Universidade de São Paulo
Escola de Engenharia de São Carlos

Claudine Mizusaki Iyomasa

***Software* de processamento de imagens para identificação de imagens de ceratometria em
dispositivos móveis**

São Carlos

2010

Claudine Mizusaki Iyomasa

***Software* de processamento de imagens para identificação de imagens de ceratometria em
dispositivos móveis**

**Dissertação apresentada à Escola de
Engenharia de São Carlos, da Universidade
de São Paulo para a obtenção do título de
Mestre em Ciências, Programa de Engenharia
Elétrica.**

**Área de Concentração: Processamento de
Sinais e Instrumentação**

Orientadora: Profa. Dra. Liliane Ventura

São Carlos

2010

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTES
TRABALHOS, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO,
PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

I96s Iyomasa, Claudine Mizusaki
 Software de processamento de imagens para identificação
de imagens de ceratometria em dispositivos móveis /
Claudine Mizusaki Iyomasa ; orientadora Liliane Ventura.
-- São Carlos, 2011.

Dissertação (Mestrado - Programa de Pós-Graduação em
Engenharia Elétrica e Área de Concentração Processamento
de Sinais e Instrumentação) -- Escola de Engenharia de
São Carlos da Universidade de São Paulo, 2011.

1. Processamento digital de imagens. 2. *Software* para
ceratômetro. 3. Dispositivos móveis. I. Título.

FOLHA DE JULGAMENTO

Candidato(a): Bacharel CLAUDINE MIZUSAKI IYOMASA.

Dissertação defendida e julgada em 20/12/2010 perante a Comissão Julgadora:

Profª. Associada **LILLANE VENTURA SCHIABEL** – (Orientadora)
(Escola de Engenharia de São Carlos/USP) APROVADA

Prof. Associado **SIDNEY JULIO DE FARIA E SOUSA**
(Faculdade de Medicina de Ribeirão Preto/FMRP) Aprovado

Prof. Dr. **JEAN JACQUES GEORGES SOARES DE GROOTE**
(Faculdades COC) Aprovado

Prof. Titular **DENIS VINÍCIUS COURY**
Coordenador do Programa de Pós-Graduação em
Engenharia Elétrica

Prof. Associado **PAULO CÉSAR LIMA SEGANTINE**
Presidente da Comissão da Pós-Graduação da EESC

*Dedico este trabalho aos meus pais, Sonia e Kazunori, por me ajudarem a superar os
obstáculos e vencer meus objetivos sempre!*

AGRADECIMENTO

À Profa. Dra. Liliane Ventura pelas oportunidades, orientação, paciência, e pelo tempo dedicado ao meu aprendizado e crescimento profissional. Obrigada por estar sempre presente quando precisei, pelo otimismo, e pela alegria contagiante.

Ao Prof. Dr. Jean-Jacques por toda colaboração e suporte a este trabalho. Seu apoio e motivação foram essenciais para que esse trabalho fosse concluído. Obrigada por acreditar e investir no meu potencial.

Ao portento Dr. Sidney pelo tempo concedido e pelas idéias brilhantes para o desenvolvimento deste projeto, e para o crescimento da área de oftalmologia e saúde.

Aos amigos do Laboratório de Instrumentação Oftálmica, os que passaram por lá e os que continuam, pela companhia e pelo suporte.

E a todos meus amigos, em especial à Thaís Ohashi e ao Renato Varoto, que acompanharam toda essa jornada me ajudando, apoiando e incentivando, com verdadeiras provas de amizade.

*“O sucesso nasce do querer, da determinação e persistência em se chegar a um objetivo.
Mesmo não atingindo o alvo, quem busca e vence obstáculos, no mínimo fará coisas
admiráveis.”*

| José de Alencar

RESUMO

IYOMASA, C. M. **Software de processamento de imagens para identificação de imagens de ceratometria em dispositivos móveis**. 2010. 79f. Dissertação (Mestrado) - Escola de Engenharia Elétrica de São Carlos, Universidade de São Paulo, São Carlos, 2010.

Neste trabalho é apresentado um *software* baseado em técnicas de processamento digital de imagens para medida de astigmatismo corneano para dispositivos portáteis. O sistema físico, um dispositivo adaptado para lâmpadas de fenda, projeta 36 pontos luminosos dispostos em um formato preciso de círculo na região central da córnea. O deslocamento, o tamanho e a deformação da imagem refletida destes pontos luminosos pela córnea são analisados proporcionando a ceratometria.

O *software* apresentado neste projeto foi desenvolvido para poder ser instalado nos principais dispositivos móveis que disponibilizem a plataforma Java ME, ou seja, pode ser implementado em computadores portáteis sem depender da plataforma utilizada.

Este projeto permitiu a viabilidade de portabilidade para o ceratômetro, ou seja, imagens originais do ceratômetro em Lâmpada de Fenda, desenvolvido pelo Laboratório de Instrumentação Oftálmica, foram inseridas em dispositivos móveis e processadas.

As imagens em formato BMP, de tamanho típico de 640x480 pixels são processadas e os 36 pontos luminosos são detectados em média em 1150 ms. Então, é traçada a elipse que melhor se ajusta a estes pontos e os raios de maior e menor dimensão são calculados em pixels.

Todo o processo é realizado em média em 1199 ms.

A contribuição deste trabalho é mostrar a viabilidade de transformar o ceratômetro de Lâmpada de Fenda em um aparelho portátil.

Palavras-chave: *software* para ceratômetro, dispositivos móveis, Processamento Digital de Imagens.

ABSTRACT

IYOMASA, C. M. Keratometry Image processing software for portable devices 2010. 79f. Dissertation (Master's Degree) - Escola de Engenharia Elétrica de São Carlos, Universidade de São Paulo, São Carlos, 2010.

This work is about software development based on digital image processing techniques for corneal astigmatism measurements for portable devices. The physical system comprises a device adapted for slit lamps and projects 36 light spots displayed in a precise circle at the lachrymal film of the examined cornea. Keratometry is provided by the analysis of displacement, size and deformation of the reflected image of the light spots.

The software presented in this project was developed to be installed mainly for mobile devices that provide the Java ME platform, ie, it can be implemented in portable computers.

Furthermore, it allows portability for keratometers and is a previous work for a portable corneal topographer. It also shows viability of portability for the keratometer, ie, the original keratometer images in Slit Lamp, developed by the Laboratory of Ophthalmic Instrumentation, were embedded in mobile devices and have been processed.

The BMP type images, - typical size of 640x480 pixels - are processed and the 36 points of light are detected in an average of 1150 ms.

The best fitting ellipse in the reflected image of these points provides the major and minor ellipse radii in pixels.

The entire process is performed in 1199 ms as an average.

The contribution of this work is to show the viability of transforming the keratometer for Slit Lamps in a handheld device.

Keywords: Software for keratometry, Mobile devices, Digital Image Processing.

LISTA DE FIGURAS

Figura 3.1. O olho humano.....	5
Figura 3.2. Região central da córnea e o astigmatismo.	7
Figura 4.1. Reflexão especular de um objeto h por um espelho convexo de raio R . A imagem y é menor que h , e virtual.....	9
Figura 4.2. Modelo Javal.....	10
Figura 4.3. Ceratômetro de Helmholtz.....	11
Figura 4.4. Alinhamento da mira no processo de ceratometria. a) Mira em seu estado padrão. b) Mira em processo de alinhamento. c) Alinhamento com o primeiro valor K determinado. d) A imagem da mira com o segundo valor K determinado.	12
Figura 5.1. Sistema completo do ceratômetro de lâmpada de Fenda [COELHO, 1999].....	14
Figura 5.2. Gráfico apresentando as medidas do raio encontrado pelo sistema de uma esfera de raio conhecido de 8.08mm a cada 1° . [VENTURA et al., 2000a].....	16
Figura 5.3. Tela de apresentação contendo os eixos de maior e menor raios (lado esquerdo) e tela com uma análise do astigmatismo encontrado (lado direito) [Ventura et al., 2000].	16
Figura 5.4. Representação tridimensional da intensidade do brilho da imagem [VENTURA et al., 2007]	19
Figura 5.5. Tela de apresentação do protótipo fixal com mira para centralização do sistema [VENTURA et al., 2007].	25

Figura 5.6. Mira de 72 furos iluminada pela própria iluminação da lâmpada de fenda: (A) Visão frontal; (B) Visão lateral; (C) Imagem dos furos luminosos projetados na córnea [VENTURA et al., 2007].	26
Figura 6.1. Estatística segunda a ITU sobre o avanço do uso de telefones móveis.....	27
Figura 6.2 Exemplo de dispositivos portáteis.	28
Figura 7.1 Imagem digital representada pelas coordenadas cartesianas.	35
Figura 7.2 Exemplo de aplicação de máscara. Adaptado de Gonzalez (2003).	37
Figura 8.1. Mira luminosa.	38
Figura 8.2. Diagrama que esquematiza o princípio ceratométrico.	40
Figura 9.1. Transformação da imagem em escala de cinza.	46
Figura 9.2. Algoritmo da varredura angular para identificação dos LEDs.	47
Figura 9.3. Anel de LEDs dividido em setores 10 setores para identificar 10 LEDs.....	48
Figura 9.4. Exemplo de uma imagem com a identificação dos LEDs refletidos.	49
Figura 9.5. Sistema de coordenadas polares.	50
Figura 9.6. Algoritmo do cálculo do maior raio da elipse.	51
Figura 9.7. Algoritmo do cálculo do menor raio da elipse.....	51
Figura 9.8. Anel contínuo formado pelo processo de interpolação utilizando os pontos dos LEDs identificados pelo software. O maior e o menor raio são mostrados.	52
Figura 10.1. Imagem obtida de um olho com astigmatismo.	54
Figura 10.2. Imagem após o processamento do <i>software</i>	54
Figura 10.3. Tela inicial no simulador da Sun.	55

Figura 10.4. Imagens para teste e comparação dos resultados.....	57
--	----

SUMÁRIO

Capítulo 1. INTRODUÇÃO	1
Capítulo 2. OBJETIVOS.....	4
Capítulo 3. A CÓRNEA E O ASTIGMATISMO.....	5
3.1. Córnea.....	5
3.2. Poder de refração	6
3.3. Ametropias oculares.....	6
3.4. Astigmatismo	7
Capítulo 4. CERATÔMETROS	9
Capítulo 5. HISTÓRICO DO MÓDULO CERATOMÉTRICO.....	13
5.1. Primeiro Protótipo.....	13
5.1.1. Software desenvolvido	14
5.1.2. Calibração	15
5.1.3. Interface do software	16
5.1.4. Considerações finais do primeiro protótipo.....	16
5.2. Segundo Protótipo.....	18
5.2.1. Software Desenvolvido.....	19
5.2.2. Calibração do Sistema	21
5.2.3. Interface do software	21
5.2.4. Considerações finais do segundo protótipo.....	22

5.3. Terceiro Protótipo.....	22
5.3.1. Calibração do Sistema.....	24
5.3.2. Software desenvolvido.....	24
5.3.3. Interface do software.....	24
5.4. Protótipo Final.....	25
5.4.1. Anel ótico sem LEDS.....	25
Capítulo 6. DISPOSITIVOS MÓVEIS	27
6.1. PDA's	29
6.2. Mini Notebooks.....	31
6.3. Tablets	32
6.4. Smartphones	33
6.5. Processadores dos dispositivos móveis	33
Capítulo 7. PROCESSAMENTO DIGITAL DE IMAGENS.....	34
7.1. Introdução ao Processamento Digital de Imagens	34
7.2. Representação de imagens digitais	34
7.3. Profundidade de Bits.....	35
7.4. Máscara	36
Capítulo 8. METODOLOGIA.....	38
8.1. Mira ceratométrica	38
8.2. Modelo Matemático	40

Capítulo 9. DENVOLVIMENTO DO <i>SOFTWARE</i>	42
9.1. Linguagem de Programação Java e a <i>Java Virtual Machine</i> (JVM)	42
9.2. Java Image, Java Advanced Imaging e o Processamento Digital de Imagens	44
9.3. Descrição do Algoritmo	45
9.3.1. Aquisição da imagem e transformação em escala de cinza.....	45
9.3.2. Identificação dos LEDs	46
Capítulo 10. RESULTADOS E DISCUSSÃO.....	53
10.1. Imagens do Software em JavaME no simulador	55
Capítulo 11. CONCLUSÃO E TRABALHOS FUTUROS	61

Capítulo 1. INTRODUÇÃO

A predominância da córnea no poder refrativo do olho humano é um importante fator ao estímulo do desenvolvimento de equipamentos voltados para a determinação da topologia corneana. Entre os erros refrativos da refração ocular que podem ser investigados, as ametropias, estão a miopia, hipermetropia e o astigmatismo [VENTURA & CASTRO NETO, 1995].

O astigmatismo, que está relacionado à visão borrada, é causado pela incapacidade do sistema ocular focalizar nitidamente imagens de um objeto sobre a retina. Uma causa importante deste problema está relacionada a uma forma toroidal da córnea, diferenciando-a de olhos normais, emétopes, caracterizados por uma superfície que se aproxima da esférica. A determinação do astigmatismo, que é o foco deste trabalho, é importante para permitir uma visão adequada, que pode ser proporcionada por meio de diagnósticos precisos, que serão utilizados na prescrição de óculos, lentes de contato e, até mesmo, cirurgias refrativas.

O instrumento de medida do astigmatismo, cujo *software* é a proposta deste projeto, segue uma linha de análise que envolve a observação da reflexão de um alvo luminoso circular projetado na superfície anterior da córnea (camada lacrimal). Se o reflexo da imagem estiver distorcido em relação à projeção circular, é possível determinar o astigmatismo desta superfície analisada. Este processo é conhecido como ceratometria [GONÇALVES & KRITZ, 1997]. Desde o primeiro instrumento apresentado por Ferdinand Cuignet em 1820, a tecnologia de cada época foi sendo incorporada aos ceratômetros, permitindo processos cada vez mais eficientes de determinação do astigmatismo. Um exemplo que incorpora a tecnologia atual é o instrumento de medida utilizado neste trabalho, um alvo que projeta a

imagem de LEDs distribuídos em um círculo preciso sobre a córnea, acoplado a um aparelho bastante comum em consultórios oftalmológicos, a chamada lâmpada de fenda. Este é um sistema formado por um suporte para a cabeça do paciente, que permite a observação do olho por meio de um sistema binocular acoplado a ele. Adaptando o anel de LEDs ao aparelho, e utilizando um divisor de feixes para desviar parcialmente a imagem observada nos binóculos para um computador, a forma que o anel de LEDs assume ao ser refletido permite o cálculo do astigmatismo com uma grande precisão [VENTURA et al., 2006]. Para isso deve ser utilizado um programa baseado em técnicas de processamento de imagens para extrair a posição dos LEDs na imagem digital e para então, estimar o astigmatismo.

Como base de desenvolvimento do *software* foi utilizada a plataforma Java, especificamente a versão para dispositivos de baixa potência, o Java Platform, Micro Edition, (JavaME, antigo J2ME¹). Esta escolha se deve a grande versatilidade da linguagem, que utiliza uma máquina virtual, instalada na maioria dos dispositivos computacionais e sistemas operacionais lançados no mercado [MARTINS & BOTELHO, 2008]. Embora outras linguagens, como C, permitam que os algoritmos sejam executados de forma mais rápida, os programas precisariam ser adaptados de aparelho para aparelho, enquanto que no caso do Java, basta que a máquina virtual esteja instalada, o que é padrão. Entre os dispositivos acessíveis estão, *palmtops*, *pockets*, celulares ou smartphones, *laptops* em geral e mesmo os *desktops*. Além disso, entre os sistemas que podem ser utilizados estão os dispositivos básicos de processamento utilizados como base para construção de celulares, que podem ser utilizados para o desenvolvimento de ceratômetros portáteis.

Para compensar o tempo superior de execução do programa, devido à utilização do Java, e torná-lo eficiente em um tempo de execução aceitável pelos oftalmologistas, o

¹ Disponível em: http://www.java.com/pt_BR/download/faq/whatis_j2me.xml. Acessado em: 19/07/2010.

algoritmo apresentado foi desenvolvido focando a otimização de todo processo de determinação do astigmatismo [BLOCH, 2008].

A seguir serão apresentadas informações sobre os tópicos discutidos, e que fazem parte do conteúdo necessário para o desenvolvimento do trabalho.

Capítulo 2. OBJETIVOS

O objetivo deste trabalho é desenvolver um *software* que faça a medida ceratométrica, e que possa ser executado em dispositivos móveis. Seu desenvolvimento está diretamente relacionado a um projeto cuja proposta é o desenvolvimento de um instrumento que incorpora o princípio de determinação do astigmatismo, com o diferencial de incorporar a tecnologia dos dispositivos móveis, que vêm conquistando o mercado de pequenos aparelhos como celulares, *Personal Digital Assistants* (PDAs) e mesmo câmeras fotográficas. O projeto visa unificar a grande precisão de um processo de determinação do astigmatismo à mobilidade permitida pelos novos processadores. O projeto original do anel de LEDs do sistema de lâmpadas de fenda precisará de modificações envolvendo LEDs adicionais a cada 90 graus para permitir a identificação e alinhamento da imagem.

O *software*, originalmente desenvolvido com a plataforma Delphi será aperfeiçoado consideravelmente para ser utilizado na linguagem Java. A razão é a versatilidade da linguagem Java, que permite a execução de programas em praticamente qualquer sistema computacional. Para esta vantagem paga-se o preço de um programa que se torna até 444% mais lento do que um programa escrito em C++ [MARTINS & BOTELHO, 2008], exigindo um algoritmo eficiente para a computação do astigmatismo em tempo real.

O objetivo é que no futuro seja desenvolvido um ceratômetro portátil.

Capítulo 3. A CÓRNEA E O ASTIGMATISMO

3.1. Córnea

A córnea é a parte anterior do olho e faz parte do sistema de focalização da luz através da pupila na retina. Sua característica de transparência e curvatura, composto de um tecido avascular, liso, com um formato frontal de uma elipse, são as características mais importantes para as funções fisiológicas da córnea. O raio de curvatura médio na superfície anterior da córnea em um adulto é de aproximadamente 7,8mm e 6,5mm na parte posterior, com uma curvatura superficial não uniforme, sendo mais acentuada na região central de 3 mm.

A Figura 3.1 ilustra as principais estruturas do olho humano [WANG, 2006].

Figura 3.1. O olho humano.²

² Adaptado de: <http://www.drsmarksandbanister.com/eyediseases.html>. Acessado em: 18/09/2009.

3.2. Poder de refração

Define-se refração como uma mudança da direção de propagação dos feixes de luz ao passar de um meio para outro cujas densidades são distintas e/ou o raio incidente não estiver penetrando no outro meio com um ângulo perpendicular a ele. Isso ocorre porque a velocidade com que a luz se propaga varia de um meio para o outro [OLIVEIRA et al., 2002].

O índice de refração (n) é caracterizado como a resistência que cada meio proporciona à propagação dos feixes. Os feixes que oferecem maior índice de refração são denominados meios mais refringentes, e seu cálculo está representado na equação (3.1),

$$n = \frac{c}{v} \quad (3.1)$$

onde c é a velocidade no vácuo e v é a velocidade da luz no meio em questão [MILÉO FILHO, 1996]. No caso do olho, a mudança é dada do ar para a córnea e seus índices de refração são, respectivamente, $n_{ar} = 1,0$ e $n_c = 1,3375$ [CARVALHO, 2005].

3.3. Ametropias oculares

Um olho é caracterizado como emétrepe (normal), quando focaliza os raios de luz provenientes de um ponto luminoso no infinito, sem fazer o uso da acomodação visual, em um único ponto, tendo como resultado uma visão bem definida. Nesse caso, a região central de 3mm da córnea é aproximadamente esférica.

Já as ametropias oculares ocorrem quando os feixes de luz que chegam paralelamente sobre o eixo visual, ao refratar, não focalizam sobre a retina, resultando em uma visão distorcida (borrada) [FERRON & RANCANO, 2007].

Os dois tipos mais básicos de erros de refração que são a miopia e a hipermetropia.

A miopia ocorre quando a imagem é formada previamente à retina, sendo, na maioria dos casos, devido à herança genética. E a característica marcante de um indivíduo míope é a deficiência em focar objetos muito distantes.

Já a hipermetropia pode ser considerada o inverso da miopia, ou seja, os feixes de luz passam pela retina antes que a imagem seja formada, devido à curvatura insuficiente da córnea. Com isso, o hipermetrope terá mais facilidade em focalizar objetos distantes.

3.4. Astigmatismo

O astigmatismo, por sua vez, ocorre quando a região central da córnea possui um formato que se assemelha a uma bola de futebol americano (Figura 3.2).

Figura 3.2. Região central da córnea e o astigmatismo.³

Este formato irregular causa diferentes distâncias focais de acordo com o plano de incidência da imagem, que não se forma em um único ponto da retina. É uma deficiência visual que resulta em imagens embaçadas, mas que pode ser corrigida com a utilização de lentes cilíndricas (tóricas) em óculos ou lentes de contato [DOME, 2001]. Cirurgia também

³ Adaptado de: http://www.tedmontgomery.com/the_eye/eyephotos/Astigmatism-grphc.html. Acessado em: 22/10/2010.

pode ser uma opção para corrigir o problema. Para qualquer um dos casos se faz necessário uma forma de medida do astigmatismo.

Capítulo 4. CERATÔMETROS

Ceratometria é a medida dos raios de curvatura da córnea na região central de 2 a 4 milímetros [WANG, 2006]. Medem-se duas curvas, a de maior e menor raio de curvatura, permitindo uma avaliação que pode indicar a presença de distorções na superfície. Entre suas utilidades também estão o acompanhamento da curvatura da córnea após cirurgia, determinação de lentes intraoculares quando necessárias, além de ajudar a prescrição de lentes de óculos e de contato. O ceratômetro, ou oftalmômetro, é um instrumento utilizado nestas avaliações, permitindo a medida da curvatura da superfície anterior da córnea [GONÇALVES & KRITZ, 1997]. Esta superfície pode ser considerada em termos óticos como um refletor especular. Neste caso, quando um objeto, em forma de anel circular, é colocado em frente a córnea, uma imagem virtual é formada, de acordo com a Figura 4.1, onde pode-se observar que seu tamanho depende do raio de curvatura da córnea.

Figura 4.1. Reflexão especular de um objeto h por um espelho convexo de raio R . A imagem y é menor que h , e virtual.

Esta reflexão é conhecida como a primeira imagem de Purkinje [DUCHOWSKI, 2007]. O astigmatismo é identificado quando a imagem do anel é refletida com uma alteração em sua forma, distorcendo o formato circular do objeto, resultando em uma elipse. A orientação

do astigmatismo pode ser determinada a partir da determinação dos eixos desta elipse. Estes eixos são meridianos ortogonais, e permitem a determinação do maior e menor poderes da córnea, conhecidos como valores-K, expressos em dioptrias (D). Esta unidade é comum em oftalmologia, e está relacionada com o raio de curvatura R pela equação (4.1).

$$D = \frac{(n_c - n_{ar})}{R} \quad (4.1)$$

O primeiro ceratômetro foi apresentado em Paris em 1728, e tem evoluído deste então em duas vertentes principais.

Um é o modelo Javal (Figura 4.2) com miras variáveis, e determinação individual de cada meridiano da córnea.

Figura 4.2. Modelo Javal.⁴

E o outro é o ceratômetro de Helmholtz que permite a determinação simultânea dos meridianos, e o tamanho da mira é fixo (Figura 4.3).

4

Disponível

em:

[http://www.oftalmics.com.br/?option=componente&comp=produtos&id=153&Itemid=18#top](http://www Oftalmics.com.br/?option=componente&comp=produtos&id=153&Itemid=18#top). Acessado em: 18/02/2010.

Figura 4.3. Ceratômetro de Helmholtz.⁵

Pode-se afirmar que melhorias significativas ocorreram desde 1854, quando Herman von Helmholtz desenvolveu seu primeiro ceratômetro. Atualmente, um número expressivo de empresas produzem esse equipamento como as internacionais Bausch&Lomb, Reichert, Topcon, e Zeiss. E também algumas nacionais, como a Eyetec, e a Opto.

A imagem da mira luminosa varia dependendo do modelo de ceratômetro utilizado. Um tipo de mira ceratométrica comum de ser encontrado é o de Bausch&Lomb, cuja imagem visualizada pelo operador do ceratômetro são três imagens refletidas, sendo cada uma dessas imagens composta por um círculo com um sinal de adição na posição de 3 e 9 horas, e um sinal negativo na posição de 6 e 12 horas, como mostra a Figura 4.4 [BRIGHTBILL et al., 2009].

⁵

Disponível

em:

<http://www.ofthalmics.com.br/?option=componente&comp=produtos&id=153&Itemid=18#top>.

Acessado

em: 18/02/2010.

Figura 4.4. Alinhamento da mira no processo de ceratometria. **a)** Mira em seu estado padrão. **b)** Mira em processo de alinhamento. **c)** Alinhamento com o primeiro valor K determinado. **d)** A imagem da mira com o segundo valor K determinado.

Neste tipo de mira de Bausch&Lomb o princípio é conhecido como princípio do duplo prisma, de forma que por um processo de birrefringência a imagem resultante seja a produção de duas imagens a partir de uma. As duas imagens e o objeto são observados e o deslocamento das duas imagens dependerá do poder ou da distância do prisma, do tamanho da mira, e da curvatura da córnea. No caso do ceratômetro de Bausch&Lomb, o duplo prisma é variável. Porém, há casos em que o duplo prisma é fixo, que é o caso do oftalmômetro de Javal-Shiötz [SCHOR, 1997].

O sistema do ceratômetro em estudo para o qual o *software* está sendo desenvolvido é baseado em um trabalho anterior [VENTURA et al., 2006], onde foi produzido um módulo ceratométrico para acoplamento na lâmpada de fenda de amplificação de 6 a 40 vezes.

Neste sistema um anel circular de LEDs é acoplado a uma lâmpada de fenda, e a imagem observada no sistema ótico é desviada por um divisor de feixes a um computador, onde a imagem da córnea do paciente é analisada. O princípio será detalhado no Capítulo 7.

Capítulo 5. HISTÓRICO DO MÓDULO CERATOMÉTRICO

A idéia de desenvolver um ceratômetro surgiu em 1996 com a necessidade de se desenvolver um ceratômetro que reunisse a qualidade de cada um dos modelos existentes até então, isto é, a alta precisão dos topógrafos e ceratômetros, um intervalo de medida de 32 a 100D, automatizado, a um baixo custo, e, principalmente, que fosse nacional.

As informações contidas neste capítulo foram extraídas da tese de livre docência da Profa. Dra. Liliane Ventura [VENTURA, 2007].

5.1. Primeiro Protótipo

No primeiro protótipo, foi desenvolvido um anel de luminosidade contínua de aproximadamente 15 mil fibras óticas, com um círculo luminoso de 42,2mm de diâmetro e o anel com dimensão total de 53,5mm de diâmetro. Esta medida é padrão para que a imagem refletida pelo anel luminoso na córnea sobreponha a região central de 3mm de diâmetro.

Foi realizado o acoplamento do anel luminoso de fibras óticas na objetiva da lâmpada de fenda, bem como um divisor de feixes, que transmite 70% de luz e reflete 30%. Este divisor foi colocado no caminho onde os feixes de luz saem paralelos (Lâmpada de Fenda do tipo Galileana). Além do acoplamento de um adaptador de vídeo no divisor de feixes. A imagem obtida pelo adaptador de vídeo é então capturada por uma câmera de vídeo do tipo CCD, que envia o vídeo para um computador por meio de uma placa de captura, onde havia um *software* instalado para processar a imagem (Figura 5.1).

Figura 5.1. Sistema completo do ceratômetro de lâmpada de Fenda [COELHO, 1999].

Além disso, um LED amarelo foi adaptado no adaptador óptico para fixação do olho do paciente para facilitar a centralização.

5.1.1. Software desenvolvido

Foi escolhido o desenvolvimento em Borland Delphi 5.0 por apresentar eficiência principalmente em relação ao tempo, desde o momento que executa até o momento de retornar os resultados.

Primeiro era feita a aquisição da imagem por meio do CCD. Esta imagem era então tratada com processamentos digitais de imagens de tal forma que fosse possível a determinação do contorno do anel luminoso. Em seguida, era feita a determinação da deformidade do anel e o cálculo do raio de curvatura em função da deformidade. Por fim, os resultados eram apresentados traçando-se os eixos de maior e menor raios e indicando o ângulo entre eles. Além disso, a apresentação era feita na notação clínica usual. E um banco de dados dedicado a este tipo de exame foi implementado.

Depois que a imagem foi capturada, foi feita a segmentação dessa imagem para identificar a borda do anel luminoso. Para isto, foi utilizada uma máscara gradiente e fez-se a varredura da imagem. Com isso, foi constatado que nem sempre a borda é contínua, devido a fatores externos como a sombra dos cílios, impedindo a uniformidade do anel. Isso ocorreu porque o tratamento era feito de forma linear, causando a comparação da borda com seus vizinhos laterais que poderiam não necessariamente ser borda.

Como possível solução, o tratamento seria feito utilizando o método de coordenadas polares. Com isso, foi realizada a determinação da borda de forma radial com sucesso.

Com a borda determinada é possível obter os pixels desta borda e determinar o centro de massa do anel, e com isso os eixos de maior e menor raios de curvatura da córnea do paciente.

Com este algoritmo foi possível fazer o cálculo tanto com eixos ortogonais, como com os não-ortogonais. Já apresentando uma vantagem em relação aos auto-refratores que não faz este tipo de cálculo.

5.1.2. Calibração

A calibração do sistema foi feita utilizando-se 5 esferas de aço com raios já conhecidos variando de 6 a 10mm, e capturadas 3 imagens de cada uma das 5 esferas.

Um teste foi feito para verificar a medida do raio de curvatura de cada ponto da esfera, devendo o gráfico, ilustrado na Figura 5.2, que a representa esboçar uma reta, uma vez que o objeto é uma esfera. Novamente o resultado foi positivo, apresentando para um raio de 8.08mm, uma variação de no mínimo 7.97mm e no máximo 8.11mm, dando uma média de 8.04mm.

Figura 5.2. Gráfico apresentando as medidas do raio encontrado pelo sistema de uma esfera de raio conhecido de 8.08mm a cada 1°. [VENTURA et al., 2000a]

5.1.3. Interface do software

A tela de resultados do primeiro protótipo mostra os eixos de maior e menor raios de curvatura encontrados, além de outra tela com uma análise do astigmatismo encontrado como mostra a Figura 5.3.

Figura 5.3. Tela de apresentação contendo os eixos de maior e menor raios (lado esquerdo) e tela com uma análise do astigmatismo encontrado (lado direito) [Ventura et al., 2000].

5.1.4. Considerações finais do primeiro protótipo

Em relação ao *hardware*, para facilitar o exame do clínico, foi desenvolvido um pedal, conectado em paralelo ao botão do *mouse*, por meio de uma chave simples *push botton*, obtendo a mesma função. Este dispositivo a mais era útil para que o médico não tivesse que desviar as mãos ou o olhar do microscópio ocular para clicar no *mouse*.

O banco de dados foi desenvolvido para que, além das características comuns de abrir, fechar, armazenar e imprimir arquivos, também dê ao clínico a opção de cadastrar os clientes, agendar consultas, cadastrar novos usuários e técnicos, entre outros.

Porém, com o tempo percebeu-se que não era necessário ter um banco de dados, uma vez que apenas a funcionalidade de armazenar imagens e resultados era necessário, podendo ser implementado no próprio *software* de medição ceratométrica.

O sistema apresentou resultados satisfatórios comparado aos outros ceratômetros.

Alguns testes foram feitos com imagens bem fora dos padrões (borradas, córnea de pacientes pós-operatório, etc.), e o resultado obtido foi a determinação correta dos eixos de curvatura.

Porém, foram encontrados alguns problemas que levaram a realizar algumas mudanças, como a centralização feita pelos usuários do sistema, que mostraram diferenças consideráveis de um para outro.

Outras necessidades de mudança foram observadas, como o anel de fibras ópticas que apresentou não homogeneidade luminosa em sua confecção. Além disso, seria preciso diminuir a quantidade de fios e cabos que ficavam em cima da lâmpada de fenda; substituir o pedal pelo botão do *joystick* da lâmpada de fenda; desenvolver um sensor para detectar o lado do olho sob análise (o médico precisa preencher no teclado ou usar o *mouse*); eliminar o sistema de banco de dados; e por fim, melhorar o sistema de centralização da mira com o olho do paciente.

5.2. Segundo Protótipo

No segundo protótipo, desenvolvido por COELHO e VENTURA, o anel de fibras ópticas foi substituído por um anel de LEDs [COELHO, 1999].

Foram feitas pesquisas para determinar a melhor dimensão do anel, quantos LEDs seriam necessários e qual a melhor cor para utilizar.

O anel é de *nylon* preto, principalmente por ser leve, isolante, mais barato, e de fácil fabricação. Foi determinado para o anel a dimensão de 83,5mm de diâmetro, e um LED amarelo foi acoplado no centro do anel para fixar o olho do paciente.

Depois de vários testes e comparações, escolheu-se utilizar 36 LEDs, pois, um número menor causaria muita imprecisão, e um número maior deixaria o processamento muito lento. Utilizar 72 LEDs seria ideal para a precisão, mas deixou o processamento 3 vezes mais lento, o que o inviabilizaria.

Na parte óptica do segundo protótipo, foi utilizado um dispositivo único que integrasse o divisor de feixes e o adaptador de vídeo, transmitindo 60% e refletindo 40%. Esta substituição rendeu um custo 40% menor que o anterior.

Além disso, o detector CCD que era colorido foi substituído por um monocromático, gerando um custo 80% menor que o anterior.

Foi desenvolvido um circuito que, ao acionar o *software* (geralmente movimentando a lâmpada de fenda), acendesse o anel automaticamente, fazendo o LED central amarelo piscar, e ativar o pedal, os sensores e o CCD.

Ao apertar o pedal, o resultado era apresentado na tela em 1s. Todo o circuito era então desligado até que se movimentasse a lâmpada de fenda novamente. Além disso, este novo *software* oferecia a opção de imprimir o laudo, clicando em um botão.

5.2.1. Software Desenvolvido

Quando a imagem era capturada, iniciava-se o processamento digital desta imagem. Para isto, o primeiro passo foi transformar a imagem em uma matriz $Im[x, y]$, em que cada elemento representava um pixel da imagem contendo o valor do brilho daquele pixel. O valor do brilho é dado em escala de cinza, ou seja, varia no intervalo de 0 a 255.

Para ter uma noção da intensidade do brilho de cada LED, foi esboçado um gráfico tridimensional, onde a coordenada de profundidade apresentava o brilho ao longo da imagem, como mostra a Figura 5.4.

Figura 5.4. Representação tridimensional da intensidade do brilho da imagem [VENTURA et al., 2007]

Apesar da não uniformidade do brilho, o algoritmo apresentado mostrou que não influencia na determinação dos eixos.

No tratamento da imagem, o primeiro passo foi a binarização dessa imagem, ou seja, transformá-la em branca (255) ou preta (0).

Muitas vezes a imagem mesmo depois de binarizada apresentava ruído, então um processo de abertura e fechamento foi utilizado para ajudar na remoção desse ruído.

Para identificar os LEDs, era feita uma varredura linear na imagem até encontrar um valor $Im_0[x_0, y_0] = 255$. Então, iniciava-se o processo de FLOOD. Neste processo, o ponto $P(x_0, y_0)$ era marcado por um identificador de intensidade intermediária I_0 , sendo o valor de I_0 diferente de 0 e 255. Cada vizinho imediato de P era checado. E, se seu valor fosse igual a 255, ele também era marcado.

Como resultado, todos os pixels pertencentes ao LED eram transformados para intensidade I_0 .

Sabendo-se a posição de cada pixel, era possível determinar o centro de massa desse LED, armazenado como $x[i], y[i]$.

E assim era feito até encontrar todos os LEDs possíveis.

Uma vez determinado o centro de massa de cada LED, pelo método FLOOD era possível encontrar o centro de massa da imagem armazenado como x_{cm0}, y_{cm0} .

Nem sempre o centro de massa da imagem coincidia com o da elipse determinada que melhor compreendesse o anel de LEDs. Para se resolver este problema, um programa de interpolação traçava uma curva “contínua” que passasse pelos pontos dos LEDs que foram identificados. Neste caso, como o anel não era contínuo, o termo referia-se à seqüência de pontos de pixels que fosse conexa.

Então, o centro de massa do LED transformava-se em coordenadas polares.

Para representar a melhor elipse que compreendesse todos os pixels encontrados, foi utilizado o cálculo do menor desvio quadrático (S), mostrado na equação (5.1).

Esta elipse possui um ângulo de inclinação φ em relação ao eixo horizontal.

Para o cálculo foi utilizado os pontos $r[i]$ e $\vartheta[i]$ encontrados na interpolação, onde:

- $\vartheta[i] = n[i] \cdot \vartheta_0$
- $n[i] = 0, 1, 2, \dots, 359$
- $\vartheta_0 = 1^\circ$

Definiu-se, então, uma elipse de teste com ângulo de desvio $\varphi = n[i] \cdot \vartheta_0$, e $n[i]$ variando de 0 a 90°.

Para cada valor de $n[i]$, calculou-se o desvio quadrático médio entre os pontos definidos pela elipse e os pontos interpolados:

$$S = \sum_{i=0}^{359} (r_i - r)^2 \quad (5.1)$$

onde,

$r_i \rightarrow$ raio obtido em função de $\vartheta[i]$

$r \rightarrow$ raio da elipse dado por $x = r \cdot \cos(\vartheta[i])$ e $y = r \cdot \sin(\vartheta[i])$

E, por fim, o menor valor de S , ou seja, o menor desvio quadrático determina o maior e menor eixos, além do ângulo de inclinação.

5.2.2. Calibração do Sistema

A melhora que ocorreu em comparação ao anterior é que foi introduzido um botão de calibração automática no *software*.

5.2.3. Interface do software

A diferença em relação ao do primeiro protótipo é a simplificação da interface, além da substituição do teclado e *mouse* por pedal e/ou o botão do *joystick* da lâmpada de fenda.

5.2.4. Considerações finais do segundo protótipo

Para alguns dos testes entre os 492 pacientes testados foram chamados dois voluntários, um com baixo astigmatismo e outro com astigmatismo considerável. Foram feitas medidas em 9 ceratômetros automáticos e 1 manual, e todos por um mesmo clínico.

Como resultados, para o paciente com astigmatismo considerável, as medidas apresentaram até 7º de diferença nos eixos. E para o paciente com baixo astigmatismo, as medidas apresentaram até 5º de diferença.

Com isso pôde-se observar que quanto menor o astigmatismo, maior a discrepância entre as medidas, e concluiu-se que a causa se devia à imprecisão dos ceratômetros que utilizavam apenas 6 pontos luminosos.

Concluiu-se então que o anel de LEDs comparado ao de fibras ópticas apresentou mais vantagens que desvantagens. A desvantagem é que a precisão é menor, por causa do espaçamento de 10º, enquanto as fibras ópticas possuem um anel contínuo nos 360º examinados.

Porém, como vantagens é possível citar a maior leveza do módulo, menor custo, e melhor padrão de iluminação com LEDs.

5.3. Terceiro Protótipo

O terceiro protótipo permaneceu com 36 LEDs que foram alojados em um disco de PVC preto.

A principal diferença em relação ao anterior é que antes os LEDs ficavam soldados por fios, o que dificultava a manutenção, além de toda a eletrônica (capacitores, resistores,

transistores, etc.) que ficava alojada em uma pequena caixa externa ao anel de LEDs, apresentando um manejo incômodo da caixa e difícil manutenção dos circuitos.

A mudança foi a utilização de uma porta paralela para interface entre *hardware* e *software*, dispensando o uso de protocolo como o da Piccolo e conseqüentemente economizando gastos.

Esta porta envia as informações de forma paralela, e suas principais características são:

- Saída paralela de 8 bits
- Entrada paralela ou sinais de controle de 5 bits
- Uso de conectores DB25 e Centronics

Outra novidade neste protótipo foi o desenvolvimento do sensor de olhos, que detecta qual olho está sendo examinado.

Seu circuito possui dois dispositivos, um LED que emite infravermelho e um fotodiodo que capta sinais. Além disso, uma fita foi fixada no meio do caminho percorrido pela lâmpada de fenda. Esta faixa possui uma divisão, sendo metade fosca.

Se o sinal for captado pelo fotodiodo, o olho medido é o esquerdo, caso contrário, o emissor está do lado fosco da fita, e o olho medido é o direito (ou vice-versa, de acordo com a definição no algoritmo).

Vários testes foram feitos sobre a possibilidade de o sensor ter interferência com o meio ambiente, e constatou-se que o sensor só é interferido se existir uma luz intensa e próxima a ele, o que não ocorre em clínicas oftálmicas.

5.3.1. Calibração do Sistema

Outra novidade foi a haste de alumínio desenvolvida com suporte em PVC preto para colocar a esfera de calibração.

5.3.2. Software desenvolvido

O software desenvolvido continuou parecido com o anterior, apenas com alguns ajustes, como não binarizar mais as imagens, e sim utilizar imagens em escala de cinza.

Além disso, a máscara utilizada para a identificação do LED de maior intensidade passou de formato quadrado para formato circular, com diâmetro igual ao tamanho médio observado para os LEDs. Com isso, o centro de massa do LED passou a ser o centro da máscara circular, e, uma vez identificados os centros de massa dos LEDs foi possível identificar o centro do anel. Para isso, foi utilizado o mesmo processo de interpolação da curva mediana do anel. E por fim foi possível a representação da melhor elipse que se encaixa no anel, também por meio do mesmo processo anteriormente utilizado (cálculo do menor desvio quadrático).

5.3.3. Interface do software

A diferença em relação ao do segundo protótipo é a organização da interface, que foi padronizada, para facilitar o uso por clínicos que estão acostumados a utilizar os outros existentes.

Além disso, foi implementada uma mira no *software*, no centro da tela, em formato quadrado com tamanho próximo ao diâmetro do LED central, onde aparece a imagem que será adquirida, cujo propósito é melhorar a centralização do sistema, como é mostrada na Figura 5.5.

Figura 5.5. Tela de apresentação do protótipo fixal com mira para centralização do sistema [VENTURA et al., 2007].

Em relação ao laudo gerado, nesta última versão o clínico tem a opção de visualizar na tela, imprimir, ou até mesmo ouvir.

Outras novidades é que, ao clicar com o botão esquerdo do *mouse*, aparece uma tela que mostra o poder de refração da superfície analisada, e outra tela de comparação é aberta para comparar os resultados com os outros sistemas existentes no mercado.

5.4. Protótipo Final

5.4.1. Anel ótico sem LEDs

No caso do ceratômetro para Lâmpada de Fenda, a diminuição de cabos, e outros componentes eletrônicos eram de fundamental importância para a melhoria interface de uso.

Assim, foi construída uma nova mira de projeção, composta essencialmente por furos e partes mecânicas polidas para reflexões óticas, que aproveitavam a luz da própria Lâmpada de Fenda e a transformava em 72 pontos luminosos homogêneos para projeção.

A Figura 5.6 ilustra as vistas frontal e lateral desta nova mira de projeção e também a imagem refletida por uma esfera de calibração.

Figura 5.6. Mira de 72 furos iluminada pela própria iluminação da lâmpada de fenda: (A) Visão frontal; (B) Visão lateral; (C) Imagem dos furos luminosos projetados na córnea [VENTURA et al., 2007].

Esta é a mira de projeção utilizada até os dias atuais na Lâmpada de Fenda.

Capítulo 6. DISPOSITIVOS MÓVEIS

O crescente avanço tecnológico na área da mobilidade tem atraído cada vez mais usuários para este novo conceito, e transformado a maneira como os usuários vêem a tecnologia. Isso inclui o uso desses dispositivos móveis por profissionais em seu trabalho.

Segundo a União Internacional de Telecomunicações (ITU), sua última pesquisa realizada em 2009 mostrou que foram alcançadas cerca de 68,2 bilhões de novas assinaturas de telefones celulares no mundo, comprovando a aceitação crescente da população nessa área (Figura 6.1).

Figura 6.1. Estatística segundo a ITU sobre o avanço do uso de telefones móveis.⁶

⁶ Disponível em: <http://www.itu.int/ITU-D/ict/statistics/index.html>. Acessado em: 07/08/2010.

Isto vem ocorrendo devido às vantagens que estes dispositivos trazem em características como a ausência de fios conhecido como tecnologia *wireless*, as diversas funcionalidades como agenda, calculadora, telefone, o acesso à Internet, televisão, e o fato de ser um telefone portátil.

Devido à expressiva concorrência no mercado globalizado, o número de dispositivos móveis aumentou, e com isso os preços se tornaram cada vez mais atrativos e acessíveis para usuários em potencial.

Entre estes dispositivos móveis (Figura 6.2) podem ser citados *Personal Digital Assistants* (PDA's), telefones celulares, mini *notebooks*, *smartphones*, *tablets*, entre outros.

Figura 6.2 Exemplo de dispositivos portáteis.

Dentro de cada um destes dispositivos móveis, há uma complexa computação envolvida, podendo estes dispositivos, também, serem chamados de computadores móveis.

6.1. PDA's

Os primeiros⁷ PDA's surgiram em meados dos anos 80, apresentando basicamente as funcionalidades de calculadora, alarme, calendário e agenda eletrônica; além de um pequeno teclado para entrada e uma tela pequena.

A principal função de um PDA, até então, era sua agenda eletrônica portátil que auxilia na organização de um planejamento. Porém, esta funcionalidade deixou de ser uma das únicas funções de um PDA para os usuários. É possível encontrar no mercado PDA's que além de possuir funções básicas como tocar músicas, são receptores GPS e, permitem a utilização de programas desenvolvidos por usuários.

Além das crescentes funcionalidades que estão surgindo para os PDA's, em paralelo estão surgindo variedades de plataformas para estes PDA's. Toda essa variedade em um só produto tem feito surgir dúvidas entre os usuários sobre os diferentes tipos de plataforma existentes. Segundo [MONTEIRO, 2006], é comum a confusão que existe sobre os PDA's quando relacionado às plataformas mais conhecidas *Palm* e *PocketPC*. A plataforma *Palm* utiliza o sistema operacional *Palm OS*, e a plataforma *Pocket PC* da *Microsoft*⁸ utiliza os sistemas operacionais *Windows CE*, *Pocket PC 2003*, *Windows Mobile*, entre outros.

Entre as funcionalidades que o *Palm OS* possui estão reconhecimento de escrita *graffiti*, uma grande quantidade de bibliotecas que podem ser adicionadas ao sistema, sincronização tanto com o sistema operacional *Windows* quanto *Macintosh*, e aplicativos *DataViz Documents to Go* (semelhantes ao *Microsoft Word*, *Excel* e *Outlook*).

⁷ Disponível em: <http://electronics.howstuffworks.com/gadgets/travel/pda.htm>. Acessado em: 18/02/2010.

⁸ Disponível em: <http://www.microsoft.com/windowsmobile/en-us/devices/default.mspx>. Acessado em: 18/02/2010.

Já em relação ao sistema operacional da *Microsoft* para a plataforma *Pocket PC*, algumas das funcionalidades são versões para *Pocket* da *Microsoft Word*, *Excel* e *Outlook*, porém, com algumas formatações a menos; sincronização com o *Microsoft Outlook* em um *PC* que tenha o sistema operacional *Windows*; três aplicativos de reconhecimento de escrita: *Transcriber*, *Letter Recognizer* e *Block Recognizer* e, *Windows Media Player* para conteúdo multimídia.

Uma das maiores vantagens dos PDA's é que está se tornando cada vez mais obrigatório, devido à pressão da concorrência, números crescentes de funcionalidades, como *bluetooth* e receptores GPS, aproximando-os cada vez mais de *PCs* completos. Além disso, o peso de um PDA pode variar de menos de 150 gramas a mais de 1,0 kg. Junto a isso, pode-se incluir a tela *touchscreen*, utilizada tanto para entrada como saída de dados.

Uma das principais desvantagens é não possuir um disco rígido, apenas uma memória de somente leitura (ROM) e uma memória de acesso aleatório (RAM), com exceção de alguns PDAs que possuem memória *flash*, na qual os dados não são perdidos mesmo quando termina a bateria, sendo um substituto do disco rígido. Porém, sua capacidade de armazenamento costuma ser muito pequena comparada a dos discos rígidos. Por isso, é essencial o *backup* ou a sincronização constantemente, não podendo, por enquanto, ser um substituto de um *PC*, apenas uma extensão. Além disso, de forma geral os PDAs possuem processadores que variam em média de 312 MHz a 528 MHz, valores muito baixos para utilizar recursos de Processamento Digital de Imagens mais complexos.

6.2. Mini Notebooks

Para que os *notebooks* possam atender ao propósito deste trabalho, o ideal é que se trabalhe apenas com os mini-notebooks, muito mais leves para carregar que os *notebooks* convencionais.

Para a aquisição dos dados⁹ de mini-*notebooks* podem ser utilizados como exemplo o modelo da Asus Eee PC 701, da Fujitsu U810, da HP 2133 Mini-Note PC, e da SONY VAIO® UX Micro PC.

O primeiro protótipo de *notebook* foi desenvolvido nos anos 70, por Alan Kay, da Xerox¹⁰ e, o primeiro *notebook*, em 1979, por William Moggridge da *Grid Systems Corporation*.

Os *notebooks*, em sua maioria, operam com o sistema operacional Windows XP ou Windows Vista, mas é possível encontrar alguns compatíveis para o sistema operacional Linux.

A grande vantagem dos mini-*notebooks* é que eles podem ser considerados computadores completos. Eles possuem em sua maioria disco rígido que pode variar desde 40 Gb a 160 Gb, com exceção do ASUS EEE PC 710, que possui memória *flash* variando de 4 Gb a 8 Gb substituindo o disco rígido. Em relação ao peso de um mini-*notebook*, a variação é de 680 gramas a 1,27 quilogramas. Além disso, a velocidade dos processadores, em média, varia de 800MHz a 1.6 GHz, o que é muito útil para utilização de recursos de PDI.

Sua desvantagem está no fato de serem bem mais pesados se comparados aos PDAs e *smartphones*. Outra desvantagem está em relação à duração da bateria. Enquanto nos PDAs as baterias podem durar em média oito horas, em mini-*notebooks* dificilmente

⁹ Dados coletados dia 22/01/2010.

¹⁰ Disponível em: <http://communication.howstuffworks.com/laptop6.htm>. Acessado em: 22/01/2010.

ultrapassa cinco horas e meia em modo *stand by* (modo de espera para economizar energia, em que o processador reduz ao mínimo sua capacidade de processamento, o *display* é desligado, e o dispositivo fica esperando uma intervenção do usuário).

6.3. Tablets

Os tablets podem ser comparados com os notebooks, porém, não possuem teclado para entrada de dados.

Seu diferencial é que o usuário utiliza uma caneta metálica para escrever a mão diretamente sobre a tela. Esta caneta possui uma tinta digital que permite escrever e grifar, além de servir como mouse.

Para exemplificar, foram utilizados dados coletados¹¹ de um *Tablet PC Panasonic Toughbook CF-08* e *Tablet PC T221* da Yuma.

O *Tablet PC IP67-07*¹² da Yuma possui tela de 7 polegadas WSVGA (*WideScreen Video Graphics Array*) TFT (*Thin-film transistor*) 1024 x 600, processador Intel Atom Z530 de 1.06GHz, 32 GB SSD, 1 GB de memória RAM, BlueTooth, e pesa 1,2 kg.

Já o *Tablet PC Toughbook CF-08* da Panasonic¹³ roda o sistema Windows CE, e tem uma bateria que dura de 4 horas e meia até 14 horas, dependendo do brilho da tela. A tela *touchscreen* tem 10.4" e uma resolução de 1024 x 768 pixels. Seu processador XSCALE tem apenas 312 MHz, só tem 128 MB de memória, e pesa 1,2 kg. Um slot de cartões de memória SD aumenta a capacidade para até 2 GB. Ele ainda possui tecnologia Wi-Fi, Bluetooth e Ethernet.

¹¹ Dados coletados dia 22/01/2010.

¹² Disponível em: <http://www.belsatia.com/pagina.asp?0=4&1=242790&2=21669&3=49845>. Acessado em: 22/01/2010.

¹³ Disponível em: <http://digitaldrops.com.br/drops/2007/04/panasonic-toughbook-cf-08-um-tablet-duro-de-matar.html>. Acessado em: 22/01/2010.

6.4. Smartphones

Os *smartphones*, segundo [SILVA, 2007], podem ser considerados PDA's, ou pelo menos possuem a maioria da funcionalidade de um, adicionando a capacidade de comunicação de telefones celulares.

Por isso, eles possuem as especificações idênticas as dos PDA's, com a vantagem do serviço telefônico, incluindo recentemente, a tecnologia 3G em vários modelos. Além disso, é possível encontrar um maior número de *smartphones* com câmeras digitais de resolução maior que a maioria dos PDA's.

6.5. Processadores dos dispositivos móveis

A arquitetura mais comum entre os dispositivos móveis é a arquitetura Advanced RISC Machine (ARM). Desenvolvida especificamente para sistemas embarcados, essa arquitetura possibilita implementações enxutas, sem prejudicar o desempenho, graças à simplicidade do processador. Além disso, quanto menor a implementação, menor o consumo de energia, o que os torna interessante para os dispositivos móveis.

Capítulo 7. PROCESSAMENTO DIGITAL DE IMAGENS

7.1. Introdução ao Processamento Digital de Imagens

Com o progresso da tecnologia e conseqüentemente a rápida evolução dos processadores, a partir dos anos 60 começou a surgir a possibilidade de se trabalhar com o tratamento digital das imagens [COSTA et al., 2009].

O Processamento Digital de Imagens (PDI) pode ser dividido em dois propósitos. Um é melhorar o aspecto visual da imagem de forma a melhorar a percepção humana, incluindo a impressão e transmissão dessas imagens. E o outro propósito é a preparação dessas imagens para a medição das características e estruturas contida nessas imagens, que é o propósito utilizado neste trabalho [RUSS, 2007].

7.2. Representação de imagens digitais

A representação das imagens digitais é dada pela função bidimensional $f(x,y)$, onde x e y são os pontos no plano formando pares de coordenadas (x,y) que representam a intensidade do nível de cinza da imagem em cada ponto desse plano. Por uma questão de convenção, a numeração das coordenadas tem início no canto superior esquerdo, como mostra a Figura 7.1.

Figura 7.1 Imagem digital representada pelas coordenadas cartesianas.

De uma forma mais específica, para que a imagem digital possa ser manipulada, ela é transformada em uma matriz.

Matriz pode ser definida como uma tabela $m \times n$, em que m é o número de linhas, e n o número de colunas dessa tabela.

Cada elemento dessa tabela contém um valor inteiro e positivo variando no intervalo de 0 a 255. Neste trabalho convencionou-se que 0 corresponderá a cor preta e 255 a cor branca, representando o valor da intensidade da cor da imagem naquele ponto.

Estes elementos são comumente conhecidos como *pixel*, que surgiu da aglutinação das palavras inglesas *picture* e *element*. Cada pixel contido na imagem representa a intensidade da cor da imagem naquele ponto. Esta intensidade pode ser representada de forma binária, com 0's e 1's (ou seja, preta e branca), em níveis de cinza, ou ainda coloridas.

7.3. Profundidade de Bits

Cada pixel armazena uma tonalidade, como o branco, preto, tonalidades de cinza, ou de cor, às quais são representadas em notação binária. Ou ainda pode-se afirmar que o número de bits utilizados para definir cada pixel e sua tonalidade é conhecido como

profundidade de bits, ou seja, quanto maior a profundidade de bits, maior é o número de tons representados. Por exemplo, se a profundidade de bits for igual a 1, a imagem está em preto e branco. Em uma imagem com profundidade de 8 bits, a imagem está em escala de cinza. E uma imagem com profundidade de 24 bits, a imagem está colorida [COSTA et al., 2009].

7.4. Máscara

A máscara (também conhecida como molde, filtro, *template* ou janela) é representada por uma matriz de tamanho $m \times n$, em que m é o número de linhas, e n o número de colunas.

O objetivo da máscara é aplicá-la sobre a matriz da imagem de forma que os valores dos pixels da máscara, quando aplicados sobre os pixels da imagem, produzam resultados do tipo redução de ruído, reforço das bordas, suavização, entre outros.

A Figura 7.2 exemplifica este conceito, em que geralmente a máscara é de ordem 3. O elemento do meio $w(0,0)$ é posicionado sobre o pixel (x,y) da imagem que receberá o novo valor, e cada elemento da máscara é multiplicado com o valor do pixel correspondente da imagem e esses resultados são somados como mostra a equação (7.1).

$$R = w_1 z_1 + w_2 z_2 + \dots + w_{mn} z_{mn} = \sum_{i,j=1}^{mn} w_i z_j \quad (7.1)$$

Nesta equação os w 's são os pixels da máscara, os z 's são os pixels da imagem (onde, para cada valor de w , há o seu correspondente z na imagem), e mn é o número total de elementos da máscara [GONZALEZ & WOODS, 2003].

Figura 7.2 Exemplo de aplicação de máscara. Adaptado de Gonzalez (2003).

Neste trabalho, o objetivo da máscara é ser aplicada na imagem de forma a identificar os LEDs da imagem. Para isso, a máscara percorre a imagem de forma estratégica para encontrar o pixel de maior intensidade em uma dada região, e esse pixel será considerado o ponto central do LED. Esta técnica será apresentada com mais detalhes no capítulo 7.

Capítulo 8. METODOLOGIA

8.1. Mira ceratométrica

Para o modelo ceratométrico proposto, o objeto luminoso que será projetado sobre a córnea é composto por uma distribuição circular de LEDs, formando um anel luminoso, como ilustrado na Figura 8.1.

Esta mira foi a escolhida para ser utilizada neste projeto, pois pretende-se desenvolver um ceratômetro portátil, cujo anel de projeção é composto por LEDs.

O *hardware* do sistema consiste em 36 LEDs (de diâmetro $\Phi=5\text{mm}$) dispostos em um formato de círculo, e 12 LEDs (de diâmetro $\Phi=3\text{mm}$) para alinhamento e orientação. Os LEDs são projetados na córnea e a imagem refletida será capturada por um sensor CCD localizado depois de um adaptador óptico de magnificação de 10 vezes.

Figura 8.1. Mira luminosa.

A imagem é enviada para o computador portátil por um cabo USB, e o *software* desenvolvido processa a imagem de forma tão eficiente quanto um ceratômetro convencional. O algoritmo identifica a coordenada central de cada LED identificado, e converte essas coordenadas em coordenadas polares relativas ao centro da imagem $c(x_0, y_0)$, por todo o círculo. Assim, considerando as coordenadas cartesianas x e y , as coordenadas polares são determinadas de acordo com as fórmulas (8.1) e (8.2).

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2} \quad (8.1)$$

$$\theta = \arctg\left(\frac{y - y_0}{x - x_0}\right) \quad (8.2)$$

Os pontos centrais de cada LED são identificados buscando a máxima verossimilhança com relação a uma máscara pré-determinada, e são então submetidos a uma verificação para que se eliminem possíveis interferências como ruídos na imagem.

As coordenadas identificadas para os LEDs são organizadas em forma crescente com relação ao ângulo, e então são utilizadas em um processo de interpolação polinomial utilizando o algoritmo de Neville, implementado para gerar um anel formado por um ponto por grau, produzindo um anel de 360 pontos igualmente espaçados, utilizados para a análise do raio de curvatura [PRESS et al., 1996].

A interpolação utilizada para determinar a posição do anel para cada grau envolve seis pontos de entrada correspondentes às coordenadas polares dos LEDs identificados, três anteriores e três posteriores.

8.2. Modelo Matemático

Para os propósitos da ceratometria, somente a região central da córnea (3mm de diâmetro) é considerada. O modelo esférico, que é mais simples de ser implementado que o modelo elíptico, é suficiente e exato, mas para pequenos raios de curvatura, nenhuma aproximação deve ser feita, como foi estudado no desenvolvimento e teste do primeiro sistema de ceratometria para lâmpada de fenda [VENTURA et al., 2006].

A região central da córnea onde a mira de luz circular é projetada é esférica, e pode ser comparada a um espelho convexo. A Figura 8.2 mostra um esquema do princípio ceratométrico.

Um objeto de tamanho h (raio da mira circular) é posicionado a uma distância a da córnea, onde é perfeitamente focalizado junto à córnea. O reflexo do objeto h produz uma imagem virtual y , no qual é metade da dimensão do alvo projetado na córnea (aproximadamente 1,5mm) e está a uma distância b da superfície da córnea. A distância d é de 200 mm para o microscópio ocular cirúrgico utilizado. A distância b é a distância entre a superfície da córnea e a atual posição da imagem virtual do objeto (mira projetada).

Figura 8.2. Diagrama que esquematiza o princípio ceratométrico.

Considerando-se que os feixes luminosos provenientes de h incidam no eixo óptico com ângulos pequenos, é possível dizer que a distância focal é metade do raio de curvatura da córnea (r).

O raio de curvatura da superfície a ser determinado é mostrado na equação (8.3).

$$r = \frac{2 d h y}{h^2 - y^2} \quad (8.3)$$

Capítulo 9. **DENVOLVIMENTO DO SOFTWARE**

9.1. Linguagem de Programação Java e a *Java Virtual Machine* (JVM)

Segundo Costa (2009), para processamento digital de imagens geralmente a linguagem que mais se utiliza é a linguagem C, pela simplicidade e agilidade na execução de seus algoritmos. Porém, assim como Java, outras linguagens estão sendo desenvolvidas para que sejam utilizadas tanto em processamento digital de imagens como também para outros propósitos.

As linguagens de programação podem ser divididas em três tipos gerais: linguagens de máquina, linguagens assembly e linguagens de alto nível. A linguagem de máquina é aquela diretamente compreensível pelo computador, geralmente instruções reduzidas a 0s e 1s.

Entretanto, essa linguagem é muito complexa para o ser humano, que como solução inicial desenvolveu uma linguagem intermediária contendo abreviações em inglês para representar as instruções, como a palavra add e store por exemplo. Para que o computador entenda essas abreviações, os programadores utilizam os programas tradutores assemblers, que faz a conversão da linguagem assembly para a linguagem de máquina.

A princípio a linguagem assembly foi aceita pelos programadores com sucesso. Mas com o tempo e a evolução do *hardware*, os programadores começaram a sentir necessidade de acelerar o processo de programação, pois a linguagem assembly apesar de ter facilitado em termos de compreensão, exigia que muitas instruções fossem inseridas para se chegar aos resultados mais simples.

Com isso, foram desenvolvidas as linguagens de alto nível, que se aproximam ainda mais da linguagem humana, e reduziu consideravelmente o número de instruções necessárias para se chegar aos resultados desejados. E para que a máquina possa entender essa linguagem, de forma resumida pode-se dizer que foram desenvolvidos os interpretadores, que são ambientes onde os programas são desenvolvidos e corrigidos para não sobrecarregar os compiladores, que convertem a linguagem de alto nível diretamente em linguagem de máquina [DEITEL, 2007].

No caso, a linguagem *Java Standard Edition* (JavaSE) foi a escolhida para desenvolver esse trabalho pelo diferencial de ser executada em uma máquina virtual chamada *Java Virtual Machine* (JVM).

Ao compilar o código-fonte desenvolvido em Java, é gerado um arquivo `.class` na qual as instruções são convertidas em *bytecodes*, ao invés de serem convertidas em linguagem de máquina, e esses *bytecodes* por sua vez são executados na JVM. Ao contrário da linguagem de máquina, os *bytecodes* não são dependentes de plataforma e de *hardware*, pois a JVM simula um computador, ocultando o sistema operacional e o *hardware* dos programas subjacentes que interagem com ela. Ou seja, para executar um programa Java chamado Pessoa, por exemplo, o arquivo Pessoa.java deve ser executado para invocar a JVM, que então inicia o processo.

O primeiro passo da JVM é armazenar os arquivos `.class` (que contém os *bytecodes*) na memória do computador. Em seguida ela verifica se o conteúdo é realmente seguro e se não viola as restrições de segurança do Java, e finalmente os *bytecodes* são executados.

Além da linguagem Java, há uma extensão chamada *Java Platform, Micro Edition* (JavaME), que permite o desenvolvimento de aplicativos para sistemas embarcados e dispositivos móveis de baixo poder de processamento e pouca memória.

Na segunda etapa deste trabalho esta tecnologia foi utilizada para desenvolver um novo *software* com o mesmo princípio que o anterior, porém que seja capaz de ser executado em dispositivos móveis.

Além da vantagem de não depender de plataforma, Java também apresenta uma forma organizada de documentação, na qual cada declaração como classes, métodos e variáveis de instância, pode ser documentada e pode-se gerar o chamado Javadoc, uma documentação Java que resulta em uma *Application Programming Interface* (API), que é uma biblioteca de classes Java que contém informações resumidas sobre seus métodos, construtores e parâmetros. Essa documentação é útil para um propósito que os desenvolvedores Java, e outros com .Net, defendem o chamado reuso de código. Isto é, reutilizar classes e interfaces que já foram desenvolvidas e que já estão com o algoritmo otimizado e estruturado pra o melhor desempenho, principalmente para economizar tempo tanto de processamento, como de desenvolvimentos dos aplicativos em si. Essas bibliotecas são oferecidas na maioria das vezes por fornecedores de compilador, e também dos chamados *Independent Software Vendors* (ISV).

Outra vantagem que pode ser citada, é que, com a explosão da popularidade da *World Wide Web* em 1993, a equipe da Sun (criadores do Java) viu a oportunidade de desenvolver conteúdos com interatividade e animações nas páginas da Web. Assim, até o momento Java se destaca nas aplicações desenvolvidas voltadas para a Web.

9.2. Java Image, Java Advanced Imaging e o Processamento Digital de Imagens

Em virtude do processamento digital de imagens estar amplamente presente na Web, várias APIs que trabalham com imagens já foram desenvolvidas. Neste trabalho foi utilizado o pacote `java.awt.image`, que contém classes para criar e modificar imagens, como a

classe `BufferedImage` que armazena os dados da imagem em um *buffer*. Outro pacote também utilizado para a manipulação da imagem é o pacote `javax.media.jai`, e as classes `JAI` e `PlanarImage`, que transformam os dados da imagem em pixels.

9.3. Descrição do Algoritmo

A imagem do anel obtida pelo CCD é enviada para processamento por um *software* que retorna como resultado o astigmatismo do olho sendo examinado. Para isso, várias etapas são necessárias como descrito a seguir.

9.3.1. Aquisição da imagem e transformação em escala de cinza

A primeira etapa para o processo de medição do astigmatismo é a aquisição da imagem da córnea do paciente. Neste trabalho, o processo é feito a partir de imagens obtidas por um ceratômetro de lâmpada de fenda, desenvolvido no próprio laboratório de instrumentação oftálmica, como já citado anteriormente.

Até o momento, a imagem obtida deve conter a extensão *Portable Network Graphics* (PNG), para ser compatível com o tipo de arquivo que a biblioteca utilizada no desenvolvimento do *software* em J2ME permite abrir. O pacote utilizado é o `javax.microedition.lcdui`, a classe é a `Image`, e o método é o `createImage()`.

Após abrir a imagem, a próxima etapa é transformá-la em uma matriz cujos elementos representam o valor de intensidade de cada pixel da imagem em escala de cinza, no intervalo $[0, 255]$, onde 0 representa a cor preta, e 255 a cor branca. A variável que armazena esta matriz foi nomeada como `grayscaleTable[][]` do tipo inteiro. O método que obtém as cores RGB da imagem é o `getRGB()`, e a conversão para escala de cinza é obtida por meio do cálculo mostrado na Figura 9.1.

```
// deixa a imagem Cinza: os dados ficam em GrayScaleTable

int iaux;
gray1 = new int [larg*alt];
grayscaleTable = new int [larg][alt];

for( int i = 0; i < rgb.length; ++i){
    iaux = (((rgb[i]>>16) &0xff)+((rgb[i]>>8) &0xff)+(rgb[i] &0xff))/3;
    gray1[i] =iaux;
    rgb[i] = (((rgb[i]>>24) &0xff) << 24) | (iaux << 16) | (iaux << 8) | iaux;
}

int i=-1;
for( int y = 0; y < alt;++y){
    for( int x = 0; x < larg;++x){
        i++;
        grayscaleTable[x][y] = gray1[i];
    }
}
```

Figura 9.1. Transformação da imagem em escala de cinza.

Nesta Figura, `larg` é a largura da imagem original em pixels, `alt` é a altura da imagem original em pixels, `rgb.length` armazena os valores RGB da imagem, `gray1` armazena o valor do cálculo de $(R + G + B)/3$, que caracteriza a cor cinza, e esse valor é armazenado na matriz `grayscaleTable[x][y]`.

9.3.2. Identificação dos LEDs

Com a imagem transformada em uma matriz, é possível iniciar o processamento para identificar a posição dos LEDs.

Para isso, o primeiro passo é encontrar a posição central da imagem que a princípio é determinada como metade da largura e metade da altura da imagem, definidos como $(x_{\text{medio}}, y_{\text{medio}})$.

O número de LEDs que se deseja identificar é pré-definido pelo usuário e armazenado na variável `ledsNumber` do tipo inteiro. A partir desse valor, a imagem é separada em setores de 360 graus do anel dividido por `ledsnumber` (Figura 9.2).

```
for (int n = 0; n < ledsNumber; n++) {
    maiorMedia = 0;
    medial = 0;
    raio = 0;
    anguloTheta = 0;
    //Varredura angular
    for (int theta = (dT * (n)) + 6; theta <= dT * (n + 1); theta++) {
        for (int r = rMin; r <= rMax; r++) {
            x1 = (int) (r * Math.cos((theta * Math.PI) / 180));
            y1 = (int) (r * Math.sin((theta * Math.PI) / 180));
            soma1 = 0;
            if ((x1 + xMedio - 3 > 0) && (y1 + yMedio - 3 > 0)) {
                if ((x1 + xMedio + 3 < larg) && (y1 + yMedio + 3 < alt)) {
                    //Máscara 7x7, centro = (xMedio,yMedio)
                    for (int k = -3; k <= 3; k++) {
                        for (int l = -3; l <= 3; l++) {
                            soma1 += grayscaleTable[xMedio + x1 + k][yMedio + y1 + l];
                        }
                    }
                    medial = soma1 / 49;
                    if (medial > maiorMedia) {
                        maiorMedia = medial;
                        xP1 = xMedio + x1;
                        yP1 = yMedio + y1;
                        anguloTheta = theta;
                        raio = r;
                    }
                }
            }
        }
    }
    //Armazena o raio e o ângulo (em radianos) do led identificado em ordem de identificação
    ledRaio[n + 1] = raio;
    ledTheta[n + 1] = anguloTheta * Math.PI / 180;
    //Posição dos Leds encontrados
    ledX[n] = xP1;
    ledY[n] = yP1;
}
```

Figura 9.2. Algoritmo da varredura angular para identificação dos LEDs.

Neste algoritmo, a varredura angular é iniciada a partir da variável `theta`, responsável pela delimitação dos setores (em ângulos). Isto é, para $n = 0$, a varredura inicia de `theta=0` até `theta = (360/ledsnumber)` multiplicado por $(n + 1)$.

A Figura 9.3 mostra um exemplo prático do processo de identificação dos LEDs.

Figura 9.3. Anel de LEDs dividido em setores 10 setores para identificar 10 LEDs.

Neste exemplo foi estabelecido que serão identificados 10 LEDs, então, os 360° foram divididos por `ledsnumber = 10`, e o resultado foi uma divisão do anel em 10 setores de 36° cada, isto é, para $n = 0$, a varredura inicia de $\theta = 0$ até $\theta = 36$; para $n = 1$, a varredura será de $\theta = 36$ até $\theta = 72$; e assim até que n seja igual a `ledsnumber` e $\theta = 360$.

Uma máscara é utilizada para percorrer cada um desses setores para identificar o pixel de maior intensidade nessa região. Essa máscara é uma matriz de tamanho 7×7 pixels.

Esta máscara permite a identificação dos pequenos círculos formados pela reflexão dos LEDs pela córnea, determinando sua localização. Com este procedimento a imagem dos LEDs pode ser identificada com relação às coordenadas do ponto central. Um exemplo pode ser observado na Figura 9.4.

Figura 9.4. Exemplo de uma imagem com a identificação dos LEDs refletidos.

As coordenadas dos LEDs identificados são dadas pela posição central da máscara quando ela indica a presença do LED. Assim cada LED identificado n é representado pelas coordenadas $\text{ledX}[n] = xP1$ e $\text{ledY}[n] = yP1$.

Neste ponto, o centro de coordenadas, representado como (x_0, y_0) , passa a ser definido pelo valor médio da posição de coordenada de cada LED, como na equação (9.1),

$$x_0 = \sum_{i=1}^N x_i, \text{ e } y_0 = \sum_{i=1}^N y_i \quad (9.1)$$

onde N é o número de LEDs identificado (Figura 9.5).

A seguir, considerando que a simetria circular é mais apropriada para este sistema, as coordenadas de cada LED, relativas ao centro de coordenadas, são convertidas para o sistema de coordenadas polares (r_i, θ_i) , segundo as equações (9.2) e (9.3).

$$r_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \quad (9.2)$$

$$\theta_i = \arctg\left(\frac{y_i - y_0}{x_i - x_0}\right) \quad (9.3)$$

Figura 9.5. Sistema de coordenadas polares.

Os LEDs devem então ser classificados em ordem crescente com relação ao ângulo, para que suas coordenadas possam ser utilizadas como entrada para um processo de interpolação polinomial, baseado no algoritmo de Neville [PRESS et al., 1996].

Os pontos serão gerados a cada grau, resultando em uma curva com um total de 360 pontos. Para cada novo ponto serão utilizadas as coordenadas de seis LEDs vizinhos como entrada para a interpolação, garantido precisão e estabilidade nos cálculos. Este processo interliga os LEDs, permitindo estimar com precisão a forma do anel de luz. Por isso, quanto maior o número de LEDs previamente identificados, maior a precisão da estimativa.

Após a interpolação, os raios maior e menor da elipse é calculada. A Figura 9.6 e Figura 9.7 mostram o algoritmo do cálculo desses raios.


```

AngMax=0;
Max2=0;
for( int j = 1; j <= 180; j++ ){
    Mx1= ((Xmed-ledXInterp[j])*(Xmed-ledXInterp[j]) +
          (Ymed-ledYInterp[j])*(Ymed-ledYInterp[j]));
    Mx2= ((Xmed-ledXInterp[j+180])*(Xmed-ledXInterp[j+180]) +
          (Ymed-ledYInterp[j+180])*(Ymed-ledYInterp[j+180]));
    Max1= Math.sqrt(Mx1)+Math.sqrt(Mx2);
    if(Max1 > Max2){
        Max2=Max1; AngMax=j;
    }
}
Rmaior = Max2/2.0;

```

Figura 9.6. Algoritmo do cálculo do maior raio da elipse.

A partir do centro de massa da elipse (X_{med} , Y_{med}), calcula-se a distância euclidiana $Mx1$ de cada coordenada de LED até o centro, somada com a distância euclidiana $Mx2$ do LED que está a 180° dele, e a soma que obtiver o maior valor será o maior raio da elipse.

```

AngMin = AngMax+90;
if( AngMax > 90){
    AngMin=AngMax-90;
}
Mx1= ((Xmed-ledXInterp[AngMin])*(Xmed-ledXInterp[AngMin]) +
      (Ymed-ledYInterp[AngMin])*(Ymed-ledYInterp[AngMin]));
Mx2= ((Xmed-ledXInterp[AngMin+180])*(Xmed-ledXInterp[AngMin+180]) +
      (Ymed-ledYInterp[AngMin+180])*(Ymed-ledYInterp[AngMin+180]));
Max1= Math.sqrt(Mx1)+Math.sqrt(Mx2);
Rmenor = Max1/2;

```

Figura 9.7. Algoritmo do cálculo do menor raio da elipse.

O mesmo cálculo é feito para identificar o menor raio da elipse, com o diferencial de definir como menor raio a soma que obteve o menor valor.

A imagem resultante de todo o processo é mostrada na Figura 9.8.

Figura 9.8. Anel contínuo formado pelo processo de interpolação utilizando os pontos dos LEDs identificados pelo software. O maior e o menor raio são mostrados.

O cálculo do ângulo de inclinação é medido entre o raio maior com a horizontal, não ultrapassando o valor de -90° a $+90^\circ$.

Capítulo 10. RESULTADOS E DISCUSSÃO

A viabilidade do algoritmo foi verificada por meio de testes preliminares realizados utilizando linguagem JavaSE com a IDE Eclipse. Os resultados apresentados a seguir foram baseados em imagens do anel de LEDs do protótipo anterior de Lâmpada de Fenda. Esta escolha para teste de viabilidade foi feita considerando a similaridade do anel de LEDs deste sistema com o necessário para este novo projeto, tornando possível estimar a eficiência do novo algoritmo no reconhecimento dos LEDs e na interpolação da curva do anel. Este novo algoritmo, além de diferir na linguagem (o módulo de Lâmpadas de Fenda foi desenvolvido com a plataforma Delphi), precisou ser adaptado para compensar uma grande diferença no tempo de execução. Embora a linguagem Java seja a indicada para o projeto devido à grande versatilidade na escolha do sistema operacional, especialmente no caso de dispositivos móveis, chega a ser 15 vezes mais lenta em um mesmo computador que o executável desenvolvido com Delphi ou C. Entretanto os testes apresentaram bons resultados. Utilizando um Intel® Core™2 Duo, 1,73 GHz, RAM de 2038 Mb, o tempo total de processamento das etapas descritas na seção anterior foi de 300 milissegundos.

Os testes e a metodologia apresentados precisarão ser testados no desenvolvimento do ceratômetro móvel proposto. Contudo, o desempenho observado no algoritmo descrito, a eficiência na identificação dos LEDs, e o cálculo dos raios maior e menor do anel de luz necessários para o oftalmologista determinar o astigmatismo indicam fortemente a viabilidade do processo proposto neste projeto.

Após fazer vários testes com uma imagem padrão em que o anel de LEDs refletido é um círculo regular, utilizou-se a Figura 10.1 obtida por um paciente com grau considerável de astigmatismo para testar se o *software* é capaz de fazer a medição.

Figura 10.1. Imagem obtida de um olho com astigmatismo.

O resultado deste teste se mostrou bem sucedido, identificando corretamente os LEDs, determinando a melhor elipse que se encaixa a esses LEDs identificados por meio da interpolação, e traçando o maior e o menor raio da elipse, como mostra a Figura 10.2.

Figura 10.2. Imagem após o processamento do *software*.

A migração desse *software* para um novo utilizando a tecnologia JavaME mostrou que, as dificuldades encontradas pela falta de materiais disponíveis para auxiliar o desenvolvimento utilizando esta tecnologia foram superados pelo desenvolvimento de

módulos próprios que se mostraram eficientes. Desta forma, o *software* é capaz de fazer todo o processamento em um tempo aceitável.

Alguns teste foram realizados no smartphone Nokia N95, com o processador ARM de 333MHz, apresentando uma média de 5,8s. E a tendência é que os processadores se tornem cada vez mais rápidos.

10.1. Imagens do Software em JavaME no simulador

A tela inicial (Figura 10.3) mostra a imagem do anel luminoso projetado na córnea do paciente obtida no ceratômetro de lâmpada de fenda.

Figura 10.3. Tela inicial no simulador da Sun.

No aplicativo há duas funcionalidades estão em Menu, a identificação dos LEDs, que mostra os LEDs identificados circulado em azul e a interpolação e traço do maior e menor raio da elipse, mostradas em verde para visualização, além dos valores dos raios e o ângulo de inclinação.

Na Figura 10.4 são apresentadas as imagens utilizadas para testar os resultados obtidos pelo *software* desenvolvido em comparação aos resultados obtidos no *software* anterior desenvolvido em Delphi.

Figura 10.4. Imagens para teste e comparação dos resultados.

A Tabela 10-1 mostra os resultados de testes realizados com imagens de alguns casos específicos em que os pontos luminosos podem aparecer borrados, pouco nítidos, sobrepostos, unidos, com ruídos internos e/ou externos, ou outros fatores diversos.

Tabela 10-1. Tabela de comparação e porcentagem de erro em relação ao software anterior desenvolvido em Delphi.

Imagem no Simulador		Software JavaME	Software anterior	Erro (%)
Imagem A		Raio Maior: 93.77767486437094	Raio Maior: 93,616	0,17197%
		Raio Menor: 90.18454395036427	Raio Menor: 90,397	0,23562%
		Ângulo: -61	Ângulo: -59,982	1,69717%
Imagem B		Raio Maior: 130.0473114784673	Raio Maior: 129,05	0,77256%
		Raio Menor: 111.80007057448472	Raio Menor: 113,12	11,669%
		Ângulo: -9	Ângulo: -9,5038	5,26315%
Imagem C		Raio Maior: 101.17949916806967	Raio Maior: 101,33	0,15789%
		Raio Menor: 90.67177484738963	Raio Menor: 91,263	0,64867%
		Ângulo: 24	Ângulo: 26,479	9,36213%

Imagem D		Raio Maior: 101.3422478185212 Raio Menor: 86.63509529960365 Ângulo: -25	Raio Maior: 102,19 Raio Menor: 86,427 Ângulo: -28,001	0,83178% 0,24066% 10,71747%
Imagem E		Raio Maior: 92.44729511141503 Raio Menor: 82.39691088792559 Ângulo: -24	Raio Maior: 92,24 Raio Menor: 84,063 Ângulo: -28,551	0,21682% 1,98303% 15,93989%

Os resultados demonstram que o *software* desenvolvido para ser executado em processadores mais simples está retornando valores com uma margem de erro praticamente nula, evidenciando o sucesso de seu desenvolvimento em relação aos valores obtidos dos raios da elipse e o ângulo de inclinação, inclusive para imagens defeituosas que desafiam desde a identificação de alguns LEDs até a determinação da elipse.

Esses resultados têm mostrado a alta viabilidade de se desenvolver um modelo portátil.

Capítulo 11. CONCLUSÃO E TRABALHOS FUTUROS

O *software* desenvolvido e testado neste trabalho é uma alternativa viável para ser utilizado no ceratômetro de lâmpada de fenda, equipamento destinado ao cálculo automático dos raios de curvatura da córnea e conseqüente medição do astigmatismo corneano.

Com este novo *software* é possível substituir os computadores *desktop* utilizados atualmente, por computadores portáteis, como um PDA ou miniPC. As vantagens obtidas são de um equipamento mais leve, moderno, que reduz o espaço ocupado, geralmente oferece preços mais baixos, e processadores rápidos o suficiente para o propósito deste trabalho, como demonstraram os resultados obtidos.

A vantagem da construção de um primeiro protótipo utilizando a tecnologia JavaSE em um computador *desktop* foi principalmente pelo suporte a algoritmos mais complexos, e mais fáceis de serem desenvolvidos, reduzindo o tempo gasto na avaliação da viabilidade de se desenvolver em JavaME para processadores mais simples.

Durante o desenvolvimento do programa, o maior desafio encontrado foi na adaptação do *software* desenvolvido em JavaSE para JavaME, ou seja, no momento de otimizar o algoritmo e substituir as bibliotecas por outras mais leves direcionadas a processadores mais simples. A escassez tanto de materiais na área de processamento digital de imagens, como classes desenvolvidas para reutilização podem ser apontados como os principais motivos desse desafio.

O erro médio comparado com o programa desenvolvido n projeto anterior foi de aproximadamente 0,0003993854%, demonstrando que a diferença entre os *softwares* é praticamente nula.

Este trabalho propôs uma forma mais moderna e dinâmica para a medição automática do astigmatismo, e pôde concluir que o desenvolvimento de um ceratômetro móvel é possível.

Como proposta para atividades futuras, sugere-se a continuidade do desenvolvimento do ceratômetro portátil. E, em relação ao *software*, propõe-se o desenvolvimento da conexão entre o *software* e a câmera CCD para que a imagem seja obtida pelo *software* em tempo real.

Outra proposta é estudar a viabilidade de se agregar ao ceratômetro portátil as funcionalidades de um topógrafo, tornando-o ainda mais útil e econômico para o oftalmologista.

REFERÊNCIAS

- BLOCH, J. (2008). **Effective Java**. 2nd ed. Upper Saddle River, New Jersey, Addison-Wesley, xxi, 346 p. Revised and updated for Java SE 6.
- BRIGHTBILL, F. S.; MCDONNELL, P. J.; MCGHEE, C. N. J.; FARJO, A. A.; SERDAREVIC, O. (2009). **Corneal surgery: theory, technique and tissue**. 4.ed. Elsevier Health Sciences, p.945.
- CARVALHO, L. A. V. de. (2005). **Técnicas para facilitar o diagnóstico da topografia da córnea *in vivo***. Arquivos Brasileiros de Oftalmologia. Vol.68 no.2: 205-12. São Paulo, Mar./Apr. 2005.
- COELHO, R. (1999). **Desenvolvimento de um Sistema Automatizado em Lâmpada de Fenda para Medidas Ceratométricas**. 108f. Dissertação (Mestrado). Faculdade de Filosofia Ciências e Letras de Ribeirão Preto, Departamento de Física e Matemática, Universidade de São Paulo, Ribeirão Preto, 1999.
- COSTA, E. B. G. da; LIMA JR., O. F. L.; FECHINE, J. M.; QUEIROZ, J. E. R. (2009). **Processamento Digital de Imagens no Domínio da Frequência**. Departamento de Sistemas e Computação. Campina Grande, Universidade Federal de Campina Grande.
- DEITEL, H. M. (2007). **Java: Como Programar**. São Paulo, Pearson Prentice Hall.
- DOMÉ, E. F. (2001). **Estudo do olho humano aplicado à optometria**. 3. ed. São Paulo: SENAC, São Paulo. p.223.
- DUCHOWSKI, A. T. (2007). **Eye Tracking Methodology: Theory and Practice**. 2.ed. Londres: Springer.
- FERRON, M.; RANCANO, J. (2007). **Grande Atlas do Corpo Humano: Anatomia, Histologia, Patologias**. Barueri, SP: Editora Manole Ltda, p.560.
- GONÇALVES, M. H. B.; KRITZ, S. (1997). **Lentes: ofício e benefícios**. Rio de Janeiro: Editora Senac Nacional. 3.ed. p.128.
- GONZALEZ, R. C.; WOODS, R. E. (2003). **Processamento digital de imagens**. Editora Edgard Blücher LTDA.
- MARTINS, H. B.; BOTELHO, F. C. (2008). **Java não é mais tão lento**. Revista Eletrônica de Iniciação Científica, SBC, número 4.
- MILÉO FILHO, P. M. (1996). **Introdução à óptica geométrica**. Série Apontamentos: Saúde. São Paulo: SENAC, p.147.
- MONTEIRO, J. D. A. (2006). **Desenvolvimento de aplicações multi-plataformas para dispositivos móveis**. Instituto de Ciências Matemáticas e de Computação (ICMC). São Carlos, SP, USP.

- OLIVEIRA, J.; WACHTER, P. H.; AZAMBUJA, A. A. (2002). **Biofísica: para ciências biomédicas**. Porto Alegre: EDIPUCRS, p.313.
- PRESS, W. H.; FLANNERY, B. P.; TEUKOLSKY, S. A.; VETTERNLING, W. T. (1996). **Numerical recipes in Pascal: The Art of Scientific Computing**. Sydney, Press Syndicate of the University of Cambridge.
- RUSS, J. C. (2007). **The Image Processing Handbook**. 5. ed. North Carolina State University. Materials Science and Engineering Department. Raleigh, North Carolina.
- SCHOR, P. (1997). **Idealização, Desenho, Construção e Teste de um Ceratômetro Cirúrgico Quantitativo**. Tese apresentada à Universidade Federal de São Paulo – Escola Paulista de Medicina para obtenção do Título de Doutor em Medicina, São Paulo.
- SILVA, J. P. d. N. (2007). **Usabilidade para sites acedíveis por dispositivos com ecrã de pequena dimensão**. Departamento de Ciências e Tecnologia da Informação, Instituto Superior de Ciências do Trabalho e da Empresa. Dissertação (Mestrado em Gestão de Sistemas de Informação), p.89.
- VENTURA, L. (2007). **Módulo ceratométrico para microscópios oculares**. Tese de livre docência. Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2007.
- VENTURA, L.; CASTRO NETO, J. C. (1995). **Ametropias Oculares**. Revista Brasileira de Ensino de Física, vol. 17, no. 4, dezembro, 1995.
- VENTURA, L.; MESSIAS, A.M.V.; SOUSA, S. J.F. (2000). **Automated system for keratometry in slit lamp**. Ophthalmic Technologies X – (SPIE - The International Society for Optical Engineering), V. 3908, p. 243 - 253, 2000.
- VENTURA, L.; MESSIAS, A.M.V.; SOUSA, S. J.F, COELHO, R. (2000a). **Automated Keratometry at Low Cost**. IEEE - Engineering in Medicine & Biology Magazine, V19, n.5, p. 97-103, Set/Out 2000.
- VENTURA, L.; RIU, C.; DE GROOTE, J.J; ROSA FILHO, A.B.; OLIVEIRA, G.C.D. (2006). **Corneal astigmatism measuring module for slip lamps**. Physics in Medicine and Biology. v.51, n12, p.1–14, jun.
- VENTURA, L.; RIUL, C.; SOUSA, S.J.F. (2007). Optical projection mires for measurements of the radii of curvature of the cornea in biomicroscopes – Applied Optics, Vol. 46, No. 30 October 2007.
- WANG, M. (2006). **Corneal Topography in the Wavefront Era: A Guide for Clinical Application**. Slack Inc. Wang Vison Institute. Nashville, Tennessee.
- WANT, R. (2007). **Carry Small, Live Large**. IEEE Pervasive Computing: 2-4.