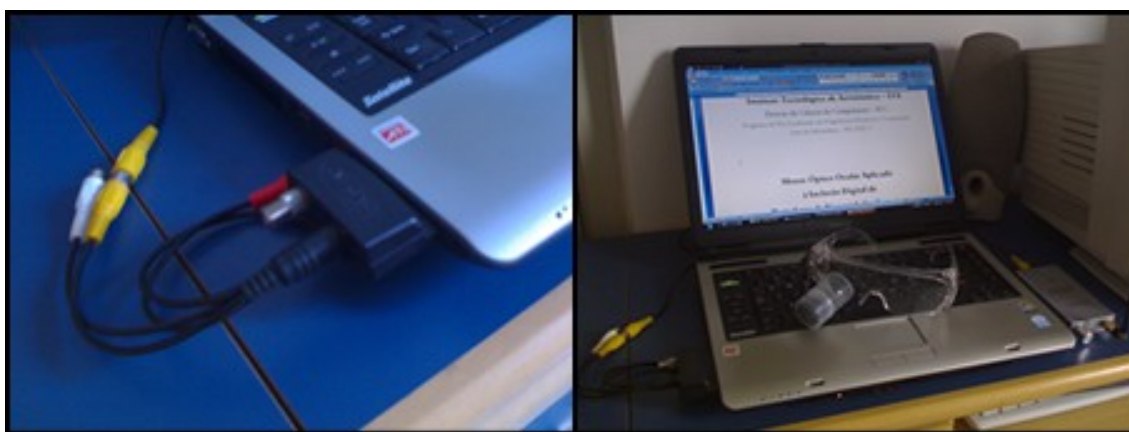


Figura 26: Dispositivo Rastreador instalado em um computador do tipo *notebook*

## 4.2 – Processo de Calibração

O procedimento de calibração segue o estudo desenvolvido por Ramanauskas [37], que define a técnica de Calibração de Segunda Ordem com 25 pontos como uma das mais eficazes na calibração de Dispositivos Rastreadores. O processo de calibração apresenta ao usuário um conjunto de pontos na tela para que ele observe enquanto é realizado um mapeamento dos pontos observados. A Figura 27 ilustra 25 pontos de calibração em um monitor no formato 4:3.

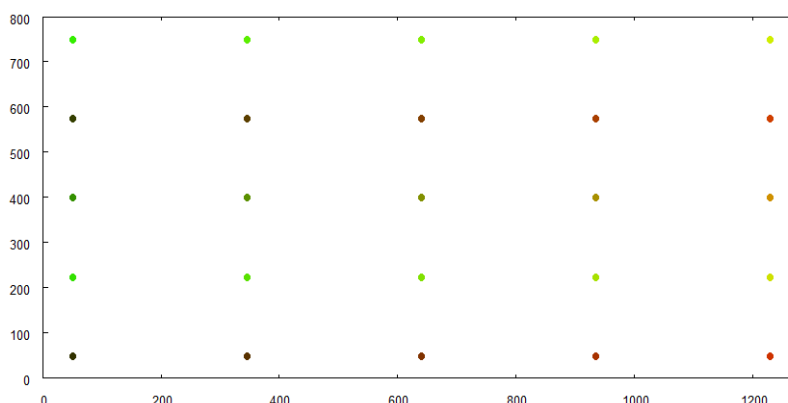


Figura 27: Exemplo de calibração em um monitor com resolução de 1280x800

Os 25 pontos são distribuídos uniformemente na tela, porém para realizar o

mapeamento apenas um ponto é mostrado por vez. A duração da visualização de cada ponto é de aproximadamente 2 segundos, tempo suficiente para estimar o Ponto de Interesse e mapear o ponto observado na tela. Para diminuir o estresse causado ao usuário, optamos por mostrar os pontos automaticamente de forma aleatória, através de uma função de distribuição de probabilidade uniforme.

A calibração é ajustada linearmente através do Método dos Mínimos Quadrados, uma técnica de otimização matemática que ajusta um conjunto de dados para minimizar a soma dos quadrados das distâncias entre uma curva e os pontos analisados. Para cada ponto observado no processo de calibração, é montado um conjunto de dados com informações sobre as coordenadas observadas pelo usuário. Esse conjunto de dados é utilizado para determinar as posições observadas pelo usuário no monitor durante o uso do Dispositivo Rastreador.

A Calibração de Segunda Ordem utiliza duas matrizes bases ( $t$  e  $v$ ) para criar uma matriz de transformação ( $T$ ) através do Método de Mínimos Quadrados. A matriz  $t$  armazena as coordenadas dos pontos que aparecem na tela do usuário durante o processo de calibração, e a matriz  $v$  indica a coordenada observada pelo usuário para cada elemento de  $t$ . Para cada imagem analisada, o sistema utiliza a matriz de transformação  $T$  e a coordenada central da circunferência da íris  $(x_i, y_i)$  para estimar a posição  $(x'_i, y'_i)$  observada pelo usuário no monitor, onde:

$$T = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_5 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_5 \end{bmatrix}, v_i = \begin{bmatrix} 1 \\ x_i \\ y_i \\ x_i y_i \\ x_i^2 \\ y_i^2 \end{bmatrix} \text{ e } v'_i = \begin{bmatrix} x'_i \\ y'_i \end{bmatrix}$$

tais que  $v'_i = T \cdot v_i$ . O método utilizado para calcular os valores dos coeficientes  $a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4$  e  $b_5$  é apresentado no Capítulo 5, no item 5.4.1.

### 4.3 – Emulação dos Botões do Mouse

O processo de emulação dos botões do mouse é uma das etapas mais difíceis no desenvolvimento de um Dispositivo Rastreador. Não é fácil achar um consenso nas pesquisas de Rastreo Ocular sobre a melhor alternativa para realizar o clique do mouse através dos olhos. Apesar do principal interesse dessa dissertação ser a etapa de Rastreo Ocular, é interessante implementar um modelo de emulação dos botões do mouse que permita ao usuário manipular os objetos visuais do Sistema Operacional durante os testes do nosso Dispositivo Rastreador.

O dispositivo desenvolvido por Kumar *et al.* [21] utiliza os botões do teclado para emular os cliques dos botões do mouse. Foram configuradas algumas teclas para executar as seguintes tarefas: clique simples e duplo, clique com o botão direito, efeito ao passar o mouse sobre um objeto e o clicar e arrastar objetos visuais. Ao visualizar um alvo desejado, o usuário pressiona uma das teclas e a região do objeto é aumentada como uma lupa. O usuário olha novamente para o alvo e libera a tecla, executando a ação apropriada no objeto em questão. Como o foco do nosso protótipo são PNE, esse modelo torna-se inviável para essa dissertação.

O sistema EYCIN criado por Hermann [14], apresenta uma das melhores alternativas de emulação dos botões do mouse dentre os dispositivos pesquisados. O sistema trabalha com “regiões sensitivas” na tela, para que o usuário observe essas áreas e o sistema interprete como uma ação do usuário. Para diminuir cliques desnecessários, as regiões sensitivas mudam de cor duas vezes antes de realizar o clique.

Souza *et al.* [42] utilizam o piscar dos olhos para simular o duplo clique do botão esquerdo do mouse. O ato de clicar do mouse é executado quando o usuário aponta para o alvo e permanece por 2 segundos com os olhos fechados. Esse é um modelo interessante, pois através do Rastreo Ocular dos dois olhos é possível simular os cliques dos botões esquerdo e direito do mouse. Adotamos esse modelo para o desenvolvimento do nosso protótipo, e assumimos o risco de um possível estresse causado ao usuário por seqüências de cliques do mouse com os olhos.

O Rastreo Ocular não fica limitado apenas à região da tela, é possível determinar se o usuário está observando áreas externas ao monitor. Isso permite criar botões virtuais nessas áreas para possibilitar que o usuário execute tarefas simples do Sistema Operacional. Os botões virtuais são utilizados nos mouses do tipo *touchpad*, onde é possível configurá-los para realizar ações como: maximizar, minimizar, abrir o menu inicial entre outros. O processo de minimização foi implementado no protótipo para ser avaliado durante a etapa de testes do sistema.

## 4.4 – Sistemas Auxiliares

Foram criados dois sistemas extras para nos auxiliar nas etapas de desenvolvimento e de testes do nosso protótipo. A seguir são abordadas as aplicabilidades de cada um deles.

### 4.4.1 – Sistema de Detecção de Círculos em Imagens Digitais

Antes do desenvolvimento do sistema de Rastreo Ocular foi realizado um estudo para a escolha do algoritmo da Transformada de Hough que seria utilizado para detectar a

circunferência da íris nas imagens capturadas em tempo real pelo Dispositivo Rastreador. O sistema de detecção de círculos em imagens digitais permitiu analisar aspectos como: elementos analisados, elementos pertencentes à borda, círculos encontrados, incrementos na matriz acumuladora, pico na matriz acumuladora e tempo computacional de processamento.

Foram analisadas imagens de diversos tamanhos e formatos, capturadas da região frontal dos olhos de vários indivíduos. Dessa forma, foi possível avaliar o desempenho da Transformada de Hough na detecção da íris em indivíduos de olhos claros e escuros. A maioria das imagens analisadas é da base de dados biométrica UBIRIS [35]. As imagens processadas durante o desenvolvimento dos algoritmos do nosso dispositivo estão ilustradas no Anexo I.

#### 4.4.2 – Sistema de Atalhos

Um dos maiores problemas do Rastreo Ocular através de análise de imagens é a precisão do rastreo. O Dispositivo Rastreador pode apresentar diferenças de até 50 *pixels* na precisão do alvo observado. Definir exatamente o *pixel* observado na tela pelo usuário não é trivial. Desse modo, permitir ao usuário abrir programas através dos atalhos do seu ambiente de trabalho pode ser uma tarefa difícil, tediosa e estressante.

Desta forma, desenvolvemos um sistema de atalhos, que acompanha o Dispositivo Rastreador, para auxiliar na execução de alguns programas básicos instalados no computador. O sistema de atalhos é composto por doze botões de tamanho 200 x 200 *pixels*, que foram configurados para permitir ao usuário executar os seguintes programas: Internet Explorer<sup>®</sup>, Windows Live Mail<sup>®</sup>, Windows Live Messenger<sup>®</sup>, Skype<sup>®</sup>, Adobe Acrobat Reader<sup>®</sup>, Word<sup>®</sup>, Excel<sup>®</sup>, Power Point<sup>®</sup>, Menu Iniciar do Sistema Operacional, Windows Media Player<sup>®</sup>, Paciência e o Bloco de Notas.

Para facilitar a interação do usuário com o computador, foi configurado um botão extra-monitor para sobrepor o sistema de atalhos sob as demais janelas abertas em seu ambiente de trabalho. Optamos por ativar o sistema de atalhos, trazendo-o para frente de todas as janelas, todas as vezes que o usuário observar a região superior externa ao monitor. Isso pode tornar mais fácil a navegação do usuário no Sistema Operacional, principalmente aos usuários Portadores de Necessidades Especiais.

### 4.5 – Teclado Óptico Ocular

Algumas pesquisas de Rastreo Ocular voltadas para Portadores de Necessidades

Especiais usam o “teclado virtual” para simular o dispositivo de entrada padrão do computador. O teclado virtual é uma ferramenta de acessibilidade que acompanha os Sistemas Operacionais, sendo que a maioria permite a entrada de texto através de cliques do mouse sobre uma imagem de teclado. Digitar frases letra a letra com o teclado virtual é um processo lento e tedioso, por isso alguns sistemas utilizam técnicas de Inteligência Artificial para oferecer ao usuário uma palavra mais próxima ao texto que ele está digitando.

Para auxiliar o seu Dispositivo Rastreador, Souza *et al.* [42] criaram um teclado virtual que fornece praticamente todas as funcionalidades de um teclado convencional. Para ativar um botão do teclado, o usuário deve mover o cursor até o botão e ficar com os olhos fechados durante 2 segundos. O teclado virtual criado por Wang, Zhai e Su [48] é específico para a escrita de ideogramas do idioma chinês, e utiliza o MAGIC Pointing para ativar os seus botões.

O grande problema de alguns teclados virtuais são os botões extremamente pequenos. Posicionar o cursor do mouse sobre eles pode ser uma tarefa difícil. Se a taxa de erro do rastreamento for superior ao tamanho do botão, significa que o usuário terá muita dificuldade para digitar textos através dos olhos. Outro fator importante é o tempo necessário para posicionar o cursor no botão. Nos testes realizados por Souza *et al.* [42], o tempo médio necessário para que o usuário movesse o cursor da letra Q à letra P no teclado virtual foi de 33 segundos.

A solução adotada nessa dissertação é a criação de um Teclado Óptico Ocular que permite ao Portador de Necessidades Especiais escrever palavras e frases através de seqüências de movimentos oculares em regiões sensíveis na tela. A tela pode ser dividida em  $n$  regiões e, para cada letra do alfabeto, é criado um mapeamento para definir os movimentos oculares que o usuário deve realizar para escrevê-las. Uma seqüência de movimentos oculares é mais rápida de ser realizada do que a movimentação do cursor do mouse entre os botões do teclado virtual.

A idéia de utilizar regiões sensíveis na tela surgiu através do sistema desenvolvido por Perlin [31]. O *Quikwriting* utiliza um *joystick* para substituir o teclado na escrita de textos, onde através de seqüências pré-definidas é possível escrever os caracteres. O *joystick* permite realizar oito movimentos em diferentes direções, além da posição de descanso central. A vantagem de utilizar esse modelo de escrita, é que eliminamos o requisito de posicionar o cursor do mouse em uma pequena região da tela (botões do teclado virtual) antes de emular o clique do mouse.

Essa técnica de escrita é similar à utilizada por dispositivos portáteis (PDA) para entrada de textos em seus aplicativos. A única diferença é o uso dos olhos ao invés das canetas que acompanham os PDA's. Uma das maiores desvantagens dessa técnica é a impossibilidade do

usuário escrever de forma cursiva, sendo possível apenas a escrita de textos, letra a letra. O importante é permitir que o usuário passe por uma fase de adaptação ao sistema, para que possa decorar o máximo de seqüências possíveis de cada letra do alfabeto.

Para o desenvolvimento do nosso protótipo, optamos por inserir apenas um teclado com seqüências de caracteres minúsculos. O Capítulo 5 descreve o funcionamento do modelo adotado e os mapeamentos utilizados com base no sistema de Perlin [31].

## 4.6 – Considerações Finais

Neste capítulo foram abordadas as principais idéias e sugestões que surgiram durante a etapa de pesquisa bibliográfica sobre o Rastreio Ocular. As técnicas e métodos mais eficazes utilizados por outros pesquisadores para desenvolvimento dos seus respectivos Dispositivos Rastreadores foram utilizadas para a criação do protótipo montado nessa dissertação.

Uma contribuição desse capítulo é a concepção de um Teclado Óptico Ocular a partir do nosso protótipo de Dispositivo Rastreador. Esse “teclado óptico” é outra aplicação do nosso Dispositivo Rastreador como dispositivo de entrada de dados.

No próximo capítulo será apresentado em detalhes o processo de desenvolvimento do Mouse Óptico Ocular. Serão abordados temas como: as bibliotecas utilizadas para o processamento das imagens, os algoritmos da Transformada de Hough utilizados, e a integração das técnicas descritas nesse capítulo que formam o nosso Dispositivo Rastreador.

---

# Desenvolvimento

---

**D**urante o desenvolvimento do Mouse Óptico Ocular, foram realizados alguns testes práticos para a escolha das melhores ferramentas e algoritmos que seriam utilizados na criação do nosso Dispositivo Rastreador. Esse capítulo mostra em detalhes o processo de criação do Mouse Óptico Ocular, que abrange desde o estudo da viabilidade do uso da Transformada de Hough no processo de detecção de círculos até a conclusão do protótipo criado nessa dissertação.

## 5.1 – Viabilidade do Uso da Transformada de Hough

O primeiro passo no desenvolvimento do Mouse Óptico Ocular foi realizar um estudo (teórico e prático) sobre a Transformada de Hough, com objetivo de avaliar o seu uso no desenvolvimento de algoritmos para detecção da íris em imagens digitais. Esperava-se que a Transformada de Hough apresentasse bom grau de precisão na detecção da borda da íris, e esse fato ocorreu mesmo nas imagens capturadas com as pálpebras cobrindo parcialmente a íris.

Além disso, devido a gama de parametrizações da Transformada de Hough existente, era necessário estabelecer a parametrização que melhor se adaptasse ao processo de detecção da íris em imagens capturadas em tempo real, mantendo as taxas de precisão e de erro em um nível aceitável. O estudo foi realizado utilizando duas bibliotecas de Processamento Digital de Imagens: a TerraLib e a OpenCV. Na seqüência as bibliotecas são apresentadas e seus respectivos papéis no desenvolvimento do Mouse Óptico Ocular.

### 5.1.1 – Bibliotecas de Processamento Digital de Imagens

A TerraLib [43] é um projeto de software livre coordenado pelo Instituto Nacional de Pesquisas Espaciais (INPE), que permite o trabalho cooperativo e padronizado entre a

comunidade de desenvolvimento de aplicações geográficas. Entre suas principais características, a TerraLib é um biblioteca de classes construída em linguagem C/C++, multiplataforma, que pode ser integrada a diversos gerenciadores de banco de dados (Oracle, MySQL e PostgreSQL). Apesar de ser uma biblioteca para desenvolvimento de sistema de Geoprocessamento, a TerraLib tem incorporada um módulo exclusivo para Processamento Digital de Imagens (TePDI).

As parametrizações da Transformada de Hough implementadas na TerraLib seguiram as regras de modelagem da biblioteca. Foram criados dois arquivos (TePDIHoughTransform.cpp e TePDIHoughTransform.hpp) com os métodos padrões CheckParameter, RunImplementation e ResetState. O CheckParameter verifica se todos os parâmetros requeridos para a execução do algoritmo foram informados, o método RunImplementation realiza a execução do algoritmo e o ResetState limpa os parâmetros informados anteriormente para uma nova execução do algoritmo.

Para executar o algoritmo de detecção de circunferências, é necessário que sejam informados: o caminho da imagem de entrada (input\_image), o caminho da imagem de saída (output\_image), o identificador da parametrização da Transformada de Hough a ser utilizada (transformType), quantidade de canais da imagem de entrada (channels), o raio mínimo e máximo (radiusMin e radiusMax) e o valor do limitante de Hough (houghThreshold).

Além do algoritmo de detecção de circunferências, foi implementado o algoritmo de detecção de linhas retas para a TerraLib. Esse algoritmo foi implementado apenas para ser disponibilizado para a comunidade científica, não sendo usado em nenhuma etapa no processo de Rastreo Ocular durante os trabalhos desta dissertação.

Inicialmente os testes dos algoritmos foram realizados apenas com imagens digitais, pois até a versão TerraLib 3.1.2, o TerraLib Kernel não tinha incorporado um módulo de captura de vídeos em tempo real. Foi necessário criar uma biblioteca dinâmica (DLL) para integrar o Dispositivo de Captura de Vídeo com a biblioteca TerraLib. Daí surgiu a necessidade de utilizar uma biblioteca com características primárias de análise de vídeos capturados em tempo real, onde as imagens fossem analisadas em memória sem a necessidade de salvá-las em disco.

A OpenCV [29] (Intel Open Source Computer Vision Library) é uma biblioteca destinada a auxiliar no desenvolvimento de aplicativos na área de Visão Computacional, desenvolvida pela Intel em meados do ano 2000. A biblioteca OpenCV contém funções escritas em C/C++ com os algoritmos mais populares de Visão Computacional e Processamento Digital de Imagem. Além de ser uma biblioteca multiplataforma contém um módulo destinado a processamento de imagens capturadas em tempo real.

A estrutura utilizada pela OpenCV destinada a representar imagens digitais nos processos internos de análise é chamada de `IPLImage`. A `IPLImage` é bastante organizada e possui campos como número de canais espectrais, dimensões da imagem (largura e altura), tamanho, matriz de dados e até informações mais específicas como máscara de cores. A OpenCV permite que qualquer estrutura interna da biblioteca possa ser salva em um arquivo de formato XML e posteriormente recarregada para uma nova instância.

São disponibilizadas funções básicas de adição, subtração, divisão, multiplicação, busca por maiores e menores valores, operações lógicas e ainda funções específicas de matrizes como matriz transposta e inversa. Nos algoritmos de Processamento Digital de Imagens estão disponíveis funções de suavização, filtros de detecção de bordas, funções para conversão e reamostragem de imagens e até filtros morfológicos. Também estão disponíveis funções para desenhar polígonos nas imagens processadas pela biblioteca.

A OpenCV disponibiliza o algoritmo da Transformada de Hough para detecção de linhas retas nas imagens processadas, porém até a versão OpenCV 5 RC1 não tinha disponível nenhuma parametrização da Transformada de Hough para detecção de circunferências. Dessa forma, baseado nos algoritmos de detecção de círculos implementados para a TerraLib, criou-se um filtro de detecção de círculos para análise em estruturas `IPLImage`.

Apesar da biblioteca OpenCV conter um módulo de aquisição e manipulação de imagens a partir de dispositivos de captura de vídeo, optamos pelo desenvolvimento de um controlador em baixo nível que manipulasse a aquisição de imagens quase em nível de hardware. Por hipótese, isso poderia garantir um desempenho superior do nosso Dispositivo Rastreador em comparação à utilização do módulo nativo de captura da OpenCV.

Os testes dos algoritmos implementados foram realizados em imagens digitais e vídeos capturados em tempo real. Devem ser considerados, nos relatos a partir desse ponto, que os testes em imagens digitais foram realizados com a estrutura implementada com a biblioteca TerraLib, e a análise das imagens capturadas em tempo real com a biblioteca OpenCV.

### **5.1.2 – Implementação dos Algoritmos de Detecção de Círculos**

A avaliação da Transformada de Hough no processo de detecção da circunferência da íris em imagens digitais, teve início com a implementação dos algoritmos de detecção de círculos parametrizados por Duda e Hart [10] e Kimme, Ballard e Sklansky [18], bem como uma extensão desses algoritmos adicionando aos dois casos a Transformada Probabilística de Hough [19][44].

A nossa maior preocupação na implementação do algoritmo de detecção de círculos era adaptar uma das parametrizações da Transformada de Hough para processar ao menos cinco imagens por segundo no processo de Rastreo Ocular. Era necessário analisar todas as funções do algoritmo em busca do melhor desempenho, desde a criação da matriz acumuladora em memória, até o cálculo do Ponto de Interesse do usuário no monitor.

### 5.1.3 – Criação da Matriz Acumuladora em Memória

O processo de detecção de círculos com a Transformada de Hough cria uma representação tridimensional do Espaço de Hough na memória para armazenar os incrementos gerados pela imagem analisada. Para criar o espaço de parâmetros, o algoritmo desenvolvido por Dorini e Rocha [8] verifica o tamanho do maior lado da imagem analisada e utiliza esse valor para criar uma matriz de tamanho  $[n][n][n]$ , conforme a representação da Figura 28.

```

01 int n = (width > height ? width : height);
02 int accum[n][n][n];
03 for (int radius = 0; radius < n; radius++)
04     for (int a = 0; a < n; a++)
05         for (int b = 0; b < n; b++)
06             accum[radius][a][b] = 0;

```

Figura 28: Criação do espaço de parâmetros tridimensional segundo o algoritmo de Dorini e Rocha [8]

Por definição, o menor círculo detectável no algoritmo de Dorini e Rocha [8] é de raio igual à 10 pixels, e o maior raio de tamanho igual à  $n$  pixels. Segundo Dorini e Rocha [8] é desnecessário processar raios muito pequenos, pois geralmente o que representam não é significativo em uma figura real. Nem todos os processos de detecção de círculos precisam encontrar todos os círculos existentes na imagem. Por exemplo, as implementações de Powell [34] e Pistori *et al.* [32] detectam apenas os círculos em um intervalo pré-definido no algoritmo entre  $[radiusMin \sim radiusMax]$ , conforme ilustra a Figura 29.

```

01 double radiusMin = 38.;
02 double radiusMax = 40.;
03 double radiusRan = radiusMax - radiusMin;
04 int **accum = new *[(int) radiusRan + 1];
05 for (int i = 0; i <= (int) radiusRan; i++) {
06     accum[i] = new int[width * height];
07     memset(accum[i], 0, width * height * sizeof(int));
08 }

```

Figura 29: Criação do espaço de parâmetros segundo os algoritmos de Powell [34] e Pistori *et al* [32].

O algoritmo da Figura 29 cria o espaço de parâmetros para detectar círculos em um intervalo de raios, informados antes da execução do algoritmo. Assintoticamente, a criação do

espaço de parâmetros de acordo com o algoritmo de Powell [34] e Pistori *et al.* [32] é linear  $O(n)$  em relação ao intervalo dos círculos, bem menor em comparação ao algoritmo de Dorini e Rocha [8] que é cúbico  $O(n^3)$  em relação ao tamanho do maior lado da imagem analisada.

Para atingir o desempenho de ordem linear  $O(n)$ , o algoritmo da Figura 29 utiliza uma representação bidimensional em vez da tridimensional. Na matriz bidimensional, uma dimensão representa o intervalo de raios a serem detectados e a outra representa os elementos da imagem analisada. Nesse caso, para saber qual o elemento  $(x, y)$  que está sendo processado na imagem, é necessário realizar a seguinte transformação:  $x = i/width$  (para representar a linha) e  $y = i \% width$  (para representar a coluna).

O algoritmo da Figura 29 foi o escolhido para criar o Espaço de Hough no nosso protótipo. É importante lembrar que para iniciar o processo de incrementos, todos os elementos da matriz acumuladora devem ser iguais a zero. Nesse caso, como o espaço de parâmetros é criado através de alocação dinâmica de memória, para cada imagem analisada é necessário liberar os recursos alocados pelo algoritmo na criação da matriz acumuladora, conforme a Figura 30.

```
01 for (int i = 0; i <= (int) radiusRan; i++) {
02     free(accum[i]);
03 free(accum);
```

Figura 30: Processo de liberação dos recursos alocados na representação do espaço de parâmetros na memória principal

### 5.1.4 – Implementação do Algoritmo de Duda e Hart

O algoritmo implementado de acordo com a parametrização de Duda e Hart [10], requer uma imagem de entrada em tons de cinza. Antes do processo de detecção de círculos, é necessário aplicar o Filtro de Sobel [13] na imagem. Para cada elemento analisado, um limitante determina se é necessário realizar o mapeamento do espaço da imagem para o espaço de parâmetros. Ao final do processo, o algoritmo varre a matriz acumuladora em busca das suas máximas locais. A Figura 31 mostra a etapa de incremento de acordo com Duda e Hart [10].

```
01 for (int j = 0; j < height; j++)
02     for (int i = 0; i < width; i++)
03         if (image[i][j] > threshold) {
04             for (int x = 0; x < (width * height); x++) {
05                 int radius = sqrt (pow((i - (x % width)), 2) +
06                                     pow((j - (x / width)), 2));
07                 int r = (int) radius - radiusMin;
08                 accum[r][x]++;
09             }
```

Figura 31: Detecção de círculos segundo a parametrização de Duda e Hart [10]

A Figura 18 do item 3.3.1, mostra 5 circunferências com pontos vazados representando os seus centros. Ao incrementar os possíveis centros das circunferências que interceptam o ponto  $(0,0)$  da coordenada, além de incrementar os pontos centrais, o algoritmo incrementa outros pontos representados pelos pontos cheios. O que difere os pontos cheios dos vazados, é que ao analisar os demais pontos pertencentes à borda da circunferência, o ponto central será incrementado várias vezes e ao final irá apresentar as máximas locais da matriz acumuladora.

A Figura 32 ilustra um exemplo do processo de detecção de círculos com a parametrização de Duda e Hart [10]. A imagem da esquerda é similar a Figura 18, porém sem as coordenadas e a representação dos possíveis centros dos círculos. A imagem da direita mostra o espaço de parâmetros bidimensional da imagem analisada, com os pontos claros representando os elementos mais incrementados na matriz acumuladora. Por último, a Figura 33 ilustra uma visualização tridimensional do espaço de parâmetros. Os picos dos cones representam os parâmetros centrais de cada uma das cinco circunferências da imagem analisada.

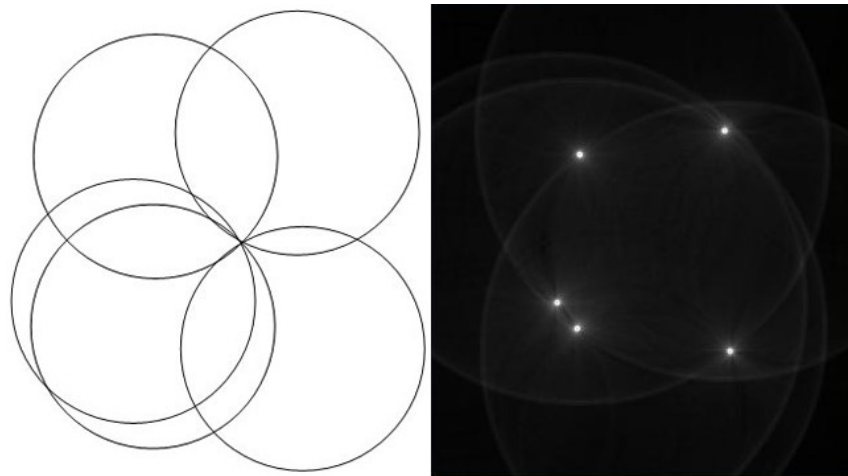


Figura 32: Imagem analisada com o algoritmo da Figura 31 e seu respectivo espaço de parâmetros bidimensional

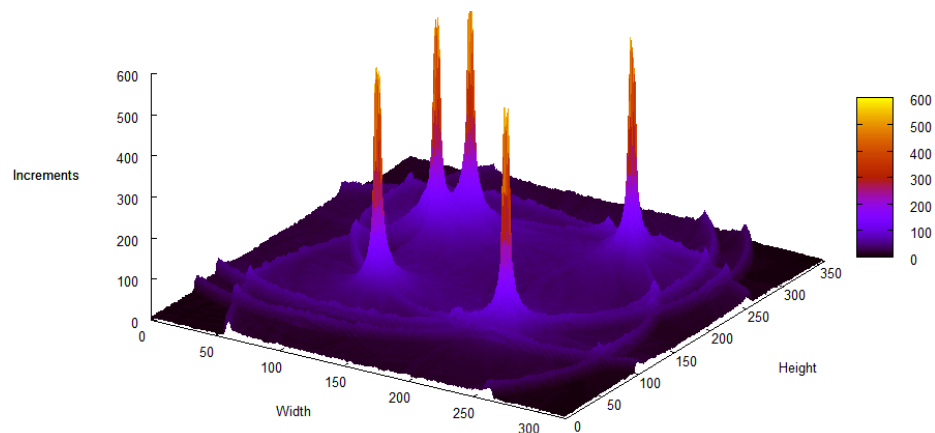


Figura 33: Visualização tridimensional do espaço de parâmetros da Figura 32

### 5.1.5 – Implementação do Algoritmo de Kimme, Ballard e Sklansky

Da mesma forma que o algoritmo anterior, a parametrização de Kimme, Ballard e Sklansky [18] também requer uma imagem de entrada em tons de cinza. Ao invés de aplicar o Filtro de Sobel [13] na imagem antes do processo de detecção de círculos, o filtro é aplicado no momento em que o elemento é processado. Com os componentes horizontais e verticais, calcula-se a direção do gradiente para eliminar porções de possíveis centros de círculos na imagem.

A Figura 34 ilustra o resultado do processo de detecção de círculos com a parametrização de Kimme, Ballard e Sklansky [18]. Observa-se no espaço de parâmetros tridimensional que os cones da Figura 35 são menores em relação aos cones da Figura 33. Isso ocorre, pois o número de incrementos na matriz acumuladora com o algoritmo da Figura 37 é menor do que com o algoritmo da Figura 31. Dessa forma foi necessário reajustar o limiar de detecção de círculos para que não ocorresse a detecção de falsos círculos na imagem analisada.

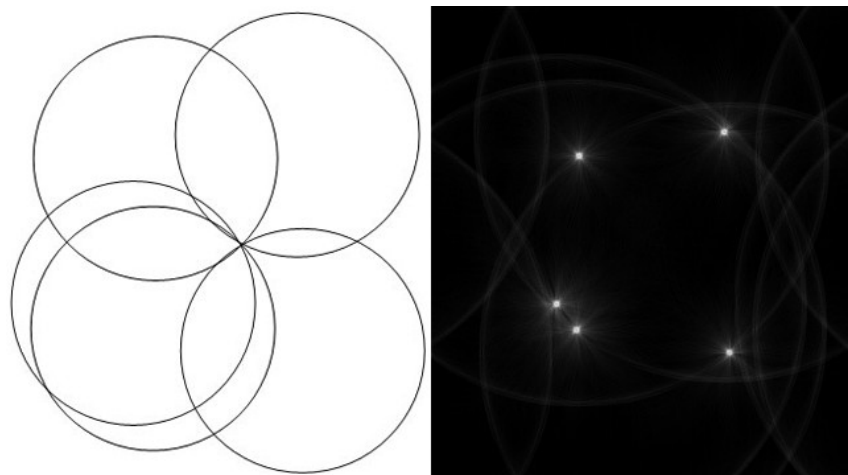


Figura 34: Imagem analisada com o algoritmo da Figura 37 e seu respectivo espaço de parâmetros bidimensional

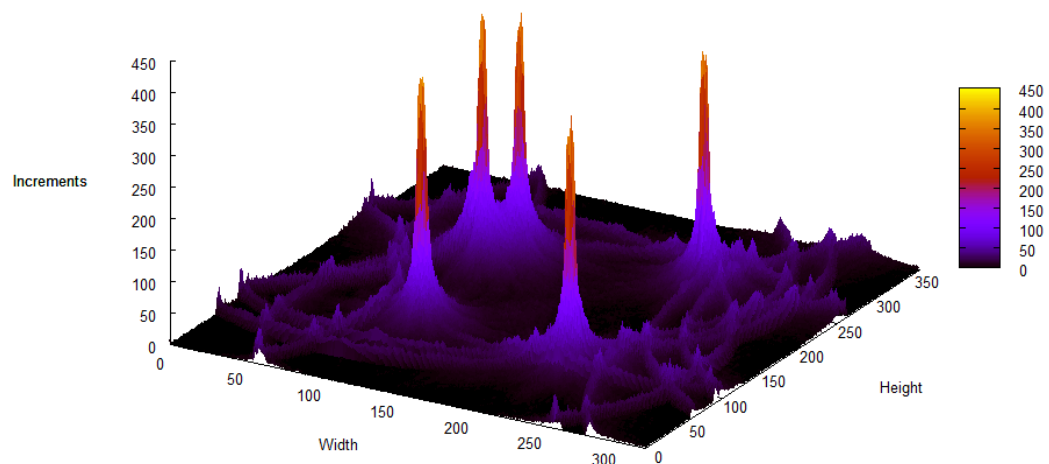


Figura 35: Visualização tridimensional do espaço de parâmetros da Figura 34

O primeiro algoritmo implementado para esta dissertação com a parametrização de Kimme, Ballard e Sklansky [18] foi o ilustrado na Figura 36. Nesse caso, apenas um elemento era incrementado para cada ponto pertencente à borda analisado. As imagens analisadas com o esse algoritmo geravam espaços de parâmetros com máximas locais muito baixas, em torno de 15 a 20 incrementos nos elementos de pico. Além do número baixo de círculos detectados por imagens, o algoritmo detectava círculos em cerca de 83% das imagens analisadas.

---

```

01 for (int j = 0; j < height; j++)
02   for (int i = 0; i < width; i++)
03     if (image[i][j] > threshold) {
04       int direct = atan (gy / gx);
05       for (double radius = radiusMin; radius <= radiusMax; radius++) {
06         double a = i - radius * cos(direct);
07         double b = j - radius * sin(direct);
08         int r = (int) radius - radiusMin;
09         int x = (int) a + b * width;
10         accum[r][x]++;
11       }
12   }

```

---

Figura 36: Detecção de círculos segundo a parametrização de Kimme, Ballard e Sklansky [18]

Com base nas idéias do item 3.3.2, para obter um melhor mapeamento no espaço de parâmetros, deve-se realizar o incremento do conjunto de elementos  $C_x$  em um ângulo maior que  $40^\circ$ . De acordo com Dorini e Rocha [8], como o cálculo da direção do gradiente trabalha com a equação de arco-tangente, deve-se considerar também a direção oposta do gradiente  $g(x)$ . É importante salientar que as variáveis “k” e “delta” sejam de tipo ponto flutuante (*float*) para calcular corretamente o intervalo  $\Delta\theta$  e garantir que todos os seus índices sejam incrementados, evitando possíveis erros de arredondamento nos valores processados.

---

```

01 for (double radius = radiusMin; radius <= radiusMax; radius++) {
02   float k = 0.8 * radius;
03   float delta = 1 / radius;
04   for (float l = (-k / 2); l <= (k / 2); l++) {
05     double a = i + radius * cos(direction + (l * delta));
06     double b = j + radius * sin(direction + (l * delta));
07     int r = (int) radius - radiusMin;
08     int x = (int) a + b * width;
09     accum[r][x]++;
10   }
11   for (float l = (-k / 2); l <= (k / 2); l++) {
12     double a = i + radius * cos(direction + PI + (l * delta));
13     double b = j + radius * sin(direction + PI + (l * delta));
14     int r = (int) radius - radiusMin;
15     int x = (int) a + b * width;
16     accum[r][x]++;
17   }
18 }

```

---

Figura 37: Incrementos do conjunto  $C_x$  entre um ângulo de  $45^\circ$  e o ângulo oposto de  $-45^\circ$

A Figura 38 ilustra os dois processos de incrementos de acordo com a parametrização de Kimme, Ballard e Sklansky [18]. A modificação no número de elementos incrementados a cada ponto analisado, em um determinado ângulo, aumentou o tempo de processamento do algoritmo. Porém, o número de círculos detectados por imagens cresceu e o algoritmo passou a detectar a circunferência da íris em 100% das imagens analisadas.

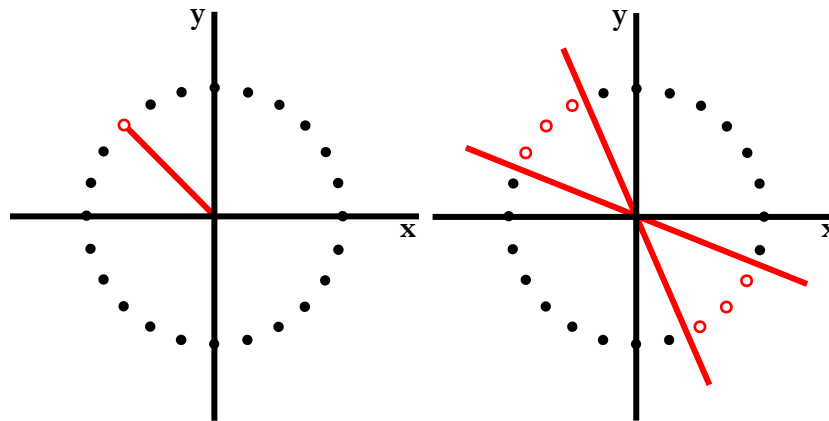


Figura 38: Conjunto de elementos incrementados com a parametrização de Kimme, Ballard e Sklansky [18]

### 5.1.6 – Implementação da Transformada Probabilística de Hough

Com as duas principais parametrizações implementadas, foi adicionado aos dois casos a Transformada Probabilística de Hough, apresentada no item 3.4.5. O algoritmo implementado segue a definição de Kiryati, Eldar e Bruckstein [19], onde são processadas pequenas porções dos pontos da imagem analisada. Esses pontos são escolhidos randomicamente de acordo com uma função de densidade de probabilidade uniforme (*rand*), como ilustrado na Figura 39.

```

01 float percent = 0.2;
02 while (numberPixels < ((width * height) * percent)) {
03   int i = rand() % (width - radiusMax) + radiusMax;
04   int j = rand() % (height - radiusMax) + radiusMax;
05 }

```

Figura 39: Seleção randômica dos elementos de imagens analisados no processo da Transformada Probabilística de Hough

### 5.1.7 – Testes da Viabilidade do Uso da Transformada de Hough

Finalizada a etapa de implementação das parametrizações de detecção de círculos, os algoritmos foram testados para verificar o desempenho da Transformada de Hough na detecção da íris em imagens digitais. Na primeira etapa foram analisadas as 12 imagens digitais ilustradas na Figura 40. Dentre algumas características das imagens, todas possuem 3 canais espectrais RGB, tamanho de 200x150 *pixels* e apresentam bom contraste e iluminação ambiente. O fator de

iluminação ambiente contribui diretamente no desempenho dos algoritmos na detecção da íris.

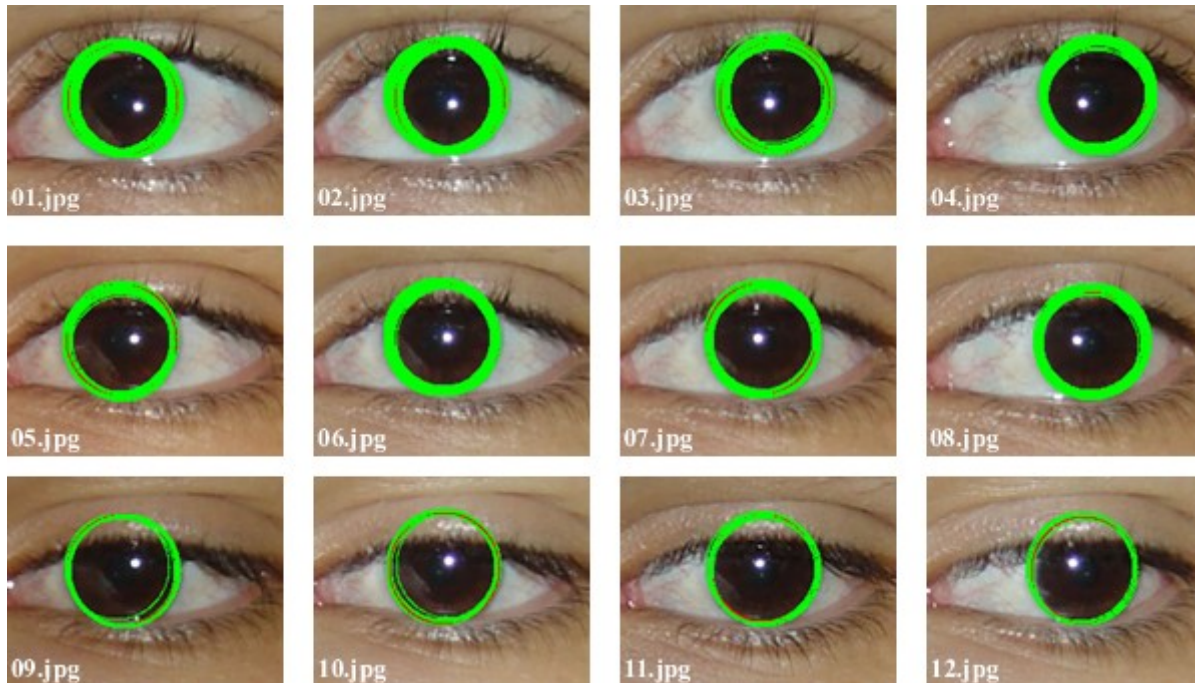


Figura 40: Amostra das imagens utilizadas no processo de detecção da circunferência da íris em imagens digitais

Na primeira etapa de testes utilizamos um computador com a seguinte configuração: processador AMD Athlon 64 2200 MHz (11 x 200) 3200+, chipset nVIDIA nForce3 250, 1024 MB DDR SDRAM 400 MHz dual channel, Sistema Operacional Microsoft Windows 2003 Server. Para definir a taxa de acerto do algoritmo, para cada imagem analisada foi informado o valor  $(x,y)$  do centro da íris com um clique mouse na imagem. Foram considerados para taxa de acerto os centros de círculos detectados até 8 *pixels* de distância do valor informado. O intervalo de raios utilizado para detecção de circunferências nas imagens analisadas foi entre 39 e 41 *pixels*.

Na tentativa de conseguir melhor desempenho computacional, foram adicionadas pequenas modificações em todos os algoritmos implementados. Ao invés de processar todos os elementos da imagem, foram excluídos alguns elementos que, por hipótese, não seriam necessários para a detecção correta da circunferência da íris. Por exemplo, uma imagem de tamanho de 200x150 *pixel* tem um total de 30.000 *pixels*. Se o objetivo for detectar circunferências de raio igual a 40 *pixels*, são excluídos 40 *pixels* das laterais e da parte superior e inferior da imagem. Dessa forma, ao invés de processar os 30.000 *pixels* da imagem, com apenas 8.400 *pixels* seria possível detectar a circunferência na imagem analisada.

A Figura 41 ilustra um exemplo dessa modificação. Ao invés de aplicar o Filtro de Sobel

[13] em toda a imagem (Figura 41a), o filtro foi aplicado somente aos *pixels* mapeados com a Transformada de Hough (Figura 41b). Testes empíricos mostraram que essa modificação só funciona corretamente com o algoritmo clássico de Kimme, Ballard e Sklansky [18]. Ao adicionar essa modificação no algoritmo de Duda e Hart [10] e aos algoritmos probabilísticos, a taxa de acerto do algoritmo caiu para cerca de 59% em média.

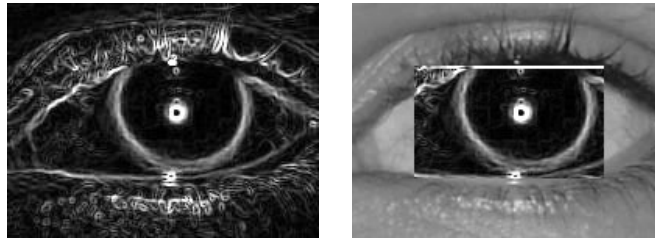


Figura 41: (a) 30.000 pixels analisados na imagem; (b) 8.400 pixels analisados na imagem

Algumas informações adicionais foram extraídas nos testes dos algoritmos implementados, entre elas: a quantidade de elementos processados na imagem, quantos elementos pertencem à borda detectada pelo Filtro de Sobel, a quantidade de círculos encontrados pelo algoritmo, quantos incrementos foram realizados na matriz acumuladora e o tempo de processamento do algoritmo.

Após a aplicação do Filtro de Sobel na imagem foram considerados *pixels* pertencentes à borda aqueles cujo o valor fosse maior que 100. A quantidade de círculos detectados variou conforme o algoritmo aplicado. Os resultados são ilustrados na Figura 42. Observamos que existe certa similaridade no número de círculos detectados nas imagens de 01 a 04, de 05 a 08 e de 09 a 12. Esse número nos ajudou na indicação da precisão vertical, informando ao sistema de Rastreamento Ocular se o usuário está olhando para a parte superior, média ou inferior do monitor.

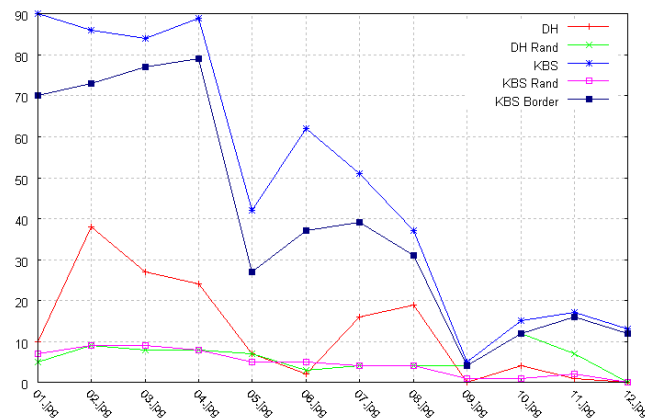


Figura 42: Número de círculos identificados nas imagens analisadas

Os algoritmos que apresentaram maiores incrementos na matriz acumuladora foram os

algoritmos clássicos de Duda e Hart [10] e Kimme, Ballard e Sklansky [18]. Das modificações do algoritmo de Kimme, Ballard e Sklansky [18], o algoritmo que exclui os *pixels* próximos as laterais da imagem ficou bem próximo em comparação ao seu algoritmo probabilístico. A Figura 43 mostra que o melhor desempenho no número de incrementos na matriz acumuladora foi dada ao algoritmo probabilístico com a parametrização de Duda e Hart [10].

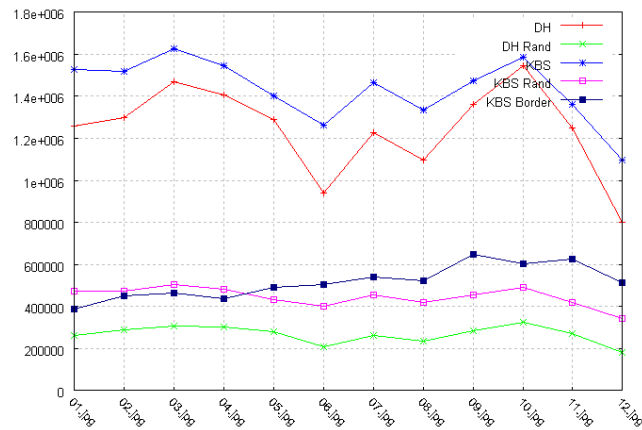


Figura 43: Número de incrementos na matriz acumuladora

Nos testes com o algoritmo clássico de Duda e Hart [10] além do alto número de *pixels* processados, o tempo de processamento é alto, em média de 9.009 ms com uma taxa de acerto de 100% das imagens analisadas. O uso desse algoritmo torna-se inviável para processar imagens capturadas em tempo real através de um dispositivo de captura de vídeo, processo essencial para a criação do Dispositivo Rastreador.

Uma boa alternativa implementada foi o algoritmo probabilístico com a parametrização de Duda e Hart [10]. Apesar das pesquisas de Ylä-Jääski e Kiryati [19][44] comprovarem que é possível detectar formas geométricas sem erro analisando 15% dos pontos da imagem, com o nosso algoritmo só foi possível atingir essa taxa com a análise de cerca de 20% dos pontos. A seleção aleatória com o algoritmo de Duda e Hart [10] melhorou o desempenho computacional do algoritmo, mantendo a taxa de acerto em cerca de 96% das imagens. Com o tempo médio de 2.389 ms é necessário reajustar o algoritmo para processar imagens capturadas em tempo real.

Nos testes realizados com o algoritmo clássico de Kimme, Ballard e Sklansky [18], em todas as imagens processadas foram encontrados círculos na região da íris. O tempo médio de processamento ficou em aproximadamente 234 ms. O algoritmo que processa apenas os pontos da região central da imagem obteve uma taxa de acerto de 100% com um tempo médio de apenas 93 ms. O uso da Transformada Probabilística de Hough no algoritmo de Kimme, Ballard e

Sklansky [18] também mostrou-se eficiente. A taxa de acerto ficou em torno de 94%, mas na média o tempo de processamento do algoritmo permaneceu quase o mesmo que o anterior, em torno de 78 ms. A Figura 44 ilustra um diagrama com o tempo de processamento dos cinco algoritmos implementados nessa dissertação.

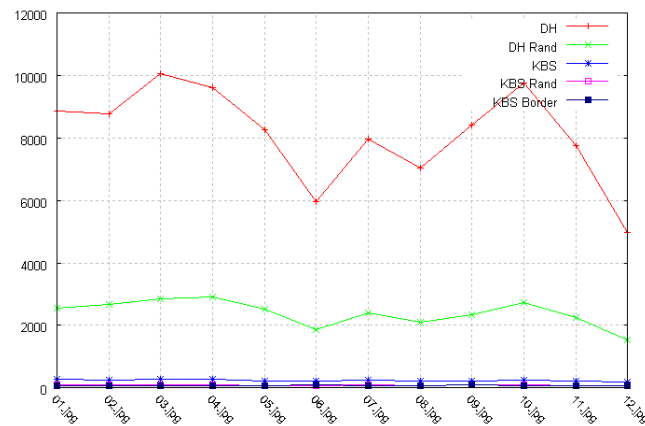


Figura 44: Tempo médio de processamento dos algoritmos da Transformada de Hough

### 5.1.8 – Conclusão do Estudo da Viabilidade da Transformada de Hough

Do estudo realizado concluímos que é viável o uso de algoritmos de detecção de círculos baseados na Transformada de Hough para o desenvolvimento do nosso protótipo de Dispositivo Rastreador. Esperávamos que a Transformada de Hough tivesse um bom desempenho na detecção da borda da íris, mesmo quando ocorresse uma cobertura parcial pelas pálpebras. Fato esse posteriormente comprovado na análise das imagens ilustradas na Figura 40.

Observamos que a parametrização da equação geral de circunferência, em função de  $a$  e  $b$  não produziu resultados computacionalmente eficientes. A melhor alternativa foi a utilização dos parâmetros  $a$  e  $b$  em coordenadas polares. A tentativa de excluir alguns elementos próximos à borda da imagem com o algoritmo de Duda e Hart [10] frustrou as expectativas. O principal motivo é que para melhores resultados, o algoritmo de Duda e Hart [10] tem um melhor desempenho se analisar todos os *pixels* pertencentes à borda da circunferência a ser detectada.

Ao contrário, o algoritmo de Kimme, Ballard e Sklansky [18] consegue detectar o círculo analisando apenas uma porção de *pixels* pertencentes à borda da circunferência. Nesse caso, concluímos que o algoritmo de Duda e Hart [10] tem um desempenho melhor para detecção de círculos em imagens sintéticas (sem ruídos) e o algoritmo de Kimme, Ballard e Sklansky [18] tem bom desempenho mesmo analisando imagens ruidosas. O algoritmo escolhido para o desenvolvimento do nosso protótipo de Dispositivo Rastreador foi o algoritmo baseado na

parametrização de Kimme, Ballard e Sklansky [18] que analisa apenas os elementos da região central da imagem capturada, excluindo os elementos próximos à borda da imagem.

## 5.2 – Sistema de Detecção de Círculos em Imagens

O sistema de detecção de círculos em imagem digitais foi utilizado para avaliar o desempenho da Transformada de Hough na detecção da circunferência da íris. É um sistema simples e intuitivo que permite ao usuário abrir uma imagem digital (compatível com os formatos JPG, JPGE, GIF, BMP e PNG) e escolher qual a parametrização da Transformada de Hough deseja executar na imagem em questão. Todas as parametrizações da Transformada de Hough citadas no tópico 5.1 se encontram nesse sistema.

Todos os algoritmos implementados executam três etapas de processamento. Na primeira etapa, a imagem de entrada é analisada para verificar se possui três canais espectrais e, nesse caso, a imagem é convertida para tons de cinza. Na segunda etapa, o Filtro de Sobel é aplicado na imagem para realçar as suas bordas. Na última etapa, o algoritmo de detecção de círculos é executado, retornando a coordenada  $(x,y)$  do centro da íris detectada na imagem.

As imagens analisadas na etapa de testes foram obtidas de duas fontes: imagens disponíveis na base de dados UBIRIS [35]; e imagens capturadas dos olhos de voluntários que autorizaram o seu uso exclusivamente para essa dissertação (as cópias das autorizações estão disponíveis no Anexo II). Ao total foram processadas cerca de 100 imagens, onde foi possível analisar o desempenho da Transformada de Hough sob diversos aspectos: indivíduos de olhos claros e escuros; indivíduos usando lentes oftálmicas; olhos entre abertos; entre outros. A Figura 45 ilustra um exemplo do processo de detecção de círculos em imagens digitais.

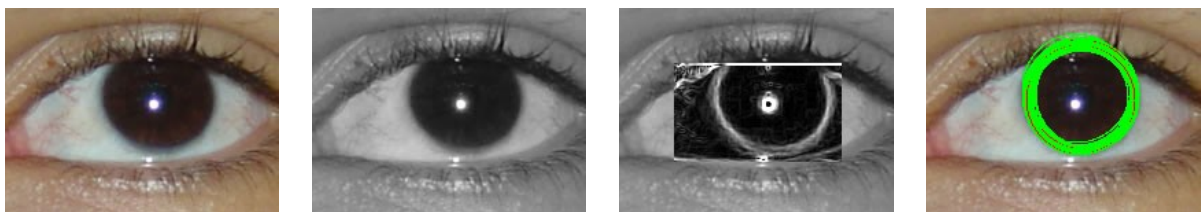


Figura 45: (a) imagem original (b) imagem em tons de cinza (c) imagem com Filtro de Sobel (d) resultado final

O sistema foi utilizado para obter alguns dados estatísticos do processamento de cada imagem analisada. Com esses dados foi possível determinar a parametrização mais apropriada para processar imagens capturadas em tempo real. As informações retornadas pelo sistema são as seguintes: elementos analisados, elementos detectados pelo Filtro de Sobel [13] pertencentes à

borda, círculos encontrados, incrementos na matriz acumuladora, pico na matriz acumuladora, círculos no pico da matriz e tempo computacional de processamento. A Figura 46 ilustra a janela de visualização das estatísticas, mostrada ao usuário após o processamento da imagem de entrada.

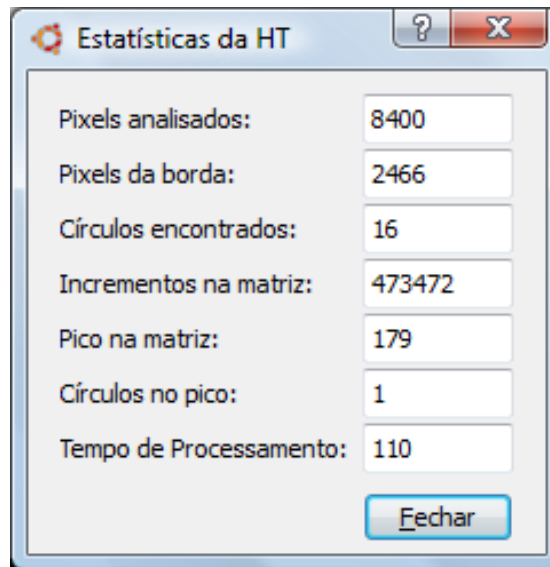


Figura 46: Janela com as estatísticas do processamento da imagem de entrada

Ao final do processamento da imagem, o sistema grava automaticamente na mesma pasta do arquivo de entrada uma seqüência de arquivos no formato JPG. Esses arquivos contêm as representações bidimensionais dos espaços de parâmetros de cada raio do intervalo processado pela Transformada de Hough, conforme ilustrado na Figura 47. Em um primeiro momento, essas imagens eram satisfatórias no estudo da Transformada de Hough, porém optamos por disponibilizar a análise do espaço de parâmetros em uma visualização tridimensional.

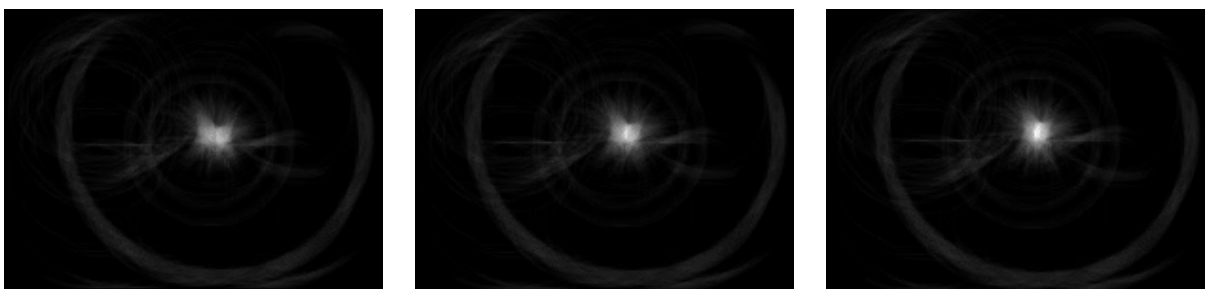


Figura 47: Visualização bidimensional do espaço de parâmetros (a) raio = 39 (b) raio = 40 e (c) raio = 41

Para possibilitar a visualização tridimensional da matriz acumuladora, criamos um arquivo de saída compatível com o sistema GNUPlot. O GNUPlot [12] é um *software* que permite plotar gráficos de funções matemáticas em duas ou três dimensões. Para gerar a visualização tridimensional, utilizamos a matriz acumuladora final do processamento da imagem para obter as

informações  $(x, y, z)$  do gráfico, onde:  $x$  representa a largura da imagem;  $y$  a altura da imagem; e  $z$  a quantidade de incrementos realizados na matriz acumuladora, no campo  $(x, y)$ . A Figura 48 ilustra um gráfico gerado a partir da análise da Figura 45, com raio igual à 39 *pixels*.

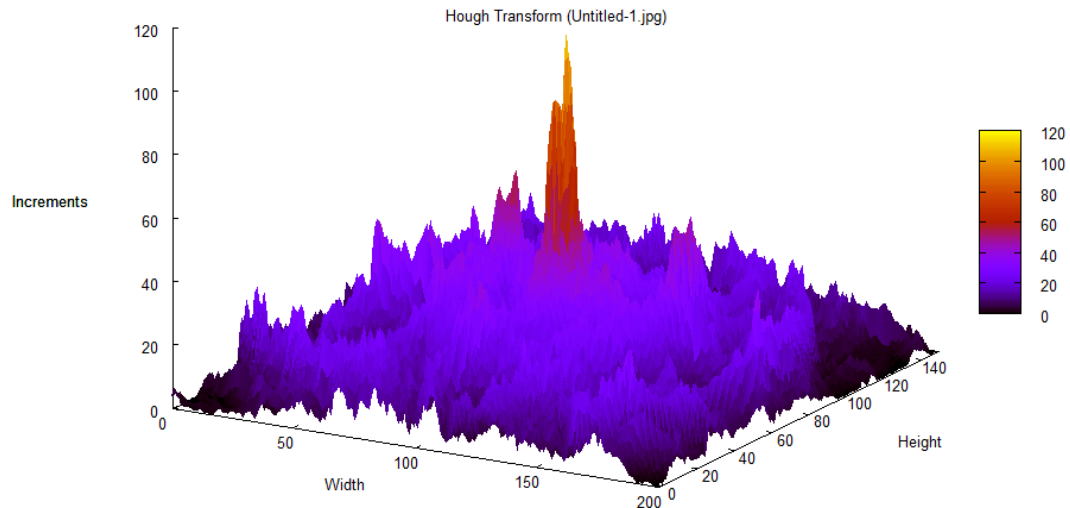


Figura 48: Visualização tridimensional do processamento de uma imagem com a parametrização de Duda e Hart

### 5.3 – Sistema de Detecção de Círculos em Vídeos

O processo de criação do sistema de detecção de círculos, em imagens capturadas em tempo real, teve como ponto inicial a escolha dos dispositivos de captura utilizados no processo de aquisição das imagens dos olhos do usuário. Foi levado em consideração alguns fatores como amostragem, iluminação e qualidade focal, que influenciam diretamente no processo de identificação da íris em algumas imagens capturadas. Durante os testes do sistema com dispositivos de captura de baixa qualidade, notamos que esses dispositivos geram ruídos nas imagens capturadas afetando principalmente a precisão do Rastreamento Ocular.

Na etapa de desenvolvimento foram utilizados dois modelos de câmeras de aquisição de vídeos: uma *Webcam* Troni PCC380 com taxa de aquisição de 15 imagens por segundo e resolução de 320x240 *pixels*; e uma câmera digital Sony DSC-P92 com taxa de aquisição de 30 imagens por segundo, resolução de 640x480 e com foco óptico de 3X. A câmera Sony DSC-P92 foi interligada a uma *framegrabber*, utilizada para digitalizar as imagens capturadas em tempo real. Somente no final do desenvolvimento utilizamos o Dispositivo Rastreador montado com a micro-câmera *wireless* e os óculos de proteção industrial, ilustrado na Figura 24 do item 4.1.1.

A integração do sistema de detecção de círculos em tempo real com o protótipo montado nessa dissertação utilizou “filtros multimídia” para trabalhar diretamente com os *drivers*

dos dispositivos de aquisição de vídeos instalados no computador do usuário. Dessa forma, para processar as imagens com o filtro da Transformada de Hough, o sistema permite que o usuário escolha quais dos dispositivos de aquisição de imagens instalados em seu computador ele deseja integrar o filtro com o decodificador do dispositivo de captura.

### 5.3.1 – Windows Driver Model (WDM)

O sistema de detecção de círculos em vídeos foi desenvolvido com a biblioteca Microsoft DirectX SDK, que atua com base no modelo Windows Driver Model (WDM). O Windows Driver Model é um modelo de desenvolvimento de *drivers* para dispositivos compatíveis com os Sistemas Operacionais Windows®, a partir da versão Windows® 98. O WDM permite que desenvolvedores criem uma comunicação entre as rotinas dos seus sistemas com os dispositivos multimídia instalados no computador do usuário.

Os principais atuadores do WDM são os filtros DirectShow, um componente da API DirectX para execução de acervos multimídia para a plataforma Windows®. O modelo WDM descreve como desenvolver filtros para trabalhar com várias opções multimídia, por exemplo: filtros para produzir determinados efeitos sonoros (como ecos) para ser integrado com o Windows Media Player®. Os filtros de renderização de vídeos são aplicados após a decodificação do vídeo capturado pelo DCV. O diagrama de aquisição e renderização de vídeos, de acordo com o modelo WDM, é ilustrado na Figura 49.

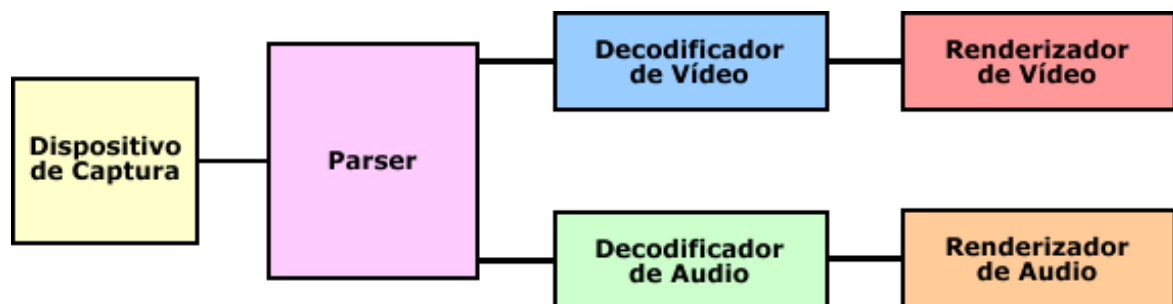


Figura 49: Diagrama de processamento de vídeos com filtros DirectShow

Com base no modelo de desenvolvimento de filtros DirectShow, foi criado um filtro da Transformada de Hough (HTFilter) para detectar o centro da íris nas imagens capturadas através dos Dispositivos de Captura de Vídeo instalados no computador do usuário. O filtro foi implementado com a biblioteca OpenCV e age como integrador entre o dispositivo de captura e o processo de análise das imagens capturadas. A Transformada de Hough é realizada em imagens digitais armazenadas temporariamente na memória principal do computador, e ao final do processo retorna a coordenada central da íris e exclui a imagem processada da memória.

### 5.3.2 – GraphEdit

O Microsoft DirectX SDK fornece a ferramenta gráfica chamada GraphEdit para testar filtros DirectShow. O GraphEdit auxilia desenvolvedores nos testes de filtros DirectShow durante a fase de implementação, para correção e ajustes de possíveis erros de programação. Durante a implementação do HTFilter, foi possível detectar erros de alocação de memória e ajustes na segmentação de círculos das imagens capturadas. A Figura 50 ilustra o ambiente de trabalho do GraphEdit com os filtros de captura de vídeo, da Transformada de Hough implementado e o renderizador que mostra o resultado em tempo real do processo de análise.

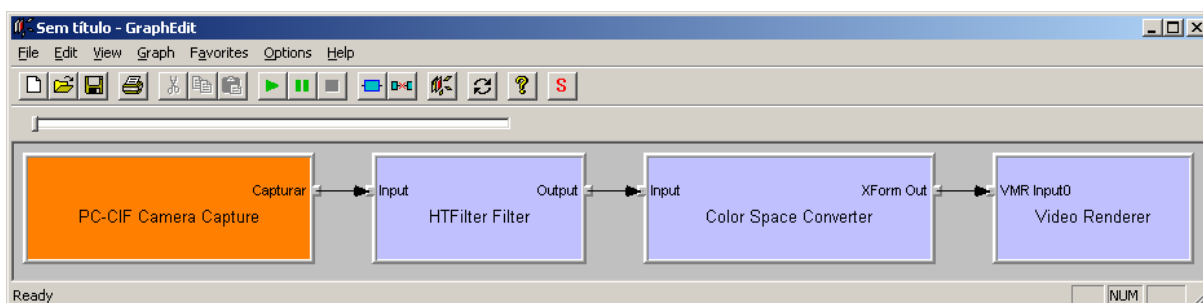


Figura 50: Ambiente de trabalho do GraphEdit com o filtro da Transformada de Hough

A Figura 51 ilustra um teste com a ferramenta GraphEdit utilizando o filtro renderizador (Video Renderer) em conjunto com o filtro HTFilter implementado nesta dissertação. Esse teste de detecção de circunferências foi realizado em imagens capturadas através da *Webcam* Troni PCC380 de resolução de 100KB. Observa-se na Figura 51 um fone de ouvido com uma goma de mascar esférica. O algoritmo identifica o raio da goma de mascar com bom grau de precisão e adiciona na região segmentada uma circunferência pontilhada.



Figura 51: Renderizador do GraphEdit

### 5.3.3 – Dispositivo de Captura de Vídeo

Durante o desenvolvimento do sistema de captura de vídeo, um dos problemas encontrados foi a seleção do dispositivo usado na execução do algoritmo de detecção de círculos. Em computadores com vários dispositivos de captura, o DirectShow seleciona automaticamente o dispositivo padrão instalado no computador do usuário. Dessa forma, foi adicionado um menu que apresenta todos os Dispositivos de Captura de Vídeo instalados no computador do usuário, para facilitar a escolha do dispositivo utilizado no processo de aquisição das imagens. Abaixo a Figura 52 ilustra a tela de seleção de dispositivos de captura adicionada ao controlador.



Figura 52: Seleção do dispositivo de captura de vídeo

Para testar a integração do filtro HTFilter com Dispositivo de Captura de Vídeo, utilizamos objetos esféricos para serem detectados nos vídeos capturados em tempo real. Com isso observamos o comportamento do filtro HTFilter em relação a alguns aspectos: o número de imagens processadas por segundo, a segmentação dos objetos circulares em movimento, a manipulação da imagem captura em memória, entre outros. A Figura 53 ilustra a etapa de testes de detecção em uma bola de massagem terapêutica através do Dispositivo de Captura de Vídeo.

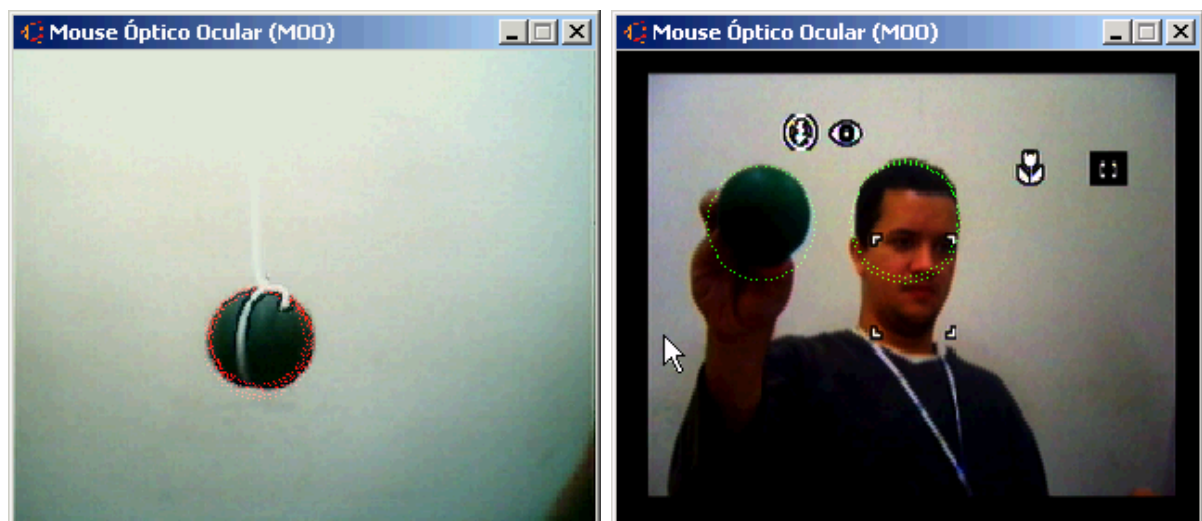


Figura 53: Detecção de bola de massagem terapêutica com raio entre 50 e 53 pixels

A Figura 53 à esquerda ilustra a primeira etapa de testes do algoritmo de detecção de

círculos com o Dispositivo de Captura de Vídeo. Foi detectada a bola de massagem terapêutica, com raio entre 50 e 53 *pixels*, amarrada a um cordão em movimentos de pêndulo. Mesmo com movimentos rápidos, o filtro HTFilter segmentou a imagem e detectou a bola amarrada ao cordão quadro-a-quadro nas imagens capturadas em tempo real.

Já a Figura 53 à direita ilustra o processo de detecção da mesma bola terapêutica também em movimento, porém dessa vez na mão de um usuário. Nesse caso o algoritmo detectou, além da bola terapêutica, a circunferência da cabeça do usuário como um círculo de raio entre 50 e 53 *pixels*. Esse foi o problema apresentado quando o dispositivo de captura estava a uma distância de aproximadamente 50 cm do usuário, detectando além do alvo proposto qualquer forma que tivesse a mesma curvatura no raio a ser detectado.

Com o dispositivo de captura distante da íris do usuário, as narinas, alguns cachos de cabelos e outras formas circulares também eram detectados. Para garantir melhor desempenho do algoritmo de detecção de círculos, foi necessário capturar imagens apenas dos olhos do usuário a uma distância média de 15 cm entre o dispositivo de captura e o olho. Nessas condições, eliminamos algumas situações que geravam ruídos nas imagens capturadas.

Em outra etapa de testes, a *Webcam* Troni PCC380 foi substituída pela câmera digital Sony DSC-P92 interligada a uma *framegrabber* PixelView PlayTV Pro. Devido a qualidade focal e o contraste da câmera digital Sony ser superior a *Webcam* Troni, o filtro HTFilter apresentou uma taxa de acerto e precisão de rastreamento, superior ao que vinha apresentando nos testes anteriores. A Figura 54 ilustra dois exemplos de imagens capturadas através da câmera digital Sony, detectando a íris de um usuário com olhos claros e outro com olhos escuros.

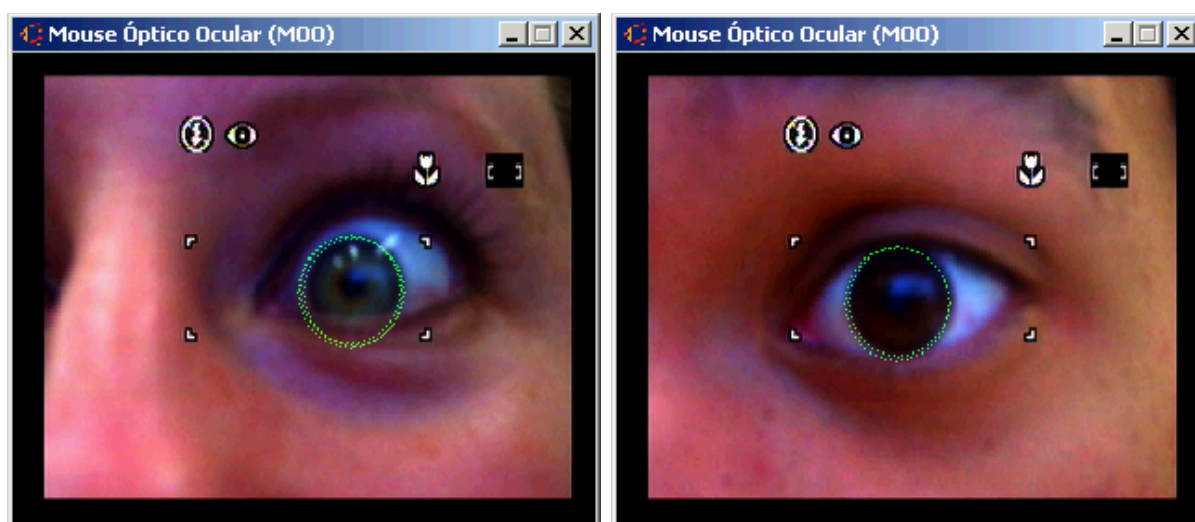


Figura 54: Detecção da íris em imagens da região do olho do usuário capturadas em tempo real

A última etapa no desenvolvimento do Dispositivo de Captura de Vídeo foi configurar o algoritmo de detecção de círculos para trabalhar com o Dispositivo Rastreador montado nessa dissertação. A micro-câmera *wireless* não possui uma qualidade focal similar à câmera digital Sony DSC-P92, e gera certos ruídos que atrapalham o processo de detecção da íris. Porém, a parametrização de Kimme, Ballard e Sklansky [18] segmentou as íris dos usuários, nas imagens capturadas com nosso Dispositivo Rastreador, com bom grau de precisão.

A principal preocupação com o protótipo montado nessa dissertação foi o fator de iluminação ambiente, que influi diretamente na detecção da íris do usuário. Ambientes escuros ou iluminados com lâmpadas fluorescentes foram situações onde o nosso protótipo apresentava elevada taxa de erro. Criamos um filtro de manipulação de Histograma para melhorar o desempenho do algoritmo de detecção da íris em ambientes escuros. Nesse caso, para realçar o contraste da imagem utilizamos uma função de transferência linear. A Figura 55 mostra dois exemplos de testes com o Mouse Óptico Ocular com duas condições de iluminação ambiente.

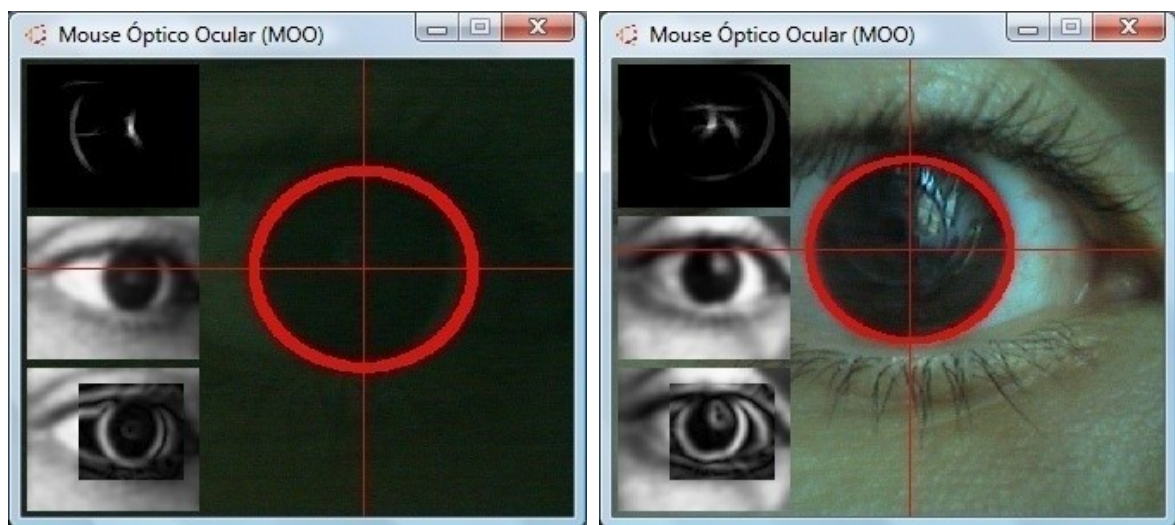


Figura 55: Duas situações de Rastreo Ocular: (a) com má iluminação ambiente e (b) com boa iluminação ambiente

## 5.4 – Sistema de Calibração do Dispositivo Rastreador

O processo de calibração é uma etapa chave que define a precisão do Rastreo Ocular nos Dispositivos Rastreadores. Por esse motivo, o desenvolvimento do sistema de calibração foi uma das etapas mais complexas durante essa dissertação. Com um sistema de calibração mal planejado, dificilmente o Dispositivo Rastreador conseguirá rastrear com êxito toda a área a ser observada pelo usuário. No caso do Rastreo Ocular aplicado a monitores computacionais, algumas regiões do monitor podem apresentar grandes variações na precisão do Rastreo Ocular.

O sistema de calibração desenvolvido nesta dissertação é baseado no modelo de calibração para Dispositivos Rastreadores Fixos na Cabeça. O nosso protótipo é intolerante a movimentos da cabeça. Se um deslocamento ocorrer após o processo de calibração, o Dispositivo Rastreador pode perder todas as configurações obtidas na calibração. Soluções para esse problema já estão sendo adotadas, como a técnica de Rastreo Ocular desenvolvida por Yoo *et al.* [45] que é tolerante a pequenas movimentações da face e o estudo de Coutinho [6] que aprimorou essa técnica obtendo uma maior tolerância aos movimentos da cabeça.

O processo de calibração adotado para o nosso protótipo utiliza o Método dos Mínimos Quadrados para criar a matriz de transformação que determina a posição observada pelo usuário no monitor do computador. Esse método trabalha com cálculos sob matrizes, e para facilitar a sua manipulação utilizamos a biblioteca GMatrix. A GMatrix [4] é uma biblioteca livre usada com alternativa para implementação de matrizes em linguagem C/C++. A biblioteca disponibiliza várias rotinas de cálculos matriciais e álgebra linear que nos auxiliou na criação da matriz de transformação e no cálculo do ponto observado pelo usuário no monitor do computador.

Para realizar a calibração do Mouse Óptico Ocular, dividimos o monitor em 25 regiões para exibir o caractere “X” em vermelho nos seus centros. Utilizamos a biblioteca do OpenGL para criar o ambiente de calibração, com fundo branco e 25 caracteres aparecendo aleatoriamente na tela. Cada caractere permanece visível por 2 segundos e o usuário deve observá-los durante a sua exibição. Esse modelo de calibração não é o ideal, pois causa desconforto ao usuário.

#### 5.4.1 – Método dos Mínimos Quadrados

Também conhecido como Teoria de Aproximação, o Método dos Mínimos Quadrados estuda diversos processos para obter funções lineares que passem o mais próximo possível de um determinado conjunto de pontos. É um dos métodos mais utilizados para resolver esse tipo de problema, também chamado de: Regressão Linear ou Ajuste Linear. No Rastreo Ocular tentamos obter uma função que passe nos pontos ou pelo menos próximos dos pontos  $(x_i, y_i)$  obtidos com a Transformada de Hough. Com o Método dos Mínimos Quadrados de Segunda Ordem, sugerido por Ramanauskas [37], calculamos a parábola de melhor ajuste dos pontos utilizados na etapa de calibração.

O processo de calibração começa com a criação da matriz  $t$ , que irá armazenar os pontos que o usuário deve observar em seu monitor. O tamanho da matriz  $t$  é de  $2 \times 25$ , onde cada coluna armazena a coordenada  $(x_t, y_t)$  do caractere “X” que aparece na tela do usuário. Os

valores armazenados na matriz são preenchidos quando o objeto do processo de calibração é instanciado. As coordenadas são calculadas com base na resolução do monitor onde o sistema é executado, garantindo que o Mouse Óptico Ocular seja compatível com qualquer monitor sem restrições de tipo, tamanho, proporção ou resolução.

A matriz  $v$  armazena o conjunto de coordenadas  $(x_v, y_v)$  da íris do usuário que são obtidas através da imagem capturada no momento em que o este observa o caractere “X” na tela. Para cada coordenada definida na matriz  $t$ , temos a respectiva coordenada da íris do usuário definida na matriz  $v$ . O tamanho da matriz  $v$  também é de  $2 \times 25$  e os seus valores são distribuídos nas colunas conforme o índice do ponto visível na tela.

Após a visualização dos 25 pontos na tela e o mapeamento da matriz  $v$ , criamos a matriz de transformação  $T$  que auxilia o sistema de Rastreo Ocular na estimativa do ponto observado pelo usuário. Primeiramente, temos que criar uma matriz  $S$  para calcular a expressão  $s_{(x,y)} = a_0 + a_1 \cdot x + a_2 \cdot y + a_3 \cdot xy + a_4 \cdot x^2 + a_5 \cdot y^2$  de acordo com os parâmetros armazenados na matriz  $v$ . Como a expressão contém 6 incógnitas, o tamanho da matriz  $S$  é de  $25 \times 6$ . O algoritmo da Figura 56 mostra o processo de criação da matriz  $S$  com a biblioteca GMatrix.

---

```

01 PGMATRIX S;
02 S = PGMATRIX_ALLOC(points, 6);
03 for (int i = 1; i <= points; i++) {
04     PGMATRIX_DATA(S, i, 1) = 1;
05     PGMATRIX_DATA(S, i, 2) = PGMATRIX_DATA(v, 1, i);
06     PGMATRIX_DATA(S, i, 3) = PGMATRIX_DATA(v, 2, i);
07     PGMATRIX_DATA(S, i, 4) = PGMATRIX_DATA(v, 1, i) *
                                PGMATRIX_DATA(v, 2, i);
08     PGMATRIX_DATA(S, i, 5) = PGMATRIX_DATA(v, 1, i) *
                                PGMATRIX_DATA(v, 1, i);
09     PGMATRIX_DATA(S, i, 6) = PGMATRIX_DATA(v, 2, i) *
                                PGMATRIX_DATA(v, 2, i);
10 }

```

---

Figura 56: Criação da matriz  $S$  no Método dos Mínimos Quadrados

Após a criação da matriz  $S$ , é gerada a matriz pseudo-inversa de  $S$  definida por  $S^+$ , de tamanho  $6 \times 25$ . O cálculo da pseudo-inversa sob a matriz  $S$  é realizado de acordo com a equação  $(S^T S)^{-1} S^T$ . Com o uso da biblioteca GMatrix foi mais fácil realizar o cálculo da pseudo-inversa de  $S$ , através de uma rotina específica. O algoritmo da Figura 57 ilustra esse processo.

---

```

01 PGMATRIX sPI;
02 sPI = PGMATRIX_ALLOC(6, points);
03 PGMATRIX matrixAux1;
04 matrixAux1 = PGMATRIX_ALLOC(6, points);
05 PGMATRIX_PSEUDOINVERSE(sPI, s, matrixAux1);

```

---

Figura 57: Criação da matriz pseudo-inversa de  $S$

A matriz  $S^+$  é utilizada para calcular as 12 incógnitas  $a_0, a_1, a_2, a_3, a_4, a_5, b_0, b_1, b_2, b_3, b_4$  e  $b_5$  usadas para estimar o Ponto de Interesse do usuário na tela. Para calcular os seus valores, inicialmente separamos as linhas e as colunas da matriz  $t$  em duas matrizes ( $t_l$  e  $t_c$ ) de tamanho  $25 \times 1$ . Multiplicamos a matriz  $t_l$  com a matriz  $S^+$  para criar uma matriz  $6 \times 1$  com os valores dos coeficientes  $a_0, a_1, a_2, a_3, a_4$  e  $a_5$  e realizamos a mesma operação com as matrizes  $t_c$  e  $S^+$  para gerar outra matriz com os valores dos coeficientes  $b_0, b_1, b_2, b_3, b_4$  e  $b_5$ .

A matriz de transformação  $T$  é criada com a união das matrizes de incógnitas. A matriz  $T$  tem tamanho de  $2 \times 6$  para armazenar todas as 12 incógnitas do Método dos Mínimos Quadrados de Segunda Ordem. Os coeficientes  $a_0, a_1, a_2, a_3, a_4$  e  $a_5$  são armazenados na primeira linha e os coeficientes  $b_0, b_1, b_2, b_3, b_4$  e  $b_5$  armazenados na segunda linha da matriz  $T$ . O cálculo das 12 incógnitas e a criação da matriz de transformação  $T$  é demonstrado na Figura 58.

---

```

01 PGMATRIX tL;
02 tL = PGMATRIX_ALLOC(25, 1);
03 for (int i = 1; i <= 25; i++)
04     PGMATRIX_DATA(tL, i, 1) = PGMATRIX_DATA(t, 1, i);
05 PGMATRIX tC;
06 tC = PGMATRIX_ALLOC(25, 1);
07 for (int i = 1; i <= 25; i++)
08     PGMATRIX_DATA(tC, i, 1) = PGMATRIX_DATA(t, 2, i);
09 PGMATRIX a0a1a2a3a4a5;
10 a0a1a2a3a4a5 = PGMATRIX_ALLOC(6, 1);
11 PGMATRIX_MULTIPLY_COPY(a0a1a2a3a4a5, SPI, tL);
12 PGMATRIX b0b1b2b3b4b5;
13 b0b1b2b3b4b5 = PGMATRIX_ALLOC(6, 1);
14 PGMATRIX_MULTIPLY_COPY(b0b1b2b3b4b5, SPI, tC);
15 for (int i = 1; i <= 6; i++) {
16     PGMATRIX_DATA(T, 1, i) = PGMATRIX_DATA(a0a1a2a3a4a5, i, 1);
17     PGMATRIX_DATA(T, 2, i) = PGMATRIX_DATA(b0b1b2b3b4b5, i, 1);
18 }

```

---

Figura 58: Cálculo das incógnitas utilizadas no Método dos Mínimos Quadrados e a criação da matriz  $T$

O sistema de calibração pode ser estendido para os Métodos dos Mínimos Quadrados de Primeira, Segunda, Terceira e Quarta Ordem. Esses métodos são conhecidos respectivamente como o ajuste da reta, parábola, cúbica e quártica. Durante o desenvolvimento do Mouse Óptico Ocular, implementamos os métodos de Primeira e Segunda Ordem. Seguindo as sugestões de Ramanauskas [37], os melhores resultados do processo de calibração foram obtidos através da Calibração de Segunda Ordem com 25, 9 e 12 pontos respectivamente.

#### 5.4.2 – Cálculo do Ponto Observado Pelo Usuário

A matriz de transformação  $T$  permanece inalterada durante toda execução do Mouse Óptico Ocular. Ela é utilizada para calcular o ponto observado pelo usuário no monitor após o

retorno da coordenada  $(x_i, y_i)$  pela rotina de processamento da imagem capturada. A coordenada  $v'_i = (x'_i, y'_i)$ , para onde o cursor será movido, é obtida através da multiplicação entre a matriz de transformação  $T$  e a matriz de Segunda Ordem com informações da coordenada central da íris.

Após a aplicação da Transformada de Hough na imagem capturada, é repassada à rotina de cálculo do ponto observado, a coordenada  $(x, y)$  da íris do usuário. A matriz  $v_i$  de tamanho de  $6 \times 1$  armazena a representação de Segunda Ordem do centro da íris do usuário e a matriz  $v'_i$  de tamanho de  $2 \times 1$  recebe o ponto observado pelo usuário no monitor através da equação  $v'_i = T \cdot v_i$ . Antes de mover o curso do mouse, verificamos se a coordenada  $(x'_i, y'_i)$  calculada pela rotina pertence a uma área válida do monitor do usuário, conforme ilustrado na Figura 59.

```

01 PGMATRIX vi;
02 vi = PGMATRIX_ALLOC(6, 1);
03 PGMATRIX_DATA(vi, 1, 1) = 1;
04 PGMATRIX_DATA(vi, 2, 1) = x;
05 PGMATRIX_DATA(vi, 3, 1) = y;
06 PGMATRIX_DATA(vi, 4, 1) = x * y;
07 PGMATRIX_DATA(vi, 5, 1) = x * x;
08 PGMATRIX_DATA(vi, 6, 1) = y * y;
09 PGMATRIX v'i;
10 v'i = PGMATRIX_ALLOC(2, 1);
11 PGMATRIX_MULTIPLY_COPY(v'i, T, vi);
12 int x'i = (int) PGMATRIX_DATA(v'i, 1, 1);
13 int y'i = (int) PGMATRIX_DATA(v'i, 2, 1);
14 if ((x'i >= 0) && (x'i <= this->width) &&
      ((y'i >= 0) && (y'i <= this->height)))
15     SetCursorPos(x'i, y'i);

```

Figura 59: Estimativa do ponto observado pelo usuário

## 5.5 – Sistema do Mouse Óptico Ocular

O protótipo desenvolvido nesta dissertação pode ser adaptado para realizar o Rastreo Ocular em diversas situações. Nesta dissertação, o desenvolvimento do Dispositivo Rastreador foi direcionado para um modelo alternativo de Interação Humano-Computador. O sistema tem o objetivo de posicionar o cursor do mouse o mais próximo possível do ponto observado no monitor pelo usuário. Algumas funções básicas dos mouses convencionais foram deixadas de lado, por exemplo: ativar e mover o cursor; selecionar e arrastar objetos. No desenvolvimento de um produto final, é desejável que a maioria das funções básicas sejam adicionadas ao protótipo.

A quantidade de imagens processadas por segundo pode ser configurada conforme o desempenho do computador onde o sistema será executado. Por exemplo, caso a configuração do computador do usuário tenha um desempenho inferior, o sistema pode ser recompilado para processar apenas 1 imagem por segundo. Na etapa de testes com os recursos disponíveis,

utilizamos o maior número possível de imagens processadas, com taxas variando entre 10 e 15 imagens por segundo com o algoritmo de Kimme, Ballard e Sklansky [18].

Em algumas seqüências de imagens capturadas, o centro da íris era detectado com pequenas variações mesmo com o usuário olhando fixamente um determinado ponto na tela. Por esse motivo, foi desenvolvido um filtro para auxiliar no cálculo do centro da íris, onde é possível identificar se houve um movimento ocular ou a alteração foi ocasionada por algum ruído no ambiente. O filtro determina a vizinhança ignorada nas seqüências das imagens processadas e o seu uso pode ser configurado antes do processo de compilação do sistema rastreador.

O clique do mouse é realizado quando o usuário permanece um determinado tempo com os olhos fechados. Quando o sistema processa uma seqüência de  $n$  imagens e não identifica em nenhuma delas a íris do usuário, o algoritmo da Figura 60 é executado realizando o clique simples na região onde o cursor está posicionado. O Mouse Óptico Ocular realiza o clique simples quando o usuário permanecer 2 segundos com os olhos fechados.

---

```

01 INPUT *inputDown = new INPUT();
02 inputDown->type = INPUT_MOUSE;
03 inputDown->mi.dwFlags = MOUSEEVENTF_LEFTDOWN;
04 inputDown->mi.dwExtraInfo = 0;
05 inputDown->mi.mouseData = 0;
06 inputDown->mi.time = 0;
07 int aResult = SendInput(1, inputDown, 28);
08 delete (inputDown);
09 INPUT *inputUp = new INPUT();
10 inputUp->type = INPUT_MOUSE;
11 inputUp->mi.dwFlags = MOUSEEVENTF_LEFTUP;
12 inputUp->mi.dwExtraInfo = 0;
13 inputUp->mi.mouseData = 0;
14 inputUp->mi.time = 0;
15 int bResult = SendInput(1, inputUp, 28);
16 delete (inputUp);

```

---

Figura 60: Emulação do duplo clique do botão esquerdo do mouse

Foram acrescentadas ao sistema do Mouse Óptico Ocular a exibição de informações sobre o processo de detecção da íris com a Transformada de Hough. Na tela de exibição da imagem capturada (Figura 55), foram adicionadas três visualizações com as seguintes informações: espaço de parâmetros bidimensional, imagem em tons de cinza com filtro de manipulação de Histograma e a imagem com o Filtro de Sobel [13] aplicado na área processada com a Transformada de Hough.

No canto superior esquerdo da área de trabalho também são exibidas informações sobre o processamento da imagem capturada, com isso podemos observar em tempo real o número de

círculos detectados, o maior incremento no espaço de parâmetros e a coordenada  $(x, y)$  central da íris na imagem analisada.

A última etapa no desenvolvimento do Mouse Óptico Ocular foi a criação de um sistema de atalhos para os principais *softwares* instalados no computador do usuário. Logo após o processo de calibração, a janela ilustrada na Figura 61 é exibida em primeiro plano na área de trabalho do usuário. Esses atalhos permitem que o usuário possa abrir qualquer um dos programas, observando o respectivo ícone e fechando os olhos por 2 segundos. Todas as vezes que o usuário quiser ver o sistema de atalhos em primeiro plano, ele deve observar o botão extra-monitor criado acima do monitor.



Figura 61: Sistema de atalhos para os principais programas instalados no Sistema Operacional

## 5.6 – Sistema do Teclado Óptico Ocular

Ao final do desenvolvimento dos sistemas de Rastreamento Ocular, notamos a necessidade da criação de um Teclado Óptico Ocular que permitisse ao usuário digitar textos através dos movimentos oculares. Algumas pesquisas de Rastreamento Ocular apresentadas no item 4.5, mostram a união de Dispositivos Rastreadores com ferramentas de acessibilidade como o teclado virtual. Porém, para digitar palavras ou frases o usuário tem um tempo de resposta relativamente longo. A solução adotada foi um modelo de escrita com os olhos baseado na técnica apresentada por Perlin [31], onde substituímos o uso de um *joystick* pelos movimentos oculares do usuário.

Criamos um conjunto com quatro mapeamentos idênticos aos utilizados por Perlin [31] no sistema *Quikwriting*. Os quatro mapeamentos tratam respectivamente: o alfabeto em letras minúsculas; o alfabeto em letras maiúsculas; o conjunto numérico com dígitos decimais e símbolos matemáticos; e um conjunto básico com caracteres especiais da tabela ASCII. Por padrão, o primeiro mapeamento assume o papel principal no sistema. Caso o usuário deseje utilizar outro mapeamento, ele pode escolher por: utilizá-lo para digitar um único caractere; ou torná-lo “fixo” para trabalhar exclusivamente com seus caracteres.

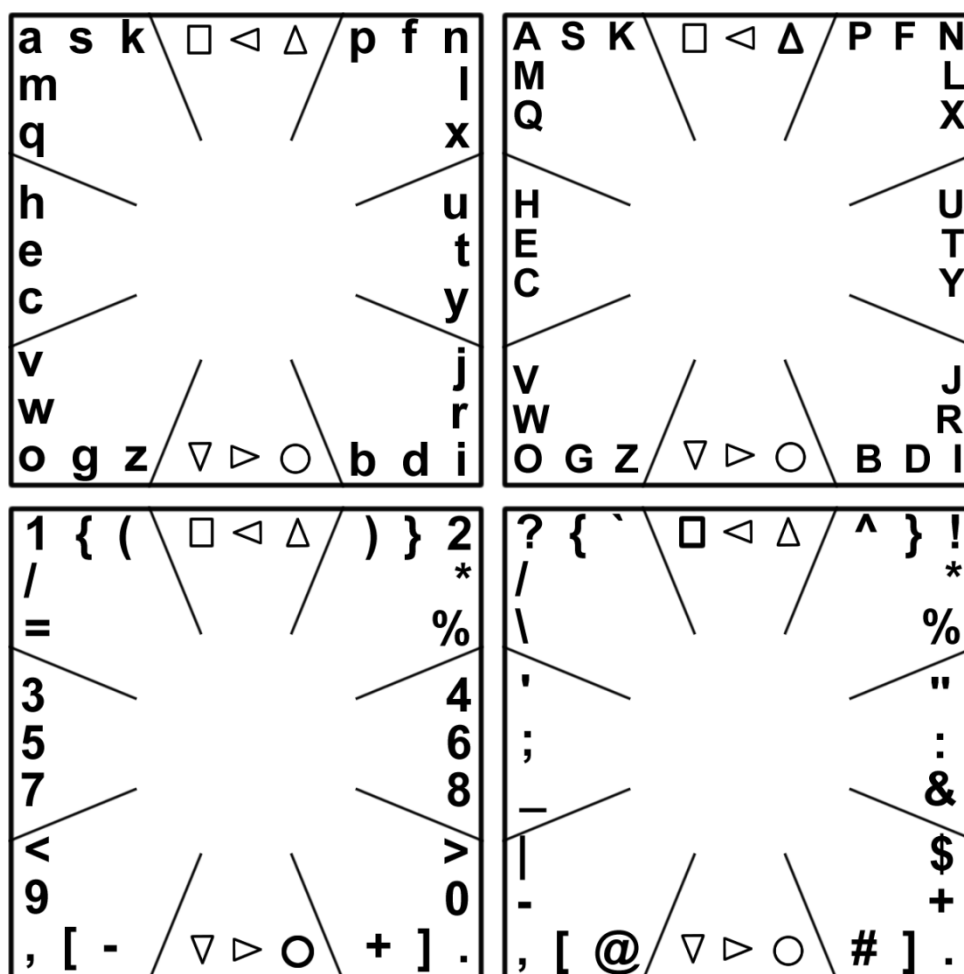


Figura 62: Exemplo dos quatro mapeamentos utilizados pelo Teclado Óptico Ocular

Para incorporar os movimentos oculares na técnica de Perlin [31], o Teclado Óptico Ocular cria nove regiões virtuais baseadas na resolução do monitor, e utiliza essas regiões no processo de escrita com os olhos. As regiões têm tamanhos iguais e são adaptadas automaticamente a qualquer tipo e resolução do monitor utilizado. As regiões virtuais não são visíveis aos olhos do usuário, porém ele precisa conhecer previamente o mapeamento do sistema para mover os olhos nas seqüências dos caracteres que deseja escrever. A Figura 63 ilustra os

identificadores das regiões virtuais em um ambiente de trabalho com resolução de 1280x800.

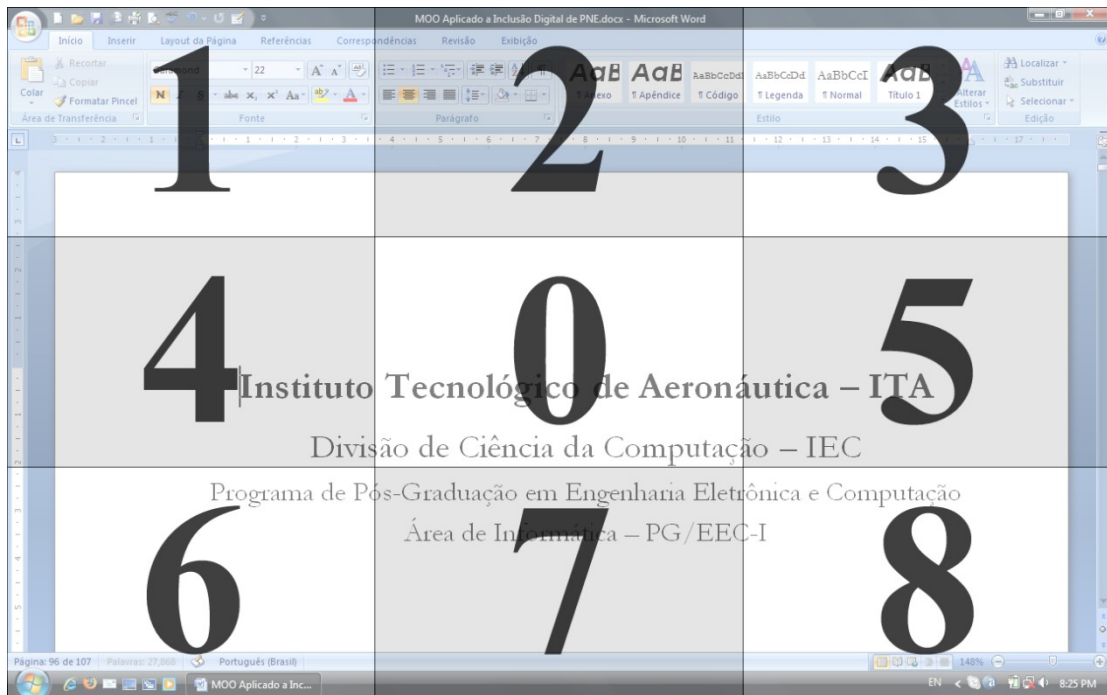


Figura 63: Visualização das nove regiões virtuais utilizadas pelo sistema do Teclado Óptico Ocular

Para o usuário digitar um caractere, ele precisa observar uma das regiões virtuais e esperar que o cursor do mouse mova-se para o centro da região. Baseado nos mapeamentos ilustrados na Figura 62, para o usuário digitar o caractere “c” ele deve: observar primeiramente a região “0”, observar a região “4”, depois a região “6” e voltar para a região “0”. O sistema identifica essa seqüência através do código 0460 e retorna para o usuário o caractere “c”.

Caso deseje mudar a tabela de mapeamento, o usuário deve observar as regiões virtuais nas seguintes seqüências: 0230 para a tabela com caracteres maiúsculos; 0780 para a tabela com caracteres numéricos; e 0210 para a tabela com caracteres especiais. Para tornar uma das tabelas fixas a partir do mapeamento padrão, o usuário deve realizar os movimentos oculares duas vezes seguidas em uma das seqüências apresentadas anteriormente. Incluímos também seqüências para representar as teclas *enter*, *space* e *backspace* de um teclado comum. Para emular respectivamente os cliques dessas teclas, o usuário deve observar as regiões virtuais nas seqüências 0760, 070 e 020.

## 5.7 – Considerações Finais

Nesse capítulo foram apresentados os algoritmos implementados na criação dos sistemas de Rastreo Ocular desenvolvidos nesta dissertação. O capítulo abordou desde o estudo da viabilidade da Transformada de Hough para detecção da íris em imagens digitais, até a

implementação do processo de calibração com o Método dos Mínimos Quadrados de Segunda Ordem utilizado pelo nosso Dispositivo Rastreador.

Utilizamos o ambiente de desenvolvimento Microsoft® Visual C++ 2005 para a implementação dos algoritmos utilizados em nosso protótipo. Para o desenvolvimento do sistema de detecção de círculos em imagens digitais foi utilizada a biblioteca TerraLIB, e para o sistema de detecção da íris em imagens capturadas em tempo real utilizamos a biblioteca OpenCV.

No próximo capítulo são abordados os detalhes da etapa de testes do Mouse Óptico Ocular e do Teclado Óptico Ocular. Também são discutidos os resultados obtidos com o desenvolvimento do nosso protótipo, comparando os nossos resultados com os dados de alguns Dispositivos Rastreadores pesquisados nessa dissertação.

---

## Testes e Resultados

---

**F**inalizada a etapa de desenvolvimento dos sistemas de Rastreo Ocular, iniciamos a avaliação de desempenho do nosso protótipo com os sistemas Mouse Óptico Ocular e Teclado Óptico Ocular. Foram realizados testes superficiais sem a preocupação de analisar aspectos de ergonomia ou de usabilidade, pois tais aspectos fogem do escopo do nosso trabalho. Além de avaliar o desempenho do protótipo, os testes mostraram as contribuições obtidas com essa dissertação para a área de pesquisa sobre Rastreo Ocular.

Os dados analisados foram obtidos manualmente e automaticamente nos testes do protótipo com alguns voluntários. No total selecionamos seis voluntários para avaliar o Rastreo Ocular de usuários com as seguintes características: olhos de diferentes cores; uso de lentes de contato; e uso de óculos. Os voluntários que usam óculos no dia-a-dia reclamaram do desconforto ao utilizar o protótipo sobre os óculos. Eliminamos os testes dos voluntários com óculos devido ao desconforto e aos reflexos de luminosidades produzidos nas lentes que reduzia o número de círculos detectados nas imagens capturadas.

Durante a etapa de testes, procurou-se utilizar o Dispositivo Rastreador em situações adversas como: ambientes com boa e má iluminação; iluminação com lâmpadas fluorescentes e incandescentes; e o auxílio de luminárias. Na primeira etapa de testes analisamos a precisão do Rastreo Ocular do nosso Dispositivo Rastreador. Na segunda etapa, avaliamos o uso do Mouse Óptico Ocular integrado ao sistema de atalhos. Por último, realizamos testes com o Teclado Óptico Ocular limitado a escrita de palavras e frases do Português Brasileiro.

Os testes foram realizados em um computador do tipo *notebook*, com processador Intel Dual Core 1.60 GHz com memória de 1 GB. Em metade dos testes, utilizamos o monitor padrão do *notebook* de 15” em formato *widescreen*; e na outra metade utilizamos um monitor de 15” em

formato 3:4, interligado ao *notebook*. Utilizando o algoritmo de Kimme, Ballard e Sklansky [18], o tempo médio de processamento é de aproximadamente 100 milissegundos para cada imagem capturada. Por padrão as imagens capturadas têm um tamanho de  $520 \times 480$  *pixels*, porém para um menor esforço computacional redimensionamos as imagens processadas para  $200 \times 184$  *pixels*.

## 6.1 – Diretivas Para o Uso do Protótipo

Definimos um padrão de posicionamento em relação ao usuário e o monitor, que foi utilizado por todos os voluntários, a fim de minimizar distorções entre os seus testes. O padrão de posicionamento usa um sistema cartesiano tridimensional  $(x, y, z)$ , onde os eixos  $x$  e  $y$  definem respectivamente as posições horizontal e vertical no monitor, e o eixo  $z$  define a distância entre o usuário e o monitor.

A Figura 64 ilustra o padrão de posicionamento adotado em nossos testes, com o usuário posicionado ao eixo central do monitor. O retângulo no plano  $xy$  define a tela do monitor em formato *widescreen*, onde os pontos  $A$  e  $B$  definem o tamanho da tela em *pixels*. O ponto  $C$  é ponto central do monitor que define a posição do usuário à frente da tela. O ponto  $D$  está alinhado ao eixo central do monitor a uma distância de 1890 *pixels*, que corresponde a aproximadamente 50 centímetros de distância entre o usuário e o monitor.

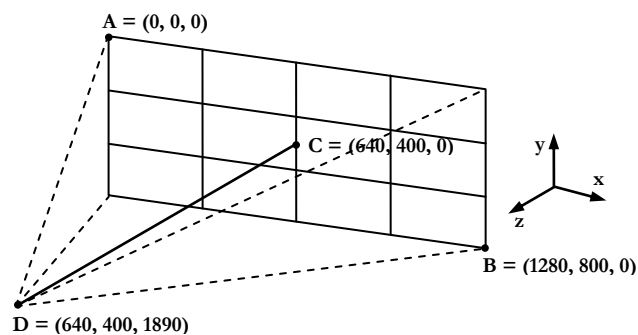


Figura 64: Padrão de posicionamento entre o usuário e o monitor

Durante o desenvolvimento do Mouse Óptico Ocular, utilizávamos o protótipo para verificar se a íris continuava sendo detectada conforme as alterações no sistema. Observamos que a região direita do monitor apresentava uma melhor precisão no Rastreamento Ocular do que a região esquerda. A principal explicação para este fato é a posição da câmera de aquisição de vídeo em um ângulo de aproximadamente  $30^\circ$  no visor direito dos óculos de proteção industrial. Por esse motivo, solicitamos aos voluntários que se posicionasse em frente do monitor, a uma distância de 50 centímetros com o olho direito alinhado ao eixo central da tela (ponto  $D$ ).

Para manter a qualidade do processo de calibração, o ideal é manter a cabeça e o pescoço dos usuários imóveis durante o uso do Dispositivo Rastreador. Barcelos [2] utilizou em seu trabalho um apoiador de queixo construído especialmente para a condução dos seus experimentos. No nosso caso, não utilizamos nenhum apoio para a cabeça do usuário e, por conseqüência, informações arbitrárias foram geradas nos dados obtidos em nossos testes. Para minimizar a fadiga causada pela imobilidade do pescoço e da cabeça, solicitamos aos usuários que realizassem pausas entre os testes sempre que desejassem.

## 6.2 – Análise da Precisão do Dispositivo Rastreador

A análise da precisão teve o objetivo de avaliar a taxa de erro apresentada por nosso Dispositivo Rastreador após o processo de calibração. Nos estudos de Rastreo Ocular, a taxa de erros é representada pela diferença do ponto observado pelo usuário e o ponto estimado pelo Dispositivo Rastreador. Essa representação é definida sob graus ( $^{\circ}$ ) onde, a uma distância de 50 centímetros entre o usuário e o monitor, cada grau corresponde a aproximadamente 10 milímetros de distância entre o ponto real e o ponto estimado no plano.

Além de avaliar a taxa de erros do protótipo, procuramos detectar a influência dos movimentos da cabeça nos eixos  $x$ ,  $y$  e  $z$  após o processo de calibração. Os testes foram realizados com o voluntário movendo a cabeça alguns milímetros em uma das coordenadas após a calibração. Foram utilizados dois monitores com formatos distintos, onde os voluntários realizaram os testes de precisão uma vez em cada monitor.

Os testes de precisão realizados nessa dissertação são similares ao modelo adotado por Barcelos [2] nos seus estudos. A diferença é que Barcelos [2] utilizou imagens sintéticas dos olhos humanos criadas com a técnica *ray tracing*, e nessa dissertação utilizamos imagens capturadas em tempo real com o nosso Dispositivo Rastreador. Por esse motivo, optamos por não comparar os nossos dados com as informações obtidas por Barcelos [2]. Comparamos a precisão do nosso protótipo com as informações das pesquisas de Yoo e Chung [46] e Coutinho [6] cujo taxa de erros média é de respectivamente  $2.37^{\circ}$  e  $0.91^{\circ}$ .

Coutinho [6] contou com a participação de cinco voluntários nos testes da sua pesquisa. Cada voluntário executou os testes do Dispositivo Rastreador por três vezes utilizando o método desenvolvido por Yoo e Chung [46] e o aprimoramento dessa técnica criada pelo próprio Coutinho [6]. Os voluntários 1c e 3c dos testes de Coutinho [6] utilizavam, respectivamente, lentes de contato e óculos. As principais características individuais dos nossos voluntários são: o

voluntário 3f tem os olhos verdes; e o voluntário 5f utilizou lentes de contato sem a especialidade de alteração da pigmentação da íris.

Seguimos as mesmas seqüências dos testes realizados por Coutinho [6], a diferença é que utilizamos dois tipos de monitores. O monitor em formato *widescreen* foi dividido em 12 regiões virtuais ( $3 \times 4$ ) e o monitor em formato normal dividido em 9 regiões virtuais ( $3 \times 3$ ). Para calcular a taxa de erros, cada região dos monitores foram subdivididas em uma matriz de tamanho  $2 \times 2$ . Desse modo, durante os testes de precisão o usuário observou 48 pontos no monitor em formato *widescreen* e 36 pontos no monitor em formato normal. A Figura 65 ilustra as regiões virtuais criadas nos monitores e os pontos observados pelos usuários.

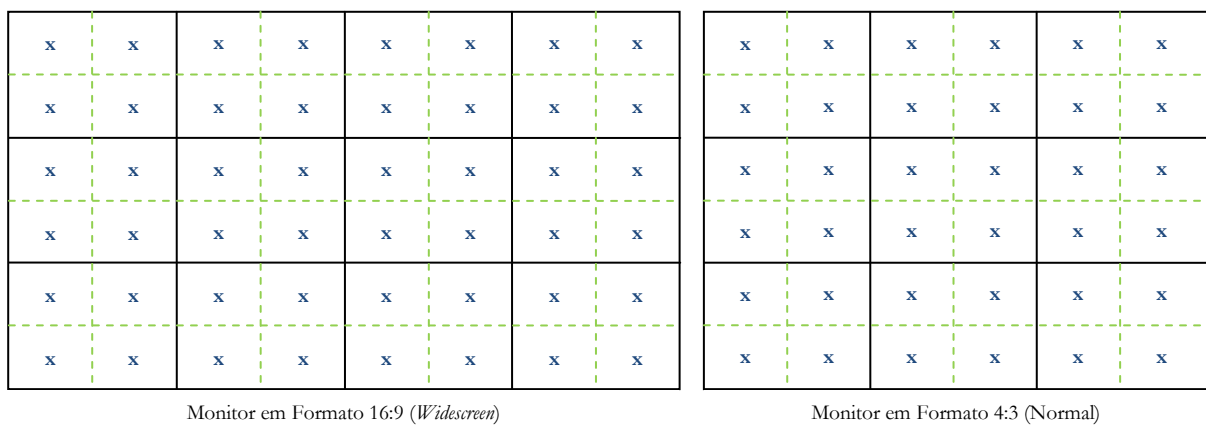


Figura 65: Regiões virtuais dos monitores utilizados nos testes de precisão do Dispositivo Rastreador

Para calcular a distância entre dois pontos no monitor, utilizamos a Equação 2.4, onde  $(x_e, y_e)$  representa o ponto observado pelo usuário e  $(x_m, y_m)$  o ponto estimado pelo Dispositivo Rastreador. Os pontos na tela são representados pelo Sistema Operacional através de valores em *pixels*, sendo que cada centímetro da tela tem aproximadamente 38 *pixels*. Para padronizar os nossos testes com as demais pesquisas, convertemos os valores obtidos para centímetros multiplicando o valor *error* por 0.0264179104477611.

Coutinho [6] separou o teste de precisão em duas etapas: a primeira execução realizada logo após o processo de calibração; e as demais realizadas em seqüência sem a necessidade de uma nova etapa de calibração. Durante a fase de desenvolvimento, o nosso protótipo mostrou-se intolerante a pequenas variações da cabeça. Por esse motivo, solicitamos aos nossos voluntários que tentassem permanecer com a cabeça imóvel durante as três execuções do teste de precisão. A Tabela 3 mostra a média dos resultados obtidos com os dois formatos de monitores utilizados para calcular a precisão do nosso Dispositivo Rastreador.

Tabela 3: Resultados obtidos para cada voluntário nos testes da precisão do Dispositivo Rastreador

Voluntário	Método de Yoo e Chung [46]				Método de Coutinho [6]				Voluntário	Mouse Óptico Ocular			
	Erro Médio		Desvio Padrão		Erro Médio		Desvio Padrão			Erro Médio		Desvio Padrão	
	1ª	2ª e 3ª	1ª	2ª e 3ª	1ª	2ª e 3ª	1ª	2ª e 3ª		1ª	2ª e 3ª	1ª	2ª e 3ª
1c	1.51 cm	1.98 cm	0.69 cm	0.96 cm	1.23 cm	1.15 cm	1.39 cm	0.96 cm	1f	1.71 cm	1.78 cm	0.94 cm	0.95 cm
2c	2.51 cm	2.53 cm	0.82 cm	1.03 cm	0.73 cm	0.98 cm	0.29 cm	0.62 cm	2f	1.92 cm	1.93 cm	0.91 cm	1.13 cm
3c	1.75 cm	1.80 cm	0.48 cm	0.59 cm	0.73 cm	0.90 cm	0.39 cm	0.55 cm	3f	1.69 cm	2.00 cm	0.91 cm	1.02 cm
4c	3.15 cm	2.95 cm	1.13 cm	1.15 cm	0.88 cm	0.98 cm	0.48 cm	0.62 cm	4f	1.69 cm	1.94 cm	1.04 cm	1.18 cm
5c	2.98 cm	3.36 cm	0.86 cm	1.11 cm	0.78 cm	0.93 cm	0.43 cm	0.51 cm	5f	1.46 cm	2.01 cm	0.71 cm	1.12 cm
									6f	1.67 cm	1.95 cm	0.84 cm	0.90 cm
m	2.45 cm		$\sigma = 0.88$ cm		0.93 cm		$\sigma = 0.62$		m	1.81 cm		$\sigma = 0.97$ cm	

Observamos que o Dispositivo Rastreador perdia a precisão obtida com a calibração no decorrer do seu uso. A perda era mínima, porém com o uso prolongado (em torno de 10 minutos) era necessário realizar uma nova etapa de calibração. As maiores variações na precisão do protótipo foram apresentadas nos testes executados pelos voluntários 3f, 4f, 5f e 6f. Apesar do voluntário 5f ter obtido a melhor taxa de precisão (1.46 centímetros), também apresentou a pior taxa dentre os demais testes (2.01 centímetros). Os testes do nosso Dispositivo Rastreador apresentaram um desvio padrão maior do que os desvios apresentados por Yoo e Chung [46] e Coutinho [6]. Esse aumento era esperado, justamente pela falta de um apoiador de queixo para minimizar a influência dos movimentos involuntários da cabeça na taxa de precisão.

A taxa de erros apresentada por nosso protótipo (1.81 centímetros) foi quase o dobro em comparação com o modelo criado por Coutinho [6]. Em comparação com o modelo de Yoo e Chung [46], o nosso Dispositivo Rastreador apresentou uma melhora de 26%. É extremamente difícil obter uma boa precisão no Rastreo Ocular, utilizando um dispositivo fixo na cabeça sem um processo que determine uma relação de distância entre o monitor e o Dispositivo Rastreador. No atual estágio de desenvolvimento do nosso protótipo, a variação na precisão do Rastreo Ocular apresentada no Mouse Óptico Ocular foi de aproximadamente  $1.75^\circ$  no campo de visão.

Após os testes de precisão do protótipo, iniciamos uma etapa de testes para verificar a influência dos movimentos da cabeça no processo de calibração. Para isso, solicitamos aos voluntários quatro novos testes para cada um dos eixos  $x$ ,  $y$  e  $z$ . Após o processo de calibração, com o olho direito posicionado ao eixo central da tela, o voluntário reposicionava o olho a uma distância média de 50 milímetros. Da mesma forma que os testes de precisão, utilizamos os dois tipos de monitores (*widescreen* e *normal*) analisando respectivamente 48 e 36 pontos na tela.

As variações na taxa de erros foram similares nos testes dos dois tipos de monitores. Por esse motivo, as tabelas apresentadas nesse capítulo mostram apenas os resultados dos testes

utilizando o monitor em formato *widescreen*. Informações detalhadas sobre os testes com o monitor em formato normal estão disponíveis no Anexo III. Em todos os testes, a distribuição da taxa de erros não foi uniforme nas regiões virtuais analisadas. Esse fato é observado na Tabela 4, que mostra os resultados da avaliação dos movimentos da cabeça em torno do eixo x.

Tabela 4: Diferença da precisão com suaves movimentos sob o eixo x

Voluntário	12 regiões virtuais no monitor em formato “widescreen”											
	1 <sup>a</sup>	2 <sup>a</sup>	3 <sup>a</sup>	4 <sup>a</sup>	5 <sup>a</sup>	6 <sup>a</sup>	7 <sup>a</sup>	8 <sup>a</sup>	9 <sup>a</sup>	10 <sup>a</sup>	11 <sup>a</sup>	12 <sup>a</sup>
1f	6.92	4.87	2.84	3.91	4.61	3.11	4.35	3.71	3.49	11.3	3.99	10.22
2f	5.03	4.00	1.80	2.84	1.66	3.02	3.76	3.77	5.07	2.50	4.43	2.79
3f	3.60	4.05	2.86	2.96	3.03	5.27	4.29	4.39	6.22	7.28	6.69	9.06
4f	4.85	4.15	4.35	5.38	2.68	3.09	1.26	3.98	6.19	5.73	2.44	3.48
5f	2.97	2.50	3.46	4.70	3.70	4.46	3.41	3.41	4.75	3.34	5.83	5.89
6f	3.94	4.08	2.51	3.57	4.93	3.66	2.72	1.95	13.92	3.33	1.37	2.25
média	4.28 centímetros $\sigma = 2.14$											

Com o olho direito do voluntário a cerca de 50 milímetros à direita no eixo x, observamos uma grande variação na taxa de erros em comparação com o olho alinhado ao eixo central do monitor. O mesmo fato ocorreu com o posicionamento do usuário em 50 milímetros à esquerda no eixo x. A diferença média na taxa de erros foi de 4.28 centímetros, que corresponde a uma adição de 2.40° na taxa de erros do Dispositivo Rastreador. A Tabela 5 mostra os valores obtidos com os movimentos da cabeça em torno do eixo y.

Tabela 5: Diferença da precisão com suaves movimentos sob o eixo y

Voluntário	12 regiões virtuais no monitor em formato “widescreen”											
	1 <sup>a</sup>	2 <sup>a</sup>	3 <sup>a</sup>	4 <sup>a</sup>	5 <sup>a</sup>	6 <sup>a</sup>	7 <sup>a</sup>	8 <sup>a</sup>	9 <sup>a</sup>	10 <sup>a</sup>	11 <sup>a</sup>	12 <sup>a</sup>
1f	5.76	5.03	5.08	3.15	4.58	4.14	2.20	2.97	6.65	5.73	8.16	6.06
2f	4.60	2.99	4.27	3.77	5.03	3.26	3.62	5.30	4.58	5.31	3.52	3.40
3f	2.74	3.19	3.20	3.28	4.84	4.87	3.92	3.21	5.48	3.65	6.03	4.00
4f	5.73	3.43	3.05	3.53	3.35	4.70	3.69	2.87	4.64	4.79	5.43	4.12
5f	2.10	1.74	1.04	4.76	2.40	3.28	1.13	4.00	4.75	3.07	3.81	4.01
6f	10.99	6.64	4.78	7.61	6.22	7.86	3.17	5.48	4.75	2.93	9.52	5.40
média	4.42 centímetros $\sigma = 1.74$											

De acordo com os dados obtidos com os movimentos da cabeça em torno do eixo y, a variação na taxa de erros foi um pouco maior do que a apresentada nos testes em torno do eixo x. Nesse caso, a diferença média foi de 4.42 centímetros. Os movimentos em torno dos eixos x e y são os mais comuns de ocorrer, porém os movimentos laterais da cabeça afetaram menos a calibração do protótipo do que os movimentos para cima ou para baixo.

A nossa maior preocupação era os movimentos em torno do eixo z, com a aproximação ou distanciamento do usuário em relação ao monitor. Para calcular a influência desses movimentos no sistema de calibração, em cada teste solicitamos aos voluntários que se posicionassem a uma distância de 45 e 55 centímetros do monitor. A Tabela 6 mostra as médias dos resultados obtidos com os testes nos dois posicionamentos em torno do eixo z.

Tabela 6: Diferença da precisão com suaves movimentos sob o eixo z

Voluntário	12 regiões virtuais no monitor em formato “widescreen”											
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª	10ª	11ª	12ª
1f	6.12	6.96	7.69	6.92	7.15	4.19	5.20	8.10	8.92	3.34	4.35	8.42
2f	19.08	7.61	4.98	15.84	8.45	6.21	5.06	9.45	12.93	6.95	5.13	10.63
3f	13.10	4.31	3.71	13.12	4.06	4.03	5.19	7.61	3.09	5.36	7.01	11.09
4f	7.79	6.81	2.97	6.84	5.12	6.09	8.49	6.92	9.41	8.71	5.61	7.56
5f	7.06	7.06	7.52	6.72	21.00	4.17	2.98	5.05	7.53	4.10	2.94	4.34
6f	3.59	3.88	2.99	3.48	3.92	4.16	2.23	4.09	6.24	4.23	3.83	7.83
média	6.73 centímetros $\sigma = 3.53$											

A Tabela 6 confirma o aumento no erro médio com os movimentos em torno do eixo z, apresentando uma média de 6.73 centímetros. Dentre as três coordenadas analisadas, a movimentação do olho direito no eixo z foi a mais sensível de todas. Com alguns milímetros de aproximação do usuário ao monitor a calibração perde o seu efeito, sendo necessário um novo processo de calibração para que o Dispositivo Rastreador volte a funcionar corretamente.

A área de pesquisa sobre Rastreamento Ocular carece de novas técnicas para aprimorar o processo de calibração dos dispositivos que trabalham com Foto-Oculografia e Video-Oculografia. Alguns avanços já foram alcançados, como nos estudos de Yoo e Chung [46] e Coutinho [6], que criaram rastreadores tolerantes aos movimentos da cabeça. No caso do nosso Dispositivo Rastreador, ele é indicado apenas para Portadores de Necessidades Especiais com lesões medulares que impossibilitem o movimento do pescoço.

### 6.3 – Análise do Funcionamento do Mouse Óptico Ocular

Os testes de funcionamento do Mouse Óptico Ocular foram os mais simples entre os executados pelos nossos voluntários. Nesses testes não avaliamos o desempenho do sistema Mouse Óptico Ocular. Definimos tarefas simples para serem executadas com o uso do nosso protótipo e o sistema de atalhos, para verificar a quantidade de erros que os voluntários cometeram e o tempo de execução dessas tarefas. Com esses testes verificamos a familiaridade do sistema pelos nossos voluntários e os aprimoramentos necessários no Dispositivo Rastreador.

A seqüência de tarefas executadas foi a mesma para os seis voluntários. O objetivo desses testes era avaliar o sistema rastreador sob os seguintes aspectos: o movimento do cursor do mouse com os olhos; a simulação do clique simples do mouse com o piscar dos olhos nos botões do sistema de atalhos; e uso de botões extra-monitor para maximização do sistema de atalhos. Configuramos a janela do sistema de atalhos para abrir automaticamente após o processo de calibração, e depois da sua visualização o voluntário iniciava a etapa de testes.

As seis tarefas executadas por nossos voluntários seguiram a seguinte seqüência: 1. Mover o cursor do mouse sob os botões do Internet Explorer®, Power Point® e Windows Media Player®; 2. Abrir o Bloco de Notas; 3. Maximizar a janela do sistema de atalhos; 4. Abrir o jogo Paciência; 5. Maximizar novamente a janela do sistema de atalhos; 6. Mover o curso do mouse, dessa vez sob os botões do Skype®, Word® e Windows Live Messenger®.

Consideramos o teste como “concluído” no momento em que o voluntário finalizava a sexta tarefa. Alguns voluntários tiveram que inicializar os testes várias vezes. Dentre os seis voluntários, o voluntário 5f foi o que teve maior dificuldade para realizar as tarefas, conseguindo concluir o teste na terceira tentativa. A Tabela 7 mostra o número de tentativas dos voluntários até conseguir concluir o teste, o tempo de execução e a quantidade de erros no teste concluído.

Tabela 7: Informações adquiridas durante o teste de funcionamento do Mouse Óptico Ocular

Voluntário	Dados do teste considerado “concluído”		
	Tempo de Execução	Quantidade de Erros	Número de Tentativas
1f	01:23	4	1
2f	01:06	3	1
3f	01:03	1	1
4f	01:43	3	2
5f	02:08	5	3
6f	01:51	2	2
média	01:32 $\sigma = 00:25$	3 erros $\sigma = 1.41$	1.66 tentativas $\sigma = 0.82$

Os usuários foram unânimes em relatar o estresse causado pelo uso do Mouse Óptico Ocular. A maior reclamação era em relação ao modelo adotado para simular o clique do botão do mouse. Os movimentos do cursor também causavam certa irritação aos voluntários, devido a diferença de até 1.81 centímetros entre as posições observada e a estimada.

## 6.4 – Análise de Desempenho do Teclado Óptico Ocular

Para testar a eficácia do modelo adotado (*Quikwriting*), realizamos testes com a escrita de

palavras e frases do Português Brasileiro. Essa limitação deve-se à dificuldade que os indivíduos têm na escrita e memorização de palavras de línguas estrangeiras. Por exemplo, a comparação entre os testes do nosso sistema de escrita ocular com o teclado ocular de ideogramas chineses, desenvolvido por Wang, Zhai e Su [48], poderia gerar informações incoerentes, principalmente na checagem do tempo médio gasto na tentativa de escrever algumas palavras do idioma chinês.

Por esse motivo, a análise de desempenho do Teclado Óptico Ocular foi realizada em comparação com à pesquisa de Souza *et al.* [42]. As principais diferenças em comparação com nossa dissertação é que: Souza *et al.* [42] utilizaram a ferramenta de acessibilidade teclado virtual para a escrita de palavras e frases do Português Brasileiro; e seus testes foram realizados em quatro voluntários com diferentes tipos de lesões medulares. Os voluntários da pesquisa de Souza *et al.* [42] apresentavam diferentes níveis de dificuldades para mover os seus membros superiores, diferentemente dos nossos voluntários que não possuem nenhum tipo de necessidade especial.

Cinco dias antes dos testes com o Teclado Óptico Ocular, encaminhamos para os nossos voluntários os mapeamentos do sistema para que eles estudassem as seqüências adotadas. Para os voluntários se adaptarem ao nosso Dispositivo Rastreador, solicitamos que escrevessem a palavra ITA (com caracteres minúsculos). Após essa fase de adaptação, os testes foram iniciados.

Analizamos o tempo médio que os nossos voluntários levaram para escrever três vezes consecutivas as seguintes palavras e frases: “casa”, “faqueiro”, “quero água” e “hoje o dia está lindo”. Essas escritas são as mesmas utilizadas por Souza *et al.* [42] na sua pesquisa. A Tabela 8 mostra os resultados obtidos nos testes realizados por Souza *et al.* [42] e uma comparação com os resultados obtidos por nossos voluntários na escrita das palavras “casa” e “faqueiro”.

Tabela 8: Tempo médio (tm) e tempo gasto (mm:ss) nas tentativas de escrever as palavras “casa” e “faqueiro”

Voluntário	Método de Souza <i>et al.</i> [42]						Voluntário	Teclado Óptico Ocular					
	“casa”			“faqueiro”				“casa”			“faqueiro”		
	1º teste	2º teste	3º teste	1º teste	2º teste	3º teste		1º teste	2º teste	3º teste	1º teste	2º teste	3º teste
1s	00:31	00:29	00:25	01:05	01:17	01:30	1f	00:12	00:09	00:10	00:33	00:24	00:21
2s	00:41	00:32	00:35	01:10	01:24	01:31	2f	00:13	00:11	00:16	00:42	00:35	00:25
3s	00:49	00:50	00:47	01:23	01:22	01:12	3f	00:10	00:11	00:08	00:22	00:24	00:22
4s	00:34	00:21	00:23	01:12	01:29	01:26	4f	00:14	00:11	00:12	00:26	00:31	00:29
							5f	00:19	00:13	00:12	00:24	0:26	00:22
							6f	00:15	00:13	00:10	00:38	00:25	00:34
tm	00:34 $\sigma=00:10$			01:20 $\sigma=00:09$			tm	00:12 $\sigma=00:03$			00:27 $\sigma=00:06$		

Os voluntários que usaram o teclado virtual na pesquisa de Souza *et al.* [42], levaram em média 09 segundos para escrever cada letra das palavras “casa” e “faqueiro”. O tempo médio é formado pelo tempo gasto para posicionar o cursor do mouse na letra a ser digitada, e o tempo

que o usuário fica com os olhos fechados para realizar o clique nos botões do teclado virtual. Segundo Souza *et al.* [42], para mover o cursor do mouse de um lado ao outro do teclado virtual (da letra Q para a letra P) os voluntários levavam em média 33 segundos para executar essa ação.

Nos nossos testes, observamos um tempo relativamente menor na escrita das mesmas palavras da pesquisa de Souza *et al.* [42]. O tempo médio apresentado na escrita dos caracteres das palavras “casa” e “faqueiro” foi de 3 segundos. O tempo gasto foi menor com o nosso sistema, pois eliminamos o requisito de posicionamento do cursor em pequenas regiões (botões do teclado virtual) do monitor. Em comparação com a pesquisa de Souza *et al.* [42], observamos que os nossos voluntários escreveram as mesmas palavras cerca de três vezes mais rápido.

Através dos testes demonstrados na Tabela 8, deduzimos que a escrita de frases curtas ou longas com o Teclado Óptico Ocular iria diminuir relativamente o tempo de resposta do usuário nos sistemas que necessitam de entrada de texto. Com o modelo baseado no sistema *Quikwriting*, o estresse do usuário ao utilizar um editor de texto ou um sistema de mensagens instantâneas é menor do que a escrita com o teclado virtual. Para confirmar essa hipótese, a Tabela 9 mostra os dados e a comparação dos nossos testes, com os dados obtidos por Souza *et al.* [42] durante a escrita das frases “quero água” e “hoje o dia está lindo”.

Tabela 9: Tempo médio (tm) e tempo gasto (mm:ss) nas tentativas de escrever as frases “quero água” e “hoje o dia está lindo”

Voluntário	Método de Souza <i>et al.</i> [42]						Voluntário	Teclado Óptico Ocular					
	“quero água”			“hoje o dia está lindo”				“quero água”			“hoje o dia está lindo”		
	1º teste	2º teste	3º teste	1º teste	2º teste	3º teste		1º teste	2º teste	3º teste	1º teste	2º teste	3º teste
1s	01:57	02:07	01:49	04:03	03:14	03:01	1f	00:27	00:29	00:27	01:11	00:56	01:34
2s	02:01	02:14	01:50	03:57	04:10	03:20	2f	00:48	00:41	00:38	01:17	01:19	01:45
3s	02:39	02:22	02:10	04:23	04:30	04:15	3f	00:34	00:36	00:37	01:25	01:32	01:13
4s	02:10	01:27	01:30	04:10	03:50	03:31	4f	00:34	00:47	00:37	01:56	01:22	01:30
							5f	00:27	00:39	00:46	02:29	01:53	01:42
							6f	00:33	00:45	00:38	01:35	01:54	01:46
tm	02:01 $\sigma=00:21$			03:52 $\sigma=00:29$			tm	00:37 $\sigma=00:06$			01:34 $\sigma=00:21$		

Para escrever a frase “quero água”, os nossos voluntários levaram em média 37 segundos. Em relação aos testes de Souza *et al.* [42], esse tempo representa um ganho de 327%. Na escrita da frase “hoje o dia está lindo” o ganho do nosso sistema foi de 246%, onde nossos voluntários realizaram a tarefa em 1min34seg e os voluntários da pesquisa de Souza *et al.* [42] levaram em média 03min52seg. Em nossos testes, o processo de escrita com o Teclado Óptico Ocular mostrou-se mais eficaz em comparação ao modelo escrita com a ferramenta de acessibilidade teclado virtual, adotado por Souza *et al.* [42].

A diferença no tempo médio apresentada dos testes dos voluntários 1f, 2f e 3f em comparação aos testes dos voluntários 4f, 5f e 6f foi relativamente menor. Isso ocorreu devido à maior familiaridade com o Dispositivo Rastreador apresentada pelos três primeiros voluntários. Os três primeiros voluntários estavam mais familiarizados com o Dispositivo Rastreador, pois eles tiveram uma participação mais efetiva no desenvolvimento do protótipo e dos sistemas rastreadores do que os demais voluntários.

Ao final da etapa de testes do sistema Teclado Óptico Ocular, concluímos que o modelo adotado com base no sistema *Quikwriting* é ideal para ser utilizado como um processo de Inclusão Digital de Portadores de Necessidades Especiais. No caso desse sistema, os voluntários não reclamaram de estresse causado pelo uso prolongado do Dispositivo Rastreador. O único detalhe abordado pelo voluntário 5f é a necessidade de incluir uma seqüência nos mapeamentos do sistema para ativar e desativar o Dispositivo Rastreador quando houver necessidade.

## 6.5 – Considerações Finais

Nesse capítulo foram abordados os testes realizados com seis voluntários, com objetivo de avaliar o desempenho do nosso Dispositivo Rastreador com os sistemas Mouse Óptico Ocular e Teclado Óptico Ocular.

A partir da análise dos dados obtidos nos testes apresentados nesse capítulo é possível obter as conclusões dessa dissertação, que são apresentadas no próximo capítulo, juntamente com as propostas de trabalhos futuros para o aprimoramento do nosso protótipo.

---

## Conclusões

---

**E**ssa dissertação apresentou o desenvolvimento de um Dispositivo Rastreador voltado à Inclusão Digital de Portadores de Necessidades Especiais. A finalidade do dispositivo desenvolvido é permitir a interação do usuário com o computador através dos seus movimentos oculares. Caso o Portador de Necessidades Especiais apresente alguma deficiência física que afete a mobilidade dos membros superiores e do pescoço, o dispositivo permite a sua interação com o computador, substituindo os mecanismos de entrada padrão do computador por seus movimentos oculares.

A partir da análise dos testes realizados com o nosso Dispositivo Rastreador, concluímos que os olhos podem ser usados como uma alternativa de interação entre o usuário e o computador. Apesar de algumas restrições e necessidades de aprimoramentos, o protótipo criado permite a substituição dos dispositivos de entrada padrão (mouse e teclado) do computador pelos movimentos oculares. No atual estágio de desenvolvimento, o público alvo do nosso Dispositivo Rastreador são portadores de tetraplegia. Porém, qualquer indivíduo pode utilizar o dispositivo desde que mantenha a cabeça imóvel durante o seu uso.

No desenvolvimento do nosso Dispositivo de Rastreador, enfrentamos um sério problema em relação à precisão vertical desse dispositivo. Através de testes realizados com a Transformada de Hough concluímos que o seu uso era viável para detectar e localizar a íris em imagens digitais. Mesmo nas imagens capturadas com altos índices de ruídos ou nos casos em que as pálpebras cobriam parcialmente a íris, o algoritmo implementado detecta a circunferência da íris, com bom grau de precisão e alta eficiência computacional.

Dentre as parametrizações da Transformada de Hough utilizadas, a de Kimme, Ballard e Sklansky [18] apresentou o melhor desempenho na detecção da íris do usuário. Porém, foram

necessárias modificações no algoritmo para reduzir o número de *pixels* processados por imagem. Eliminamos os elementos próximos às bordas da imagem e o algoritmo continuou detectando a circunferência da íris com precisão. O sistema de Rastreo Ocular apresentou um tempo médio de 93 ms por imagem processada, tempo esse satisfatório para os requisitos do nosso dispositivo.

Destacamos no desenvolvimento dessa dissertação a montagem do nosso protótipo de Dispositivo Rastreador. Utilizamos uma micro-câmera com transmissão *wireless* para a aquisição de imagens em tempo real. O protótipo foi montado de forma artesanal fixando a micro-câmera no visor de acrílico de um óculos de proteção industrial. Durante a etapa de desenvolvimento tivemos alguns problemas com a micro-câmera utilizada, pois a mesma possui uma baixa qualidade focal e a maioria das imagens capturadas apresentava níveis variados de ruídos.

Mesmo em imagens ruidosas, a Transformada de Hough consegue detectar a circunferência da íris em um número elevado de imagens. O sistema de Rastreo Ocular apresentava alta taxa de erros nas imagens capturadas em ambientes com baixa iluminação. Após adicionarmos um filtro de manipulação de Histograma, o Dispositivo Rastreador passou a detectar a circunferência da íris mesmo em ambientes com baixa iluminação.

O processo de calibração desenvolvido utilizou a técnica de Calibração de Segunda Ordem com a análise de 25 pontos fixos na tela. Os pontos são mostrados de forma randômica para manter a atenção do usuário durante toda etapa de calibração. Se o usuário mantiver a cabeça imóvel durante o uso do sistema, a calibração do dispositivo mantém-se inalterada. O problema são os movimentos involuntários da cabeça do usuário que diminui gradativamente a precisão do Rastreo Ocular, sendo necessária uma nova etapa de calibração do protótipo.

A função de emulação dos cliques do mouse representa uma das maiores necessidade de aprimoramento no estágio atual do nosso Dispositivo Rastreador. Salientamos que o modelo adotado para emular os cliques do mouse com os piscar os olhos é inviável. Os voluntários da nossa pesquisa foram unânimes em relatar o estresse causado pelo modelo adotado. A idéia a princípio parecia ideal, pois poderíamos emular os cliques dos dois botões do mouse através da análise do piscar dos olhos esquerdo e direito do usuário.

Criamos um botão extra-monitor para auxiliar o sistema de Rastreo Ocular na maximização do sistema de atalhos. Todas as vezes que o usuário desejar abrir um programa instalado no seu computador e mapeado no sistema de atalhos, ele deve observar a região acima do monitor para que o sistema de atalhos seja maximizado acima das demais janelas abertas. A exploração do campo visual além da área do monitor é uma alternativa interessante de interação,

principalmente na criação de botões extra-monitor que respondam os comandos básicos do Sistema Operacional.

Uma das maiores contribuições dessa dissertação para a área de pesquisa sobre Rastreo Ocular foi o modelo de escrita com os olhos que utilizamos no sistema Teclado Óptico Ocular. O uso de regiões virtuais na tela para a escrita de textos é uma boa alternativa de Inclusão Digital para Pessoas Portadoras de Necessidades Especiais. Esse modelo permite o uso de Dispositivos Rastreadores com baixa precisão, já que as nove regiões virtuais criadas na tela são grandes o bastante para permitir um maior índice na taxa de erro do Dispositivo Rastreador.

Os testes com os voluntários mostraram que a taxa de erros apresentada por nosso protótipo foi de aproximadamente 1.75°. É uma taxa que consideramos aceitável para um rastreador fixo na cabeça sem um sensor que permita definir uma relação de distância entre o usuário e o monitor.

## 7.1 – Trabalhos Futuros

Durante a etapa de testes, observamos alguns aspectos que precisam ser melhorados no nosso Dispositivo Rastreador. O objetivo é dar continuidade a esse trabalho para permitir que o dispositivo seja utilizado com uma interface amigável de interação entre o usuário e o computador. A partir dessa dissertação é possível desenvolver pesquisas que envolvam estudos de Biometria, Ergonomia, Usabilidade, Visão Computacional entre outros.

Com a micro-câmera posicionada à frente do visor dos óculos de proteção, o usuário tem uma visão ofuscada do seu monitor. É necessária uma modificação na estrutura do protótipo para liberar o campo visual do olho direito do usuário. O protótipo será aprimorado com a micro-câmera fixa na parte superior dos óculos, e com um espelho fixo na parte inferior com uma leve inclinação. Desse modo, as imagens serão capturadas através do reflexo dos olhos no espelho, liberando o campo de visão do usuário.

O protótipo foi desenvolvido para realizar apenas o clique simples do botão esquerdo do mouse. É necessário a implementação da emulação do clique duplo e do botão direito do mouse. Para diminuir o estresse causado pelo modelo atual de emulação dos cliques do mouse, será realizada uma pesquisa para avaliar o modelo que melhor se adapte ao Mouse Óptico Ocular. Uma dos modelos que será avaliado é o criado por Hermann [14], que utiliza regiões sensíveis que mudam de cor antes de realizar o clique do mouse.

Ao final dessa dissertação, os estudos serão voltados para processos de calibração de Dispositivos Rastreadores. É um campo de pesquisa que vem crescendo dentro da área de Rastreo Ocular, mas ainda carece de resultados eficazes. Para evitar a perda de precisão do protótipo com o decorrer do seu uso, um processo de recalibração automática será incorporado ao Dispositivo Rastreador. O ideal é que o usuário possa utilizar o Dispositivo Rastreador de modo mais simples possível. Para uma melhor interação com o usuário, o processo de calibração deve incluir um modelo de “Calibração Interativa” similar ao sistema Tobii, onde são usados círculos que mudam de tamanho e cor; e se movimentam com efeito de interpolação no monitor.

Para finalizar o sistema do Teclado Óptico Ocular, é necessário incluir os mapeamentos com seqüências para a escrita de caracteres maiúsculos, especiais e numéricos. Além disso, é interessante desenvolver um sistema de atalhos para palavras mais próximas ao texto digitado pelo usuário. Por último, precisamos incluir uma seqüência nos mapeamentos desse sistema para que o usuário possa ativar e desativar o Dispositivo Rastreador quando houver necessidade.

---

## Referências Bibliográficas

---

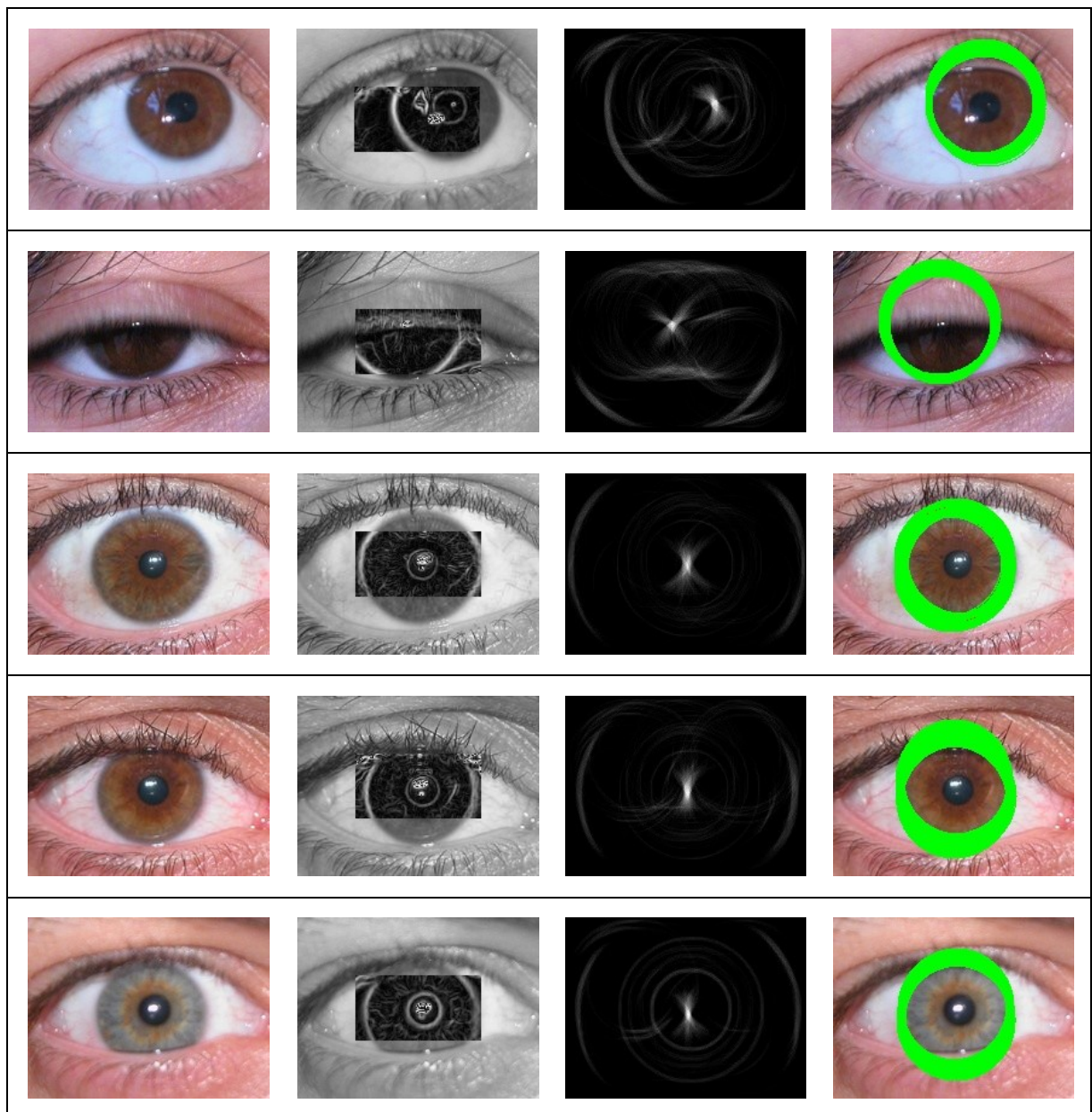
- [1] BALLARD, Dana H. “Generalizing the Hough Transform to Detect Arbitrary Shapes”, In: Pattern Recognition, v.13, n.2, p.111-122, 1981.
- [2] BARCELOS, Thiago Schumacher Barros. “Interfaces Prestativas Baseadas em Visão Computacional e Informação de Contexto”, Dissertação de Mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo – USP, São Paulo, 2004.
- [3] BEN-TZVI, D; SANDLER, Mark B. “A Combinatorial Hough Transform”, In: Pattern Recognition Letters, v.11, n.3, p.167-174, 1990.
- [4] BORGES, Geovany A. “GMATRIX: Uma Biblioteca Matricial para C/C++”. Disponível em <<http://www.ene.unb.br/~gaborges/recursos/programacao/gmatrix/gmatrixdoc.pdf>>. Acesso em: 07 de Dezembro de 2008.
- [5] CARDOSO, Manuel. “Fundação Desembargador Paulo Feitoza: Mouse Ocular”. Disponível em <<http://www.fpf.br/cont.php?modulo=hardware&op=mouse>>. Acesso em: 12 de Janeiro de 2007.
- [6] COUTINHO, Flávio Luiz. “Um Sistema de Rastreamento de Olhar Tolerante a Movimentações da Face”, Dissertação de Mestrado, Instituto de Matemática e Estatística, Universidade de São Paulo – USP, São Paulo, 2006.
- [7] DAUNYS, G.; RAMANAUSKAS, N. “Nonlinear mapping of Pupil Centre Coordinates from Image Sensor to Screen for Gaze Control Systems”, In: Proceedings of the ICCHP 2006, LNCS 4061, p.962-965, 2006.
- [8] DORINI, Leyza Elmeri Boldo; ROCHA, Anderson de Rezende. “Detecção de Círculos em Imagens Através da Transformada de Hough”, Trabalho Final da Disciplina de Processamento Digital de Imagens, Universidade Estadual de Campinas – UNICAMP, Campinas, 2005.
- [9] DUCHOWSKI, Andrew T. “Eye Tracking Methodology: Theory and Practice – Second Edition” London, Springer, 2007.
- [10] DUDA, Richard O.; HART, Peter E. “Use of the Hough Transformation to Detect Lines and Curves in Pictures”, In: Communications of the ACM, v.15, n.1, p.11-15, 1972.
- [11] FALCON, Jacques. “Sistema de Detecção de Objetos Circulares Através da Transformada de Hough Randômica”. Disponível em <<http://www.ppgia.pucpr.br/~facon/Hough/>>. Acesso em: 03 de Janeiro de 2006.

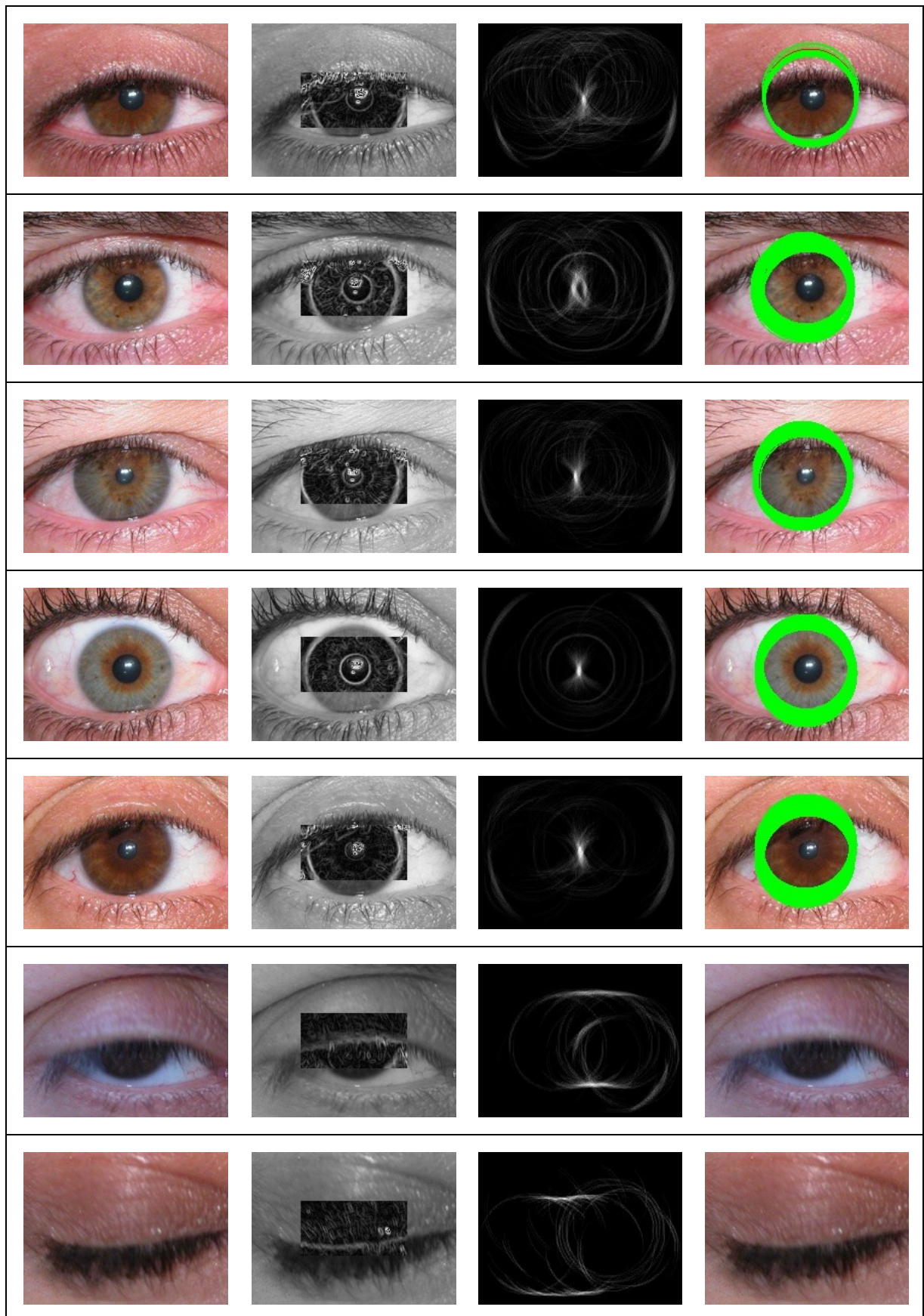
- [12] GNUPLOT. “GNUPlot Homepage”. Disponível em <<http://www.gnuplot.info/>>. Acesso em: 07 de Dezembro de 2008.
- [13] GONZALES, Rafael C.; WOODS, Richard C. “Digital Image Processing – 2nd Edition”, New Jersey, Prentice-Hall, 2002.
- [14] HERMANN, Fabian. “Eye-Controlled Computer Operation”. Disponível em <<http://www.fraunhofer.de/EN/press/pi/2006/09/ResearchNews92006Topic6.jsp>>. Acesso em: 10 de Setembro de 2007.
- [15] HOUGH, Paul V. C.; ARBOR, Michel Ann; “Method and Means for Recognizing Complex Patterns”, In: U. S. Patent 3.069.654, 1962.
- [16] ILLINGWORTH, John; KITTLER, Josef. “The Adaptive Hough Transform”, In: IEEE Transactions on Pattern Analysis and Machine Intelligence, v.9, n.5, p.690-698, 1987.
- [17] JAMUNDÁ, Teobaldo. “Reconhecimento de Formas: A Transformada de Hough”. Disponível em <<http://www.inf.ufsc.br/~visao/2000/Hough/>>. Acesso em: 09 de Fevereiro de 2006.
- [18] KIMME, Carolyn; BALLARD, Dana; SKLANSKY, Jack. “Finding Circles by an Array of Accumulators”, In: Communications of the ACM, v.18, p.120-122, 1975.
- [19] KIRYATI, Nahum; ELDAR, Yuval; BRUCKSTEIN, Alfred M. “A Probabilistic Hough Transform”, In: Pattern Recognition, v.24, n.4, p.303-316, 1991.
- [20] KULTANEN, Pekka; XU, Lei; OJA, Erkki. “A new curve detection method: randomized Hough transform (RHT)”, In: Pattern Recognition Letters, v.11, n.5, p.331-338, 1990.
- [21] KUMAR, Manu; PAEPCKER, Andreas; WINOGRAD, Terry. “EyePoint: Practical Pointing and Selection Using Gaze and Keyboard”, In: CHI: Conference on Human Factors in Computing Systems, San Jose, 2007.
- [22] LUCS-KHO. “Imagem disponível sob a licença de Domínio Público”. Disponível em: <<http://en.wikipedia.org/wiki/Image:ReadingFixationsSaccades.jpg>> Acesso em: 17 de Agosto de 2008.
- [23] MORAN, Thomas. “The Command Language Grammars: A Representation for the User Interface of Interactive Computer Systems”. In: International Journal of Man-Machine Studies, v.15, p.3-50, 1981.
- [24] MORIMOTO, Carlos Hitoshi; KOONS, Dave; AMIR, Arnon; FLICKNER, Myron; SHUMIN, Zhai. “Keeping an Eye for HCI”, In: SIBGRAPP'99 – XII Brazilian Symposium in Computer Graphics and Image Processing, Universidade Estadual de Campinas – UNICAMP, Campinas, 1999.
- [25] MORIMOTO, Carlos Hitoshi; KOONS, Dave; AMIR, Arnon; FLICKNER, Myron. “Pupil Detection and Tracking Using Multiple Light Sources”, In: Image Vision Computing, v.18, n.4, p.331-335, 2000.
- [26] MORIMOTO, Carlos Hitoshi; KOONS, Dave; AMIR, Arnon; FLICKNER, Myron. “Frame-Rate Pupil Detector and Gaze Tracker”. Disponível em: <<http://www.ime.usp.br/~hitoshi/framerate/>>. Acesso em: 10 de Dezembro de 2005.
- [27] O’GORMAN, Frank; CLOWES, M. B. “Finding Pictures Edges through Collinearity of Feature Points”, In: Proc. Third International Joint Conference on Artificial Intelligence, p.543-555, 1973.

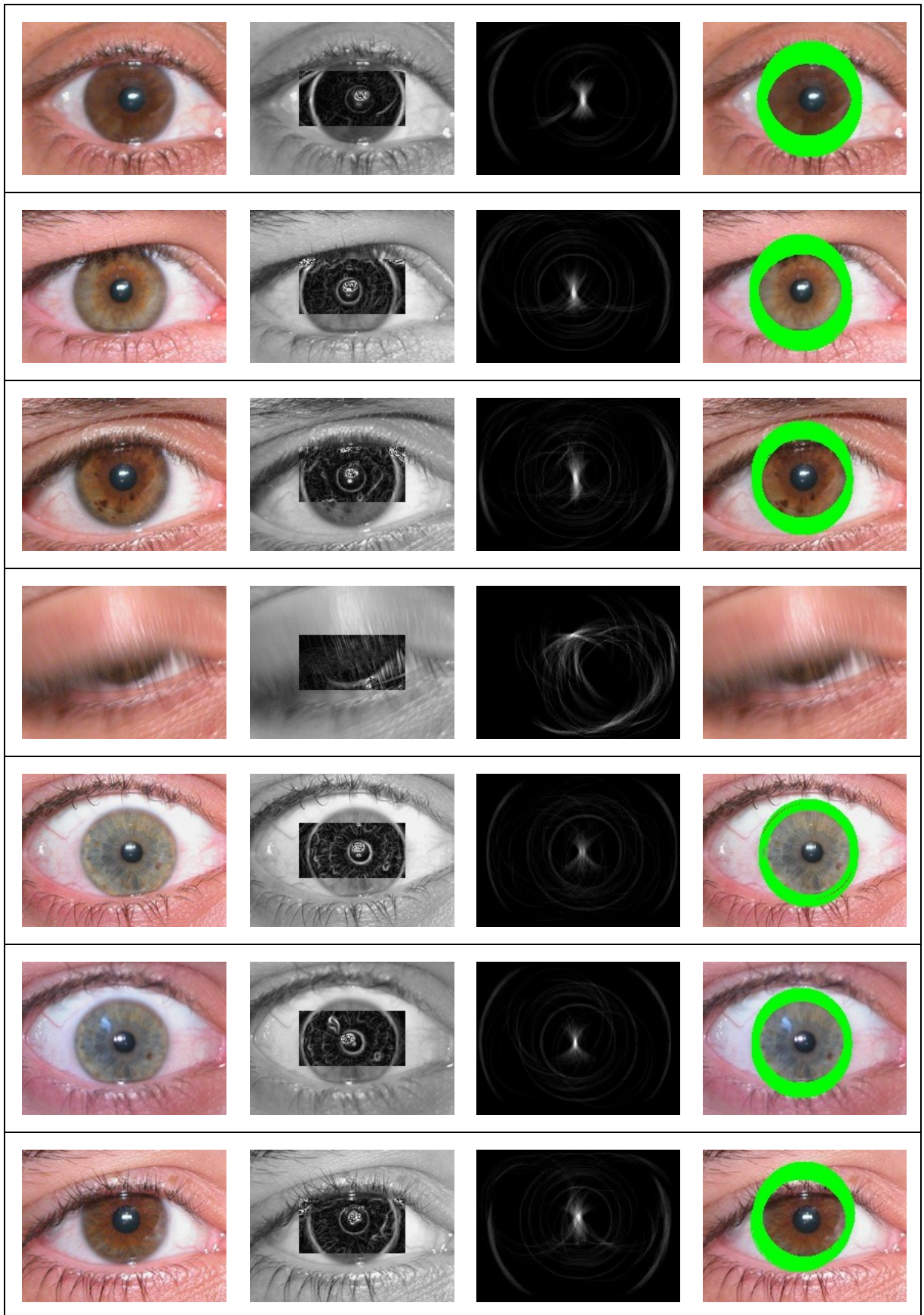
- [28] OHNO, Takehiko; MUKAWA, Naoki; YOSHIKAWA, Atsushi. “FreeGaze: A Gaze Tracking System for Everyday Gaze Interaction”, In: Proceedings of the Symposium on ETRA 2002: Eye Tracking & Applications Symposium, p.125-132, 2002.
- [29] OPENCV. “Welcome to the OpenCV Wiki”. Disponível em: <<http://opencv.willowgarage.com/wiki/>>. Acesso em: 07 de Dezembro de 2008.
- [30] PAULSON, Eric J. “Viewing Eye Movements During Reading Through the Lens of Chaos Theory: How Reading Is Like the Weather”, In: Reading Research Quarterly, v.40, n.3, p.338-358, 2005.
- [31] PERLIN, Ken. “Quikwriting: Continuous Stylus-Based Text Entry”, In: Proceedings of the ACM Symposium on User Interface Software and Technology – UIST’98 ACM, p.215-216, 1998.
- [32] PISTORI, Hemerson; PISTORI, Jeferson; COSTA, Eduardo Rocha. “Hough-Circles: Um Módulo de Detecção de Circunferências Para o ImageJ”, In: 6º Workshop de Software Livre 2005, Porto Alegre, 2005.
- [33] PRINCEN, John; ILLINGWORTH, John; KITTLER, Josef. “A Hierarchical Approach to Line Extraction Based on the Hough Transform”, In: Computer Vision, Graphics, and Image Processing, v.52, n.1, p.57-77, 1990.
- [34] POWELL, Mark. “Find Circles in an Image with a Hough Transform”, Software GNU Image Vision, 2000.
- [35] PROENÇA, Hugo; ALEXANDRE, Luís A. “UBIRIS: A Noisy Iris Image Database”, In: Proceedings of ICIAP 2005 - International Conference on Image Analysis and Processing, v.1, p.970-977, 2005.
- [36] RAMANAUSKAS, N.; DAUNYS, G. “Noise of Eye Movement Detection by Videoculography”, Electronics and Electrical Engineering, n.6 (55), p.25-28, 2004.
- [37] RAMANAUSKAS, N. “Calibration of Video-Oculographical Eye-Tracking System”, Electronics and Electrical Engineering, n.8 (72), p.65-68, 2006.
- [38] RHCSTILHO. “Imagem disponível sob a licença de Domínio Público”. Disponível em: <[http://pt.wikipedia.org/wiki/Imagem:Schematic\\_diagram\\_of\\_the\\_human\\_eye\\_pt.svg](http://pt.wikipedia.org/wiki/Imagem:Schematic_diagram_of_the_human_eye_pt.svg)> Acesso em: 17 de Agosto de 2008.
- [39] ROBINSON, D. A.; “A Method of Measuring Eye Movements Using a Scleral Search Coil in a Magnetic Field”, In: IEEE Transactions on Biomedical Engineering, BME 10, p.137-145, 1963.
- [40] SALVUCCI, Dario D.; GOLDBERG, Joseph H. “Identifying Fixations and Saccades in Eye-Tracking Protocols”. In: Proceedings of the Eye Tracking Research and Applications Symposium, ACM Press, New York, p.71-78, 2000.
- [41] SÉRIE CORPO HUMANO. “O Olho e a Visão”. Disponível em <<http://www.mundovestibular.com.br/articles/469/2/O-OLHO-E-A-VISAO/>>. Acesso em: 11 de Março de 2008.
- [42] SOUZA, Morgana N.; FOGGIATTO, Anderson; OSOWSKY, Jefferson; NOHAMA, Percy; GAMBA, Humberto Remigio. “Mouse Controlado Pelo Movimento do Olho”, In: Anais do XVIII Congresso Brasileiro de Engenharia Biomédica, Universidade do Vale do Paraíba – UNIVAP, São José dos Campos, v.5, p.237-240, 2000.
- [43] TERRALIB. “Tutorial de Programação: TerraLib 3.1.3”. Disponível em

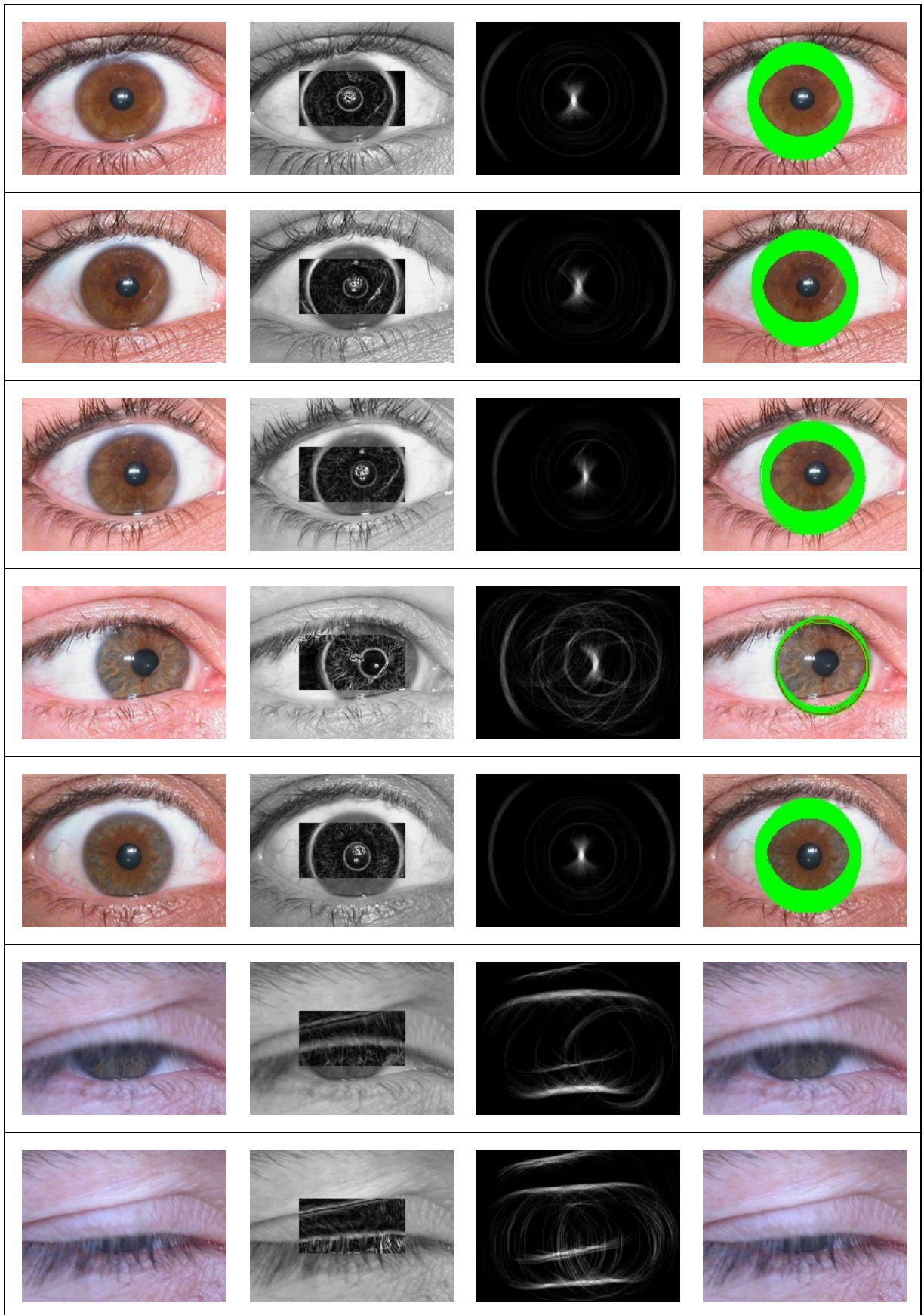
- <<http://www.terralib.org/docs/v313/TutorialProgramacaoTerraLib313.htm>>. Acesso em: 07 de Dezembro de 2008.
- [44] YLÄ-JÄÄSKI, Antti; KIRYATI, Nahum. “Adaptive Termination of Voting in the Probabilistic Circular Hough Transform”, In: IEEE Transactions on Pattern Analysis and Machine Intelligence, v.16, n.9, p.911-915, 1994.
- [45] YOO, Dong Hyun; KIM, Jae Heon; LEE, Bang Rae; CHUNG, Myoung Jin. “Non-Contact Eye Gaze Tracking System by Mapping of Corneal Reflections”, In: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition – FG’02, p.101, 2002.
- [46] YOO, Dong Hyun; CHUNG, Myoung Jin. “A Novel Non-Intrusive Eye Gaze Estimation Using Cross-Ratio Under Large Head Motion”, In: Computation Vision and Image Understanding”, v.98, n.1, p.25-51, 2005.
- [47] YOUNG, L.; SHEENA, D. “Survey of Eye Movement Recording Methods”, In: Behavioral Research Methods & Instrumentation, v.7 (5), p.397-429, 1975.
- [48] WANG, Jingtao; ZHAI, Shumin; SU, Hui. “Chinese Input With Keyboard and Eye-Tracking – An Anatomical Study”, In: Proceedings of the CHI’01, p.349-356, 2001.
- [49] ZHAI, Shumin; MORIMOTO, Carlos Hitoshi; IHDE, Steven. “Manual and Gaze Input Cascaded (MAGIC) Pointing”, In: Conference on Human Factors in Computing Systems, Pittsburgh, Pennsylvania, p.246-253, 1999.
- [50] ZHU, Jie.; YANG, Jie. “Subpixel Eye Gaze Tracking”, In: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, p.131-136, 2002.

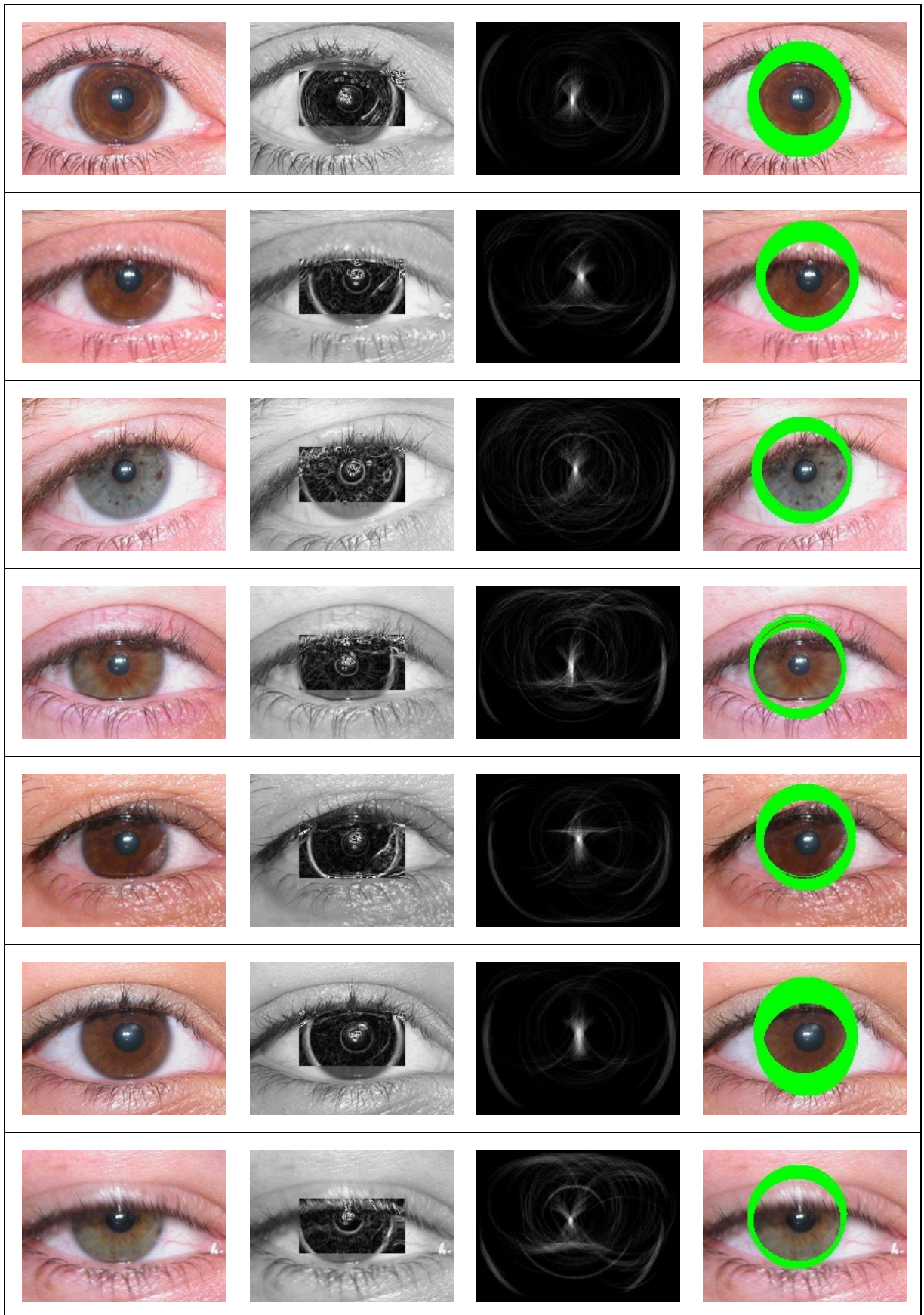
## Imagens Processadas no Trabalho

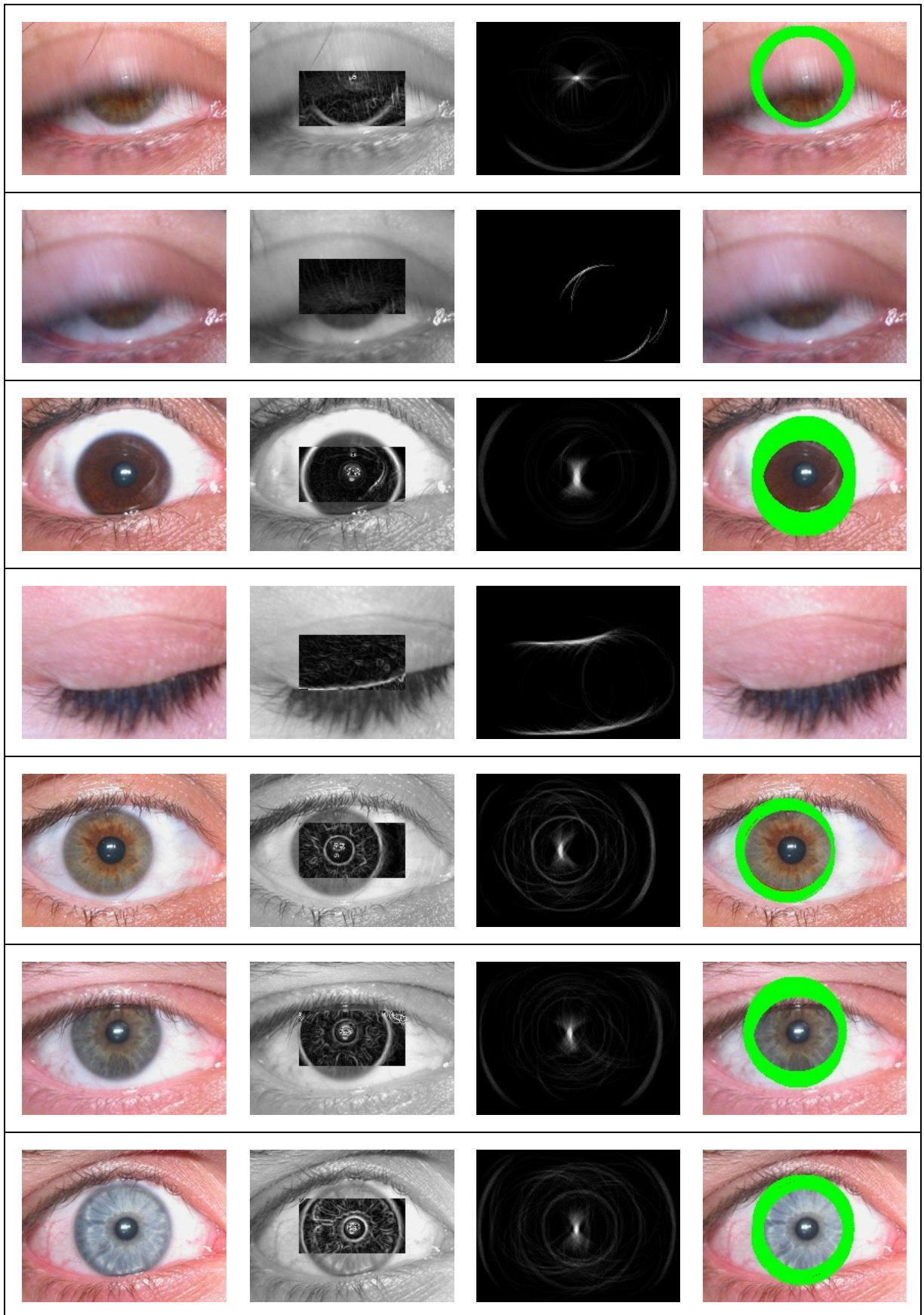


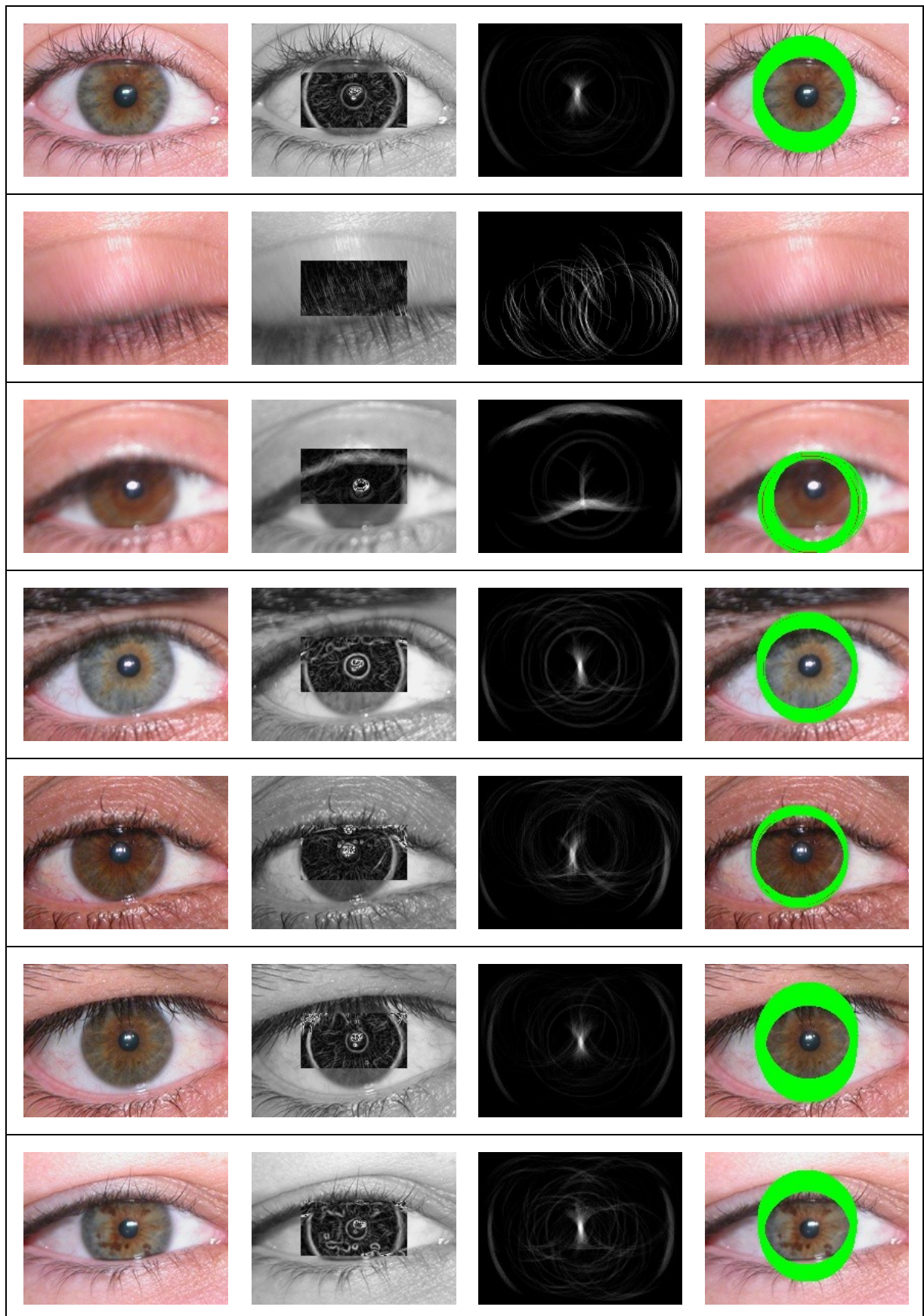


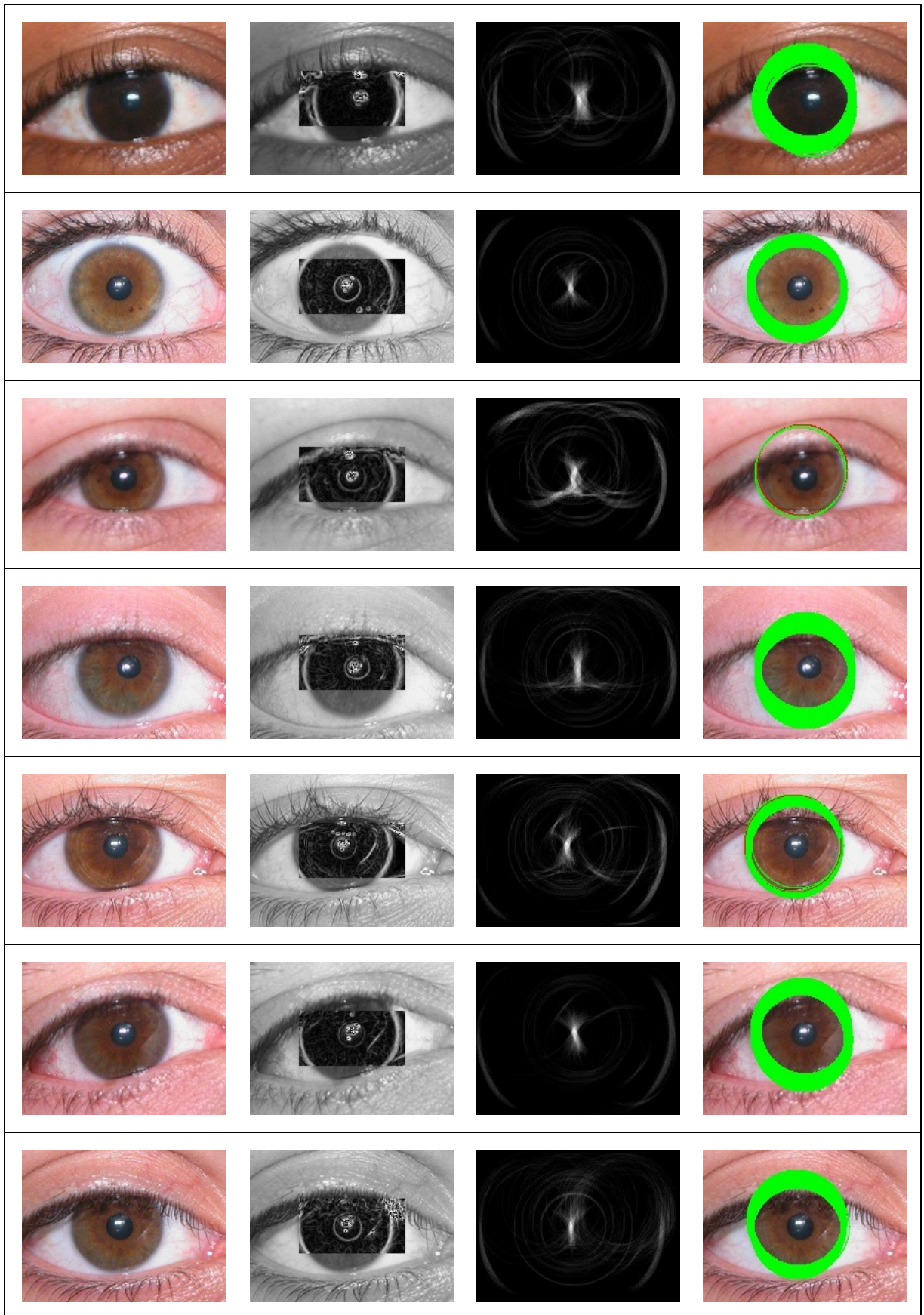


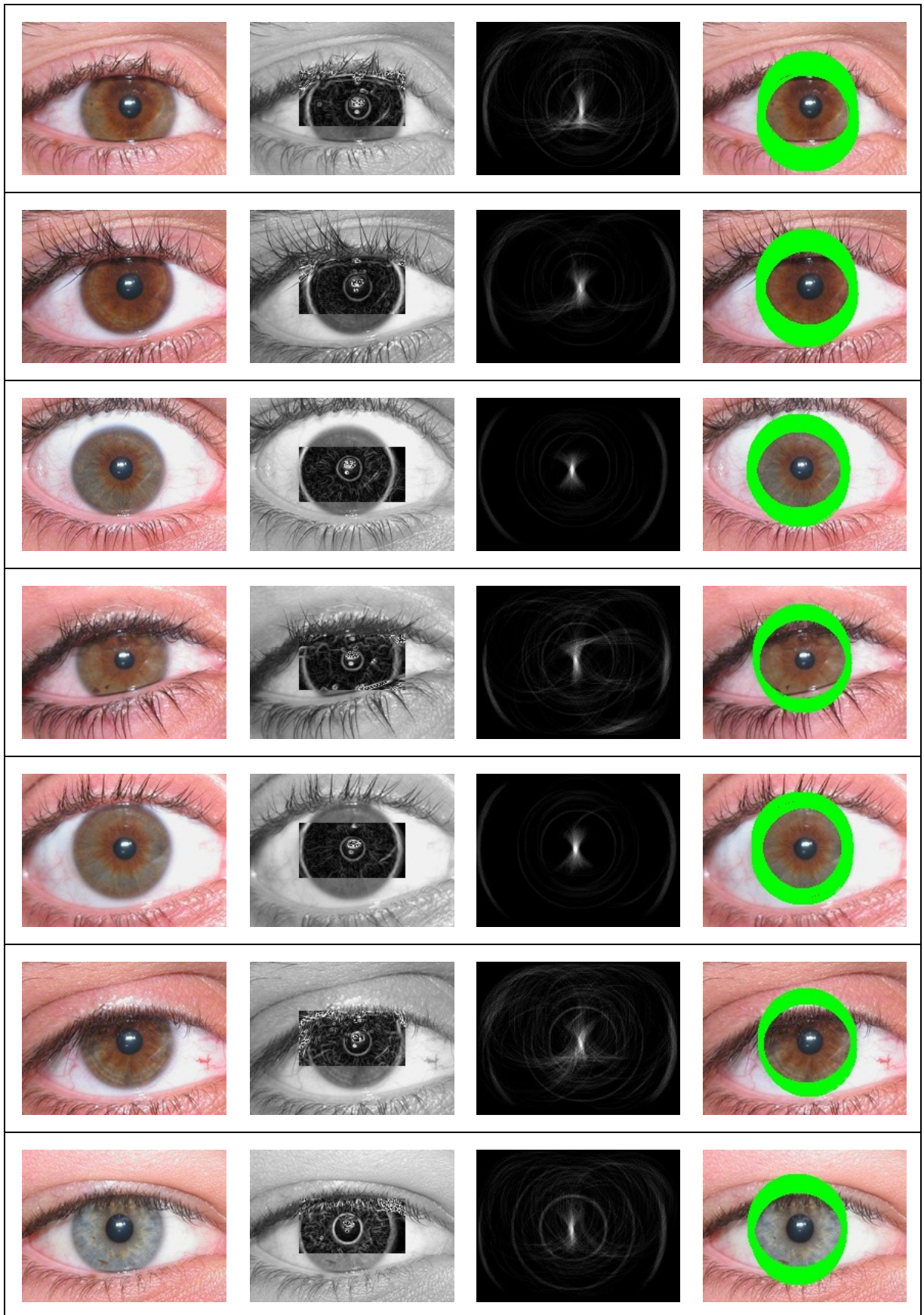


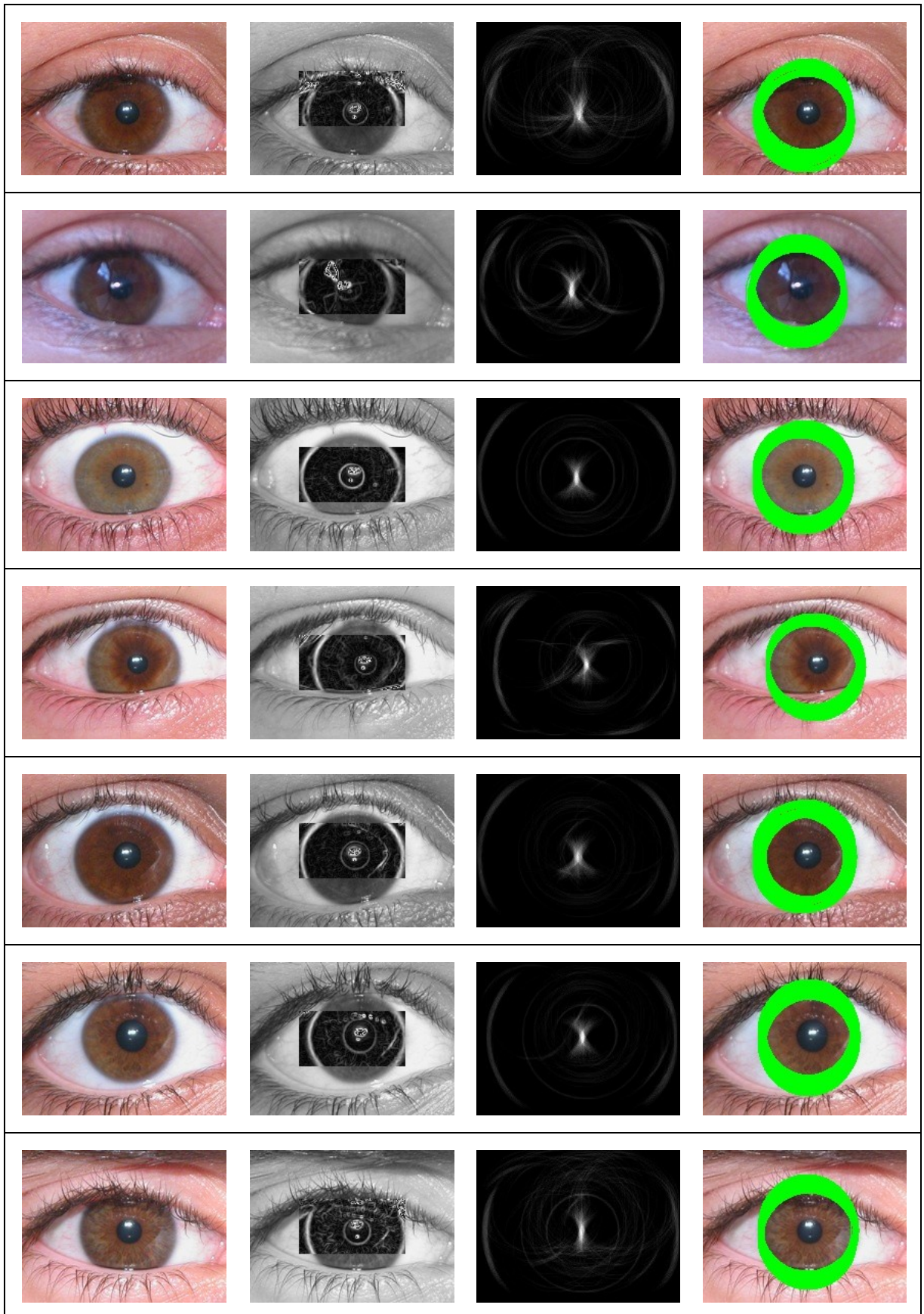


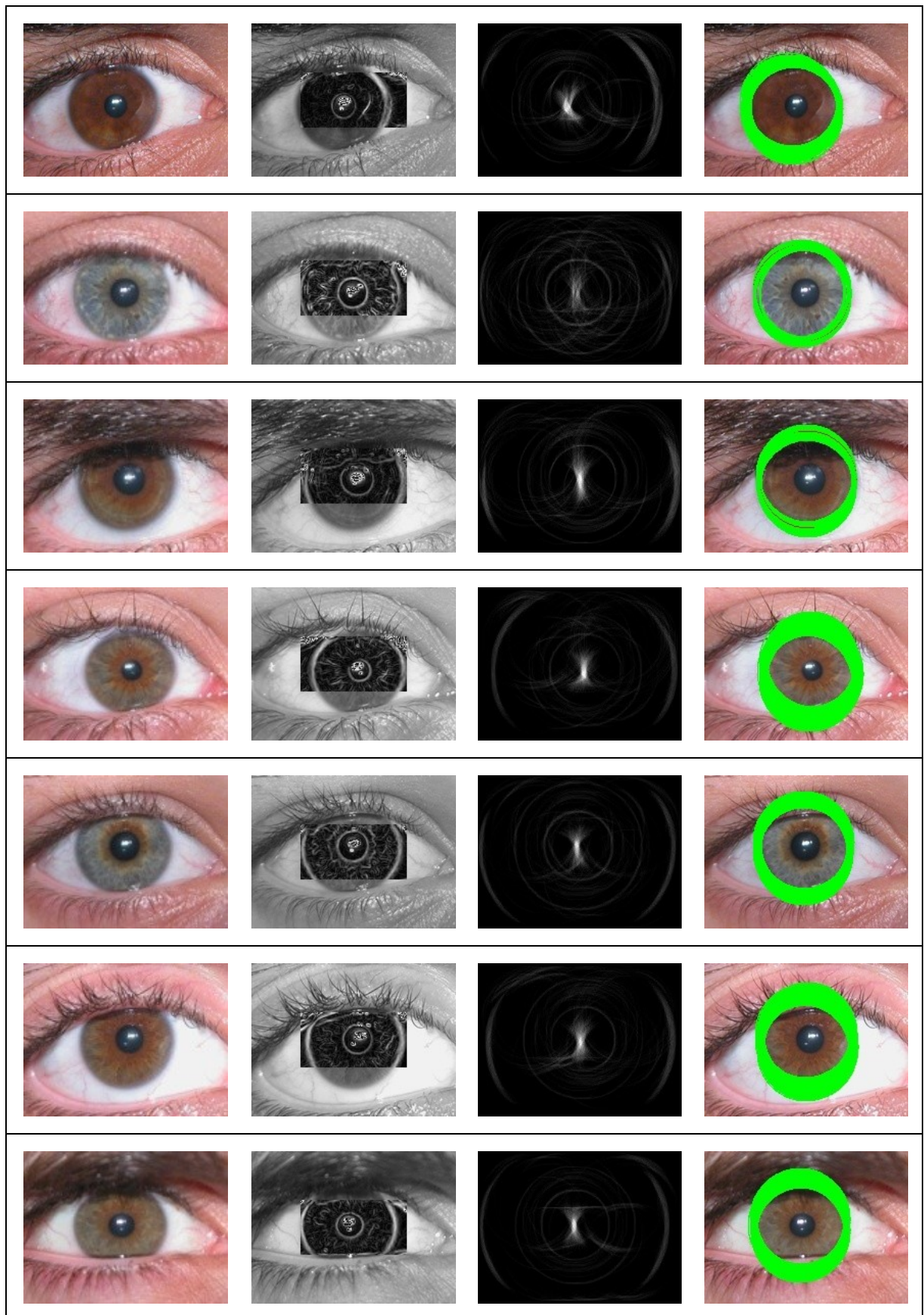


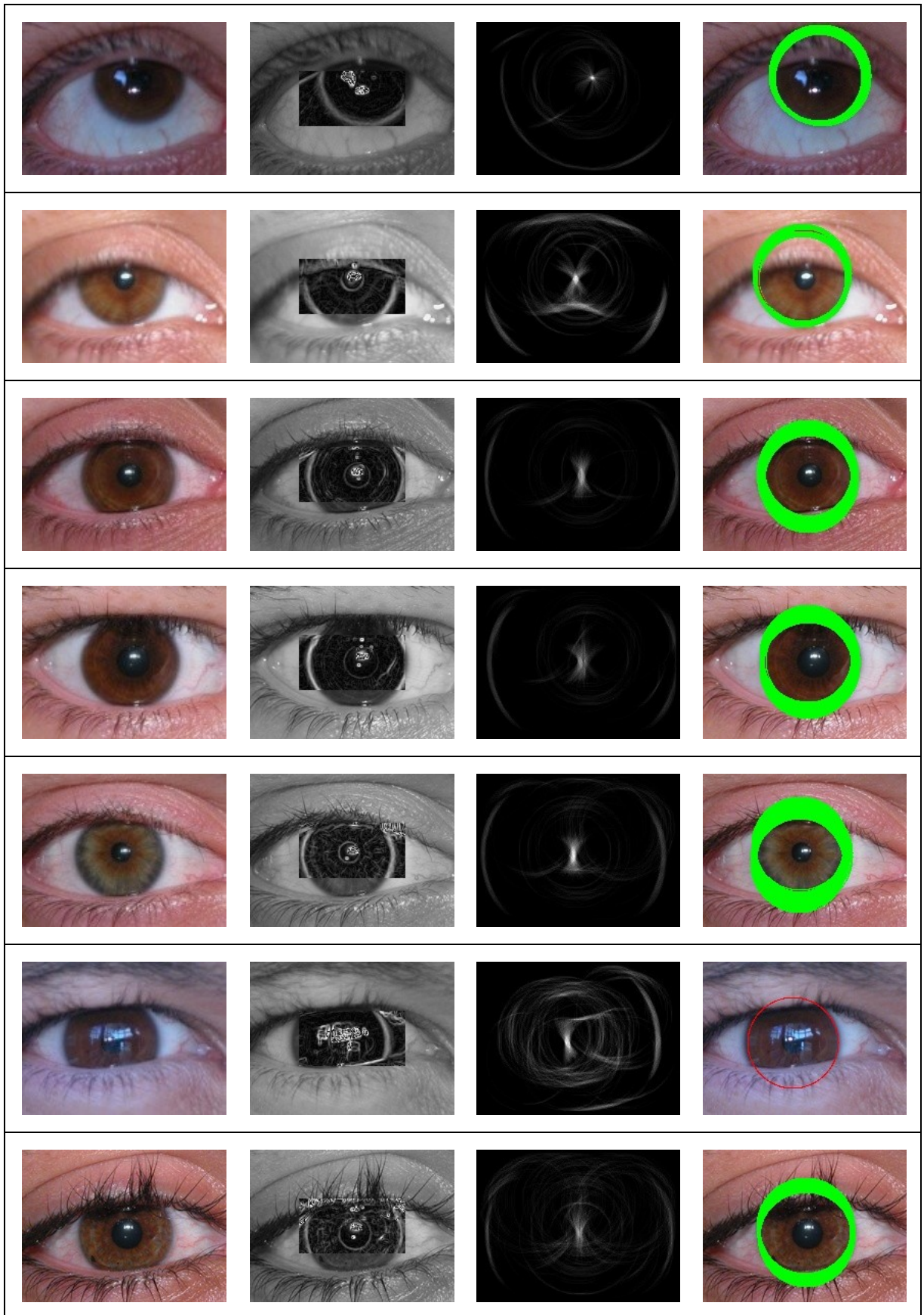


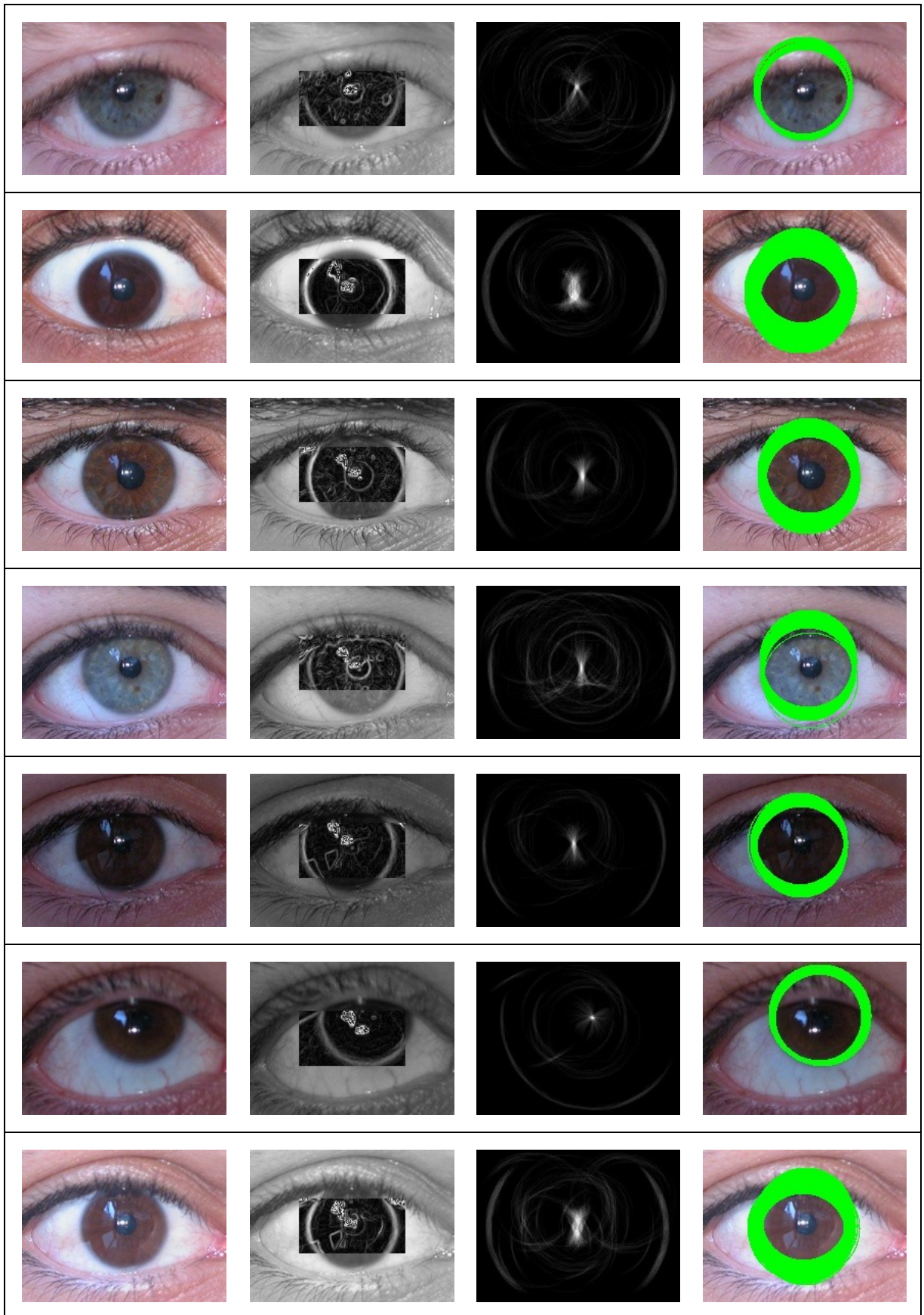


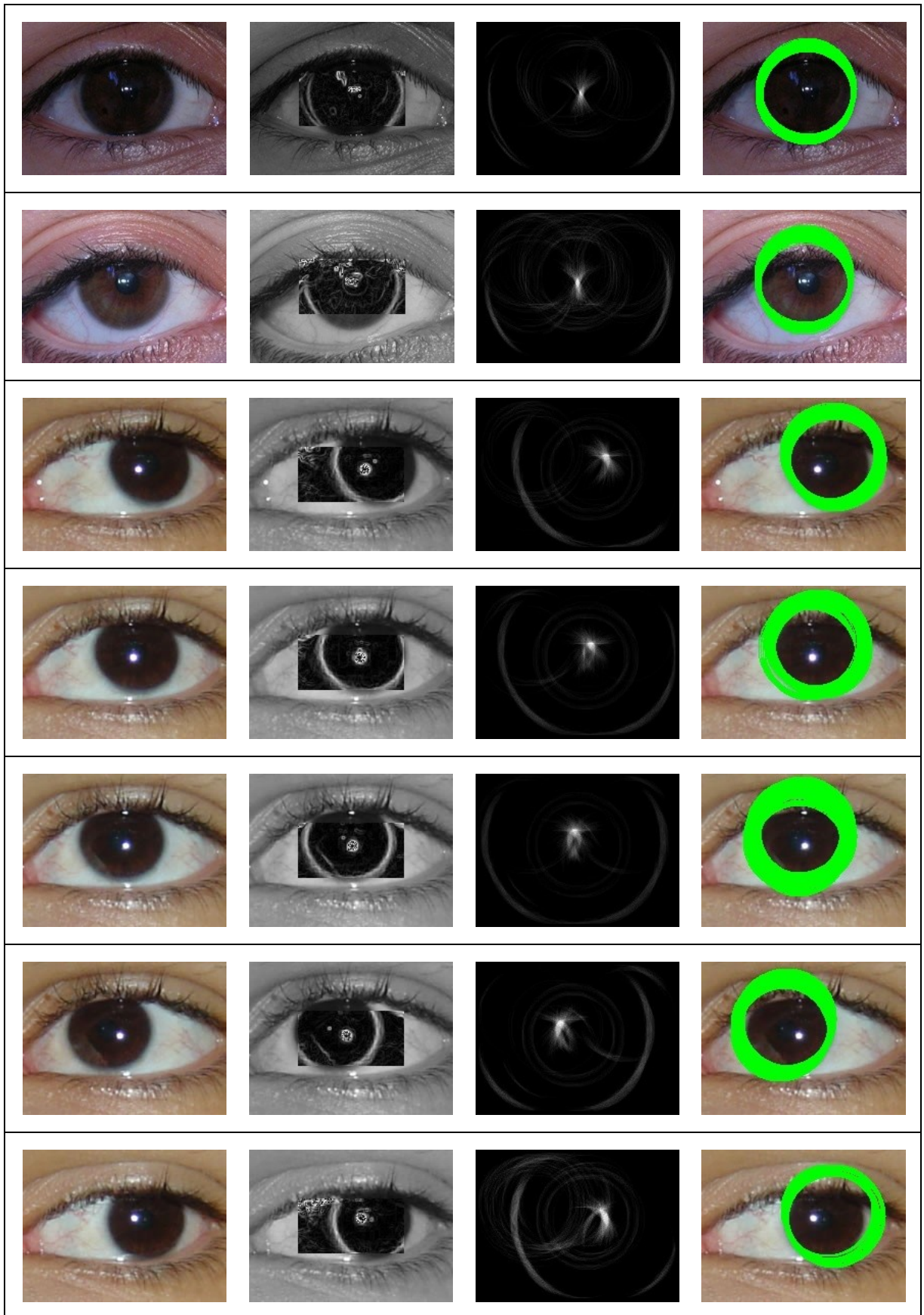


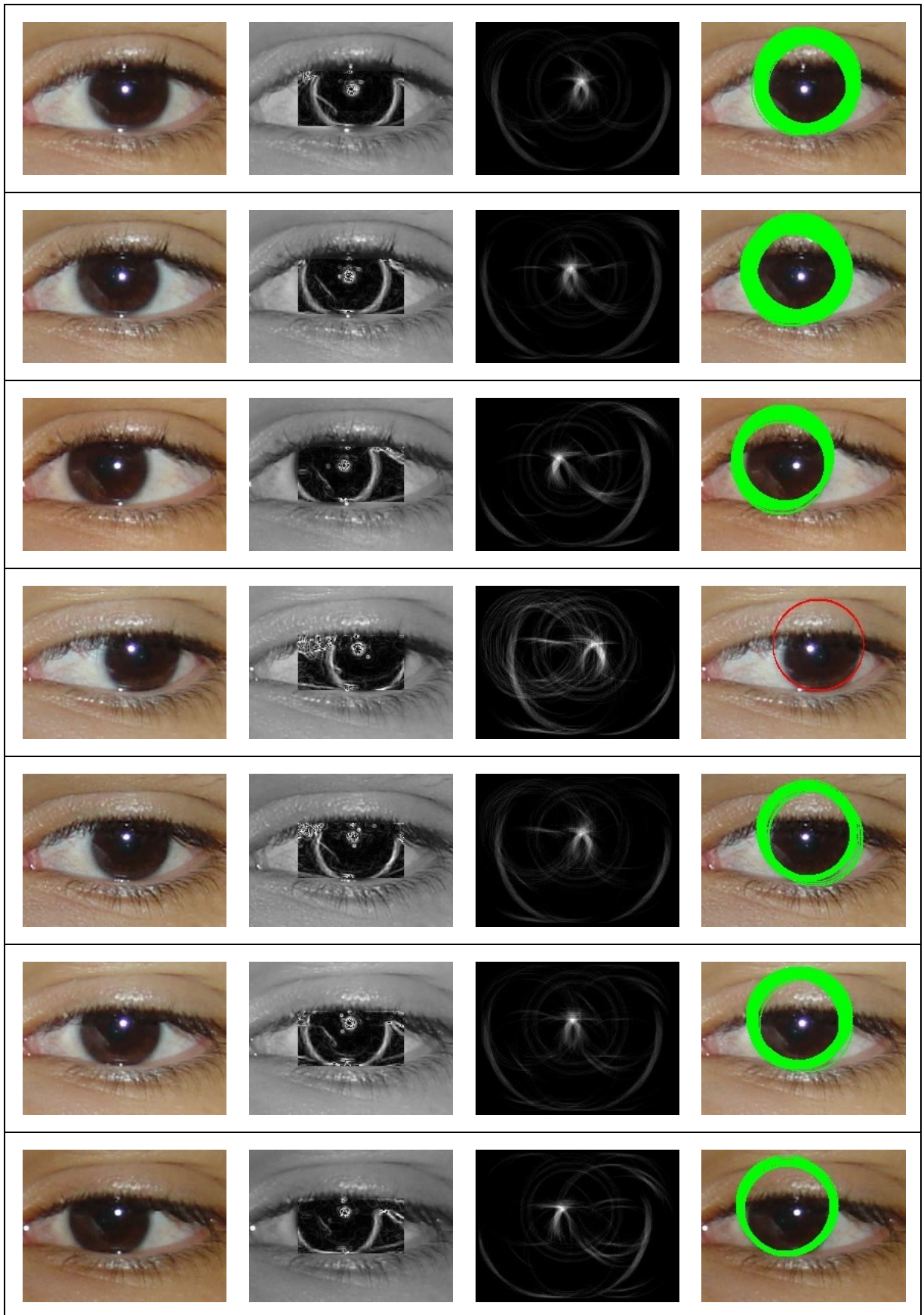












# Autorizações

## TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS (SEM FINS COMERCIAIS)

Eu, Elizabete Munzlinger, 3722454 SSP/SC autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008

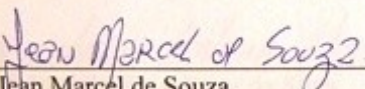
  
Elizabete Munzlinger

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Jean Marcel de Souza, RG 33764012-9 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008

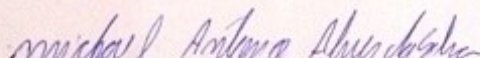
  
Jean Marcel de Souza

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Michael Antônio Alves da Silva, 42226598-6 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008

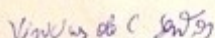
  
Michael Antônio Alves da Silva

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Vinicius de Carvalho Santos, 29839132-6 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008

  
\_\_\_\_\_  
Vinicius de Carvalho Santos

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Talyta Serrano Mathias Pinheiro, RG 32276236-4 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008



---

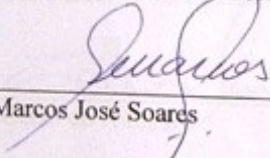
Talyta Serrano Mathias Pinheiro

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Marcos José Soares, 27685755-0 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008

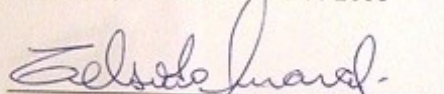
  
\_\_\_\_\_  
Marcos José Soares

TERMO DE AUTORIZAÇÃO DE USO DE DADOS E IMAGENS  
(SEM FINS COMERCIAIS)

Eu, Celso do Amaral, RG 23416276-4 SSP/SP autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar dados e imagens obtidas nos testes dos sistemas Mouse Óptico Ocular e Teclado Óptico Ocular, em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso dos dados e imagens para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

São Paulo, 15 de Setembro de 2008


  
Celso do Amaral

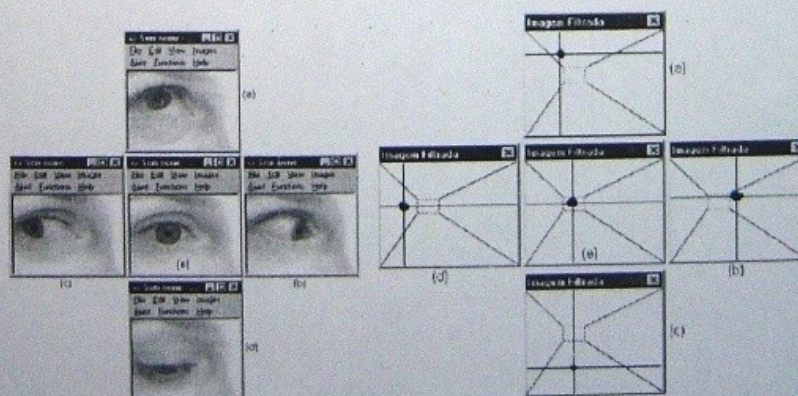
TERMO DE AUTORIZAÇÃO DE USO DE IMAGEM  
(SEM FINS COMERCIAIS)

Eu, Humberto Remigio Gamba, RG 30809653 IIPR autorizo Fabricio da Silva Soares, RG 06843413-87 SSP/BA a utilizar a imagem, de minha propriedade intelectual (inserida em Anexo I abaixo), em seu trabalho de Dissertação de Mestrado pelo Instituto Tecnológico de Aeronáutica (2008), em documentos impressos ou digitais, que se façam necessários, com a finalidade única de complementar seu trabalho acadêmico.

A presente autorização de uso abrange, exclusivamente, a concessão de uso da imagem para os fins aqui estabelecidos, pelo que qualquer outra forma de utilização e/ou reprodução, deverá ser previamente autorizada para tanto.

Curitiba, 21 de agosto de 2008

  
Humberto Remigio Gamba



Anexo I – Imagem de uso concedido

## Tabelas de Testes

Diferença da precisão com suaves movimentos sob o eixo x (monitor em formato normal)

Voluntário	12 regiões virtuais no monitor em formato "normal"								
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª
1	2.07	1.70	3.51	1.95	3.28	3.47	5.79	5.89	4.08
2	3.26	2.85	2.58	4.46	2.50	6.77	4.23	3.04	2.00
3	1.77	3.07	5.56	3.66	1.65	2.35	3.30	7.77	2.08
4	3.13	2.43	4.49	4.39	2.82	2.48	3.37	2.54	2.19
5	3.09	3.88	1.79	3.46	3.61	3.74	4.84	5.02	2.60
6	1.82	1.95	7.11	1.47	2.00	2.28	2.64	2.50	4.83
média	3.35 centímetros $\sigma = 1.46$								

Diferença da precisão com suaves movimentos sob o eixo y (monitor em formato normal)

Voluntário	12 regiões virtuais no monitor em formato "normal"								
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª
1	3.08	5.43	3.41	4.39	5.22	3.24	5.41	6.49	4.59
2	4.23	4.07	3.59	4.86	4.15	3.81	7.39	4.84	4.70
3	5.48	4.21	8.61	5.67	2.38	4.91	7.04	5.40	3.71
4	2.95	5.29	3.21	8.05	1.96	3.80	5.73	5.29	5.41
5	2.03	2.85	9.66	2.96	1.70	6.29	6.01	2.19	2.80
6	4.99	4.57	3.30	7.97	4.12	3.33	3.81	6.08	5.76
média	4.67 centímetros $\sigma = 1.73$								

Diferença da precisão com suaves movimentos sob o eixo z (monitor em formato normal)

Voluntário	12 regiões virtuais no monitor em formato "normal"								
	1ª	2ª	3ª	4ª	5ª	6ª	7ª	8ª	9ª
1	7.99	3.24	1.18	4.93	3.33	4.18	6.06	9.93	4.69
2	8.19	6.29	4.54	5.81	5.37	4.09	5.86	5.80	10.23
3	6.89	5.63	4.37	5.79	4.71	4.19	7.48	9.99	11.79
4	2.81	9.74	7.60	10.73	10.08	5.69	7.50	4.59	2.70
5	6.72	5.96	6.49	5.36	6.24	4.86	8.65	12.14	5.37
6	3.51	2.31	11.45	4.74	2.77	7.57	6.54	5.69	6.70
média	6.24 centímetros $\sigma = 2.54$								

## FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO <p style="text-align: center;">DM</p>	2. DATA <p style="text-align: center;">01 de dezembro de 2008</p>	3. REGISTRO N° <p style="text-align: center;">CTA/ITA/DM-075/2008</p>	4. N° DE PÁGINAS <p style="text-align: center;">140</p>
5. TÍTULO E SUBTÍTULO: Mecanismos de Interação Ocular Baseados em Imagens Voltados à Inclusão Digital de Portadores de Necessidades Especiais			
6. AUTOR(ES): <b>Fabricio da Silva Soares</b>			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: Rastreo Ocular; Dispositivos Rastreadores; Processamento Digital de Imagens; Transformada de Hough; Portadores de Necessidades Especiais.			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Rastreamento óptico; Processamento digital de imagens; Reconhecimento de alvo; Detecção de sinais; Controle automático; Transformada de Hough; Algoritmos; Deficientes físicos; Computação; Engenharia eletrônica			
10. APRESENTAÇÃO: ITA, São José dos Campos. Curso de Mestrado. Programa de Pós-Graduação em Engenharia Eletrônica e Computação. Área de Informática. Orientador: Carlos Henrique Quartucci Forster. Defesa em 28/11/2008. Publicada em 2008.			
11. RESUMO: Essa dissertação apresenta o desenvolvimento de um protótipo de Dispositivo Rastreador (Eye Tracker) que permite a interação entre o usuário e o computador através dos movimentos oculares. O Rastreo Ocular (Eye Tracking) é o principal estudo dessa dissertação. Através de técnicas de Processamento Digital de Imagens procuramos detectar o Ponto de Interesse (Point of Regard) do usuário na tela do seu computador. O nosso protótipo foi desenvolvido com base nas técnicas de Foto-Oculografia e Vídeo-Oculografia, onde é possível detectar a íris do usuário em imagens e vídeos capturados em tempo real. O principal problema na utilização da íris para detectar o Ponto de Interesse do usuário, é obter um bom grau de precisão vertical mesmo nas imagens capturadas com a íris parcialmente coberta pelas pálpebras. A solução adotada foi a utilização dos algoritmos de detecção de círculos baseados na Transformada de Hough, onde além de detectar a íris com bom grau de precisão, foi possível obter uma taxa de processamento de 93 milissegundos por imagem analisada. O hardware do Dispositivo Rastreador foi montado artesanalmente, fixando o Dispositivo de Captura de Vídeo no visor direito de um óculos de proteção industrial. Em conjunto com o hardware, foram desenvolvidos softwares para permitir que Portadores de Necessidades Especiais possam emular o uso dos dispositivos de entrada padrão do computador (mouse e teclado) através dos movimentos oculares. Os principais sistemas criados foram: o “Mouse Óptico Ocular” e o “Teclado Óptico Ocular”. O Mouse Óptico Ocular posiciona o cursor do mouse na região observada pelo usuário na tela do computador e o Teclado Óptico Ocular digita seqüências de caracteres através do mapeamento dos movimentos oculares do usuário. Ao final dessa dissertação, testes mostraram a eficácia dos modelos adotados no desenvolvimento do nosso protótipo e a possibilidade real do Dispositivo Rastreador auxiliar na Inclusão Digital de indivíduos sem a mobilidade dos seus membros superiores.			
12. GRAU DE SIGILO: <b>(X) OSTENSIVO</b> ( ) RESERVADO                      ( ) CONFIDENCIAL                      ( ) SECRETO			