

Filipe Hilário Muler
Frederico Uemura Pimenta
Rafael Galucci Pereira Passos

Serviço de Bibliotecas
Biblioteca de Engenharia Elétrica

Sistema de rastreamento do olhar baseado em câmera

São Paulo

2008

**Filipe Hilário Muler
Frederico Uemura Pimenta
Rafael Galucci Pereira Passos**

Sistema de rastreamento do olhar baseado em câmera

Projeto de Formatura

Área de Concentração:
Engenharia Elétrica - Computação
Semestral

Orientador:
Prof. Dr. Romero Tori

Co-orientador:
Mestrando Lucas Padovani Trias

São Paulo

2008

Resumo

As formas de interfaceamento humano-máquina mais comuns consistem em dispositivos de entrada tais como *mouse*, teclado, *touchpad* e tela sensível ao toque. Além destas, a utilização de outras formas de interação como voz e movimento aumentam o grau de interatividade com os sistemas computacionais, facilitam a execução de tarefas e melhoram a experiência imersiva em ambientes virtuais. Um método de interação ainda pouco difundido é a interação pelo olhar. Neste trabalho aborda-se a criação de um sistema que consiga estimar em que ponto de um monitor o usuário esteja olhando num dado momento. Pretende-se com ele, criar e analisar uma forma alternativa de interação humano-máquina que traga um maior grau de conforto e agilidade às tarefas do dia a dia, além de abrir a possibilidade de uma maior acessibilidade a portadores de necessidades especiais que não consigam operar dispositivos convencionais de entrada/saída de forma adequada.

Palavras-chave: rastreamento do olhar. interação humano-máquina

Abstract

The most common devices used for human-machine interaction are mouse, keyboard, touchpad and touchscreen. The use of alternative methods like the use of voice and movement recognition increase the interactivity of computer systems, improve task execution and immersion in virtual environments. A not commonly used method for interactivity is eye gaze tracking. In this document it is desired to develop a system that can estimate the point where the user's gaze rests at a given time.

Keywords: eye tracking. human-machine interaction.

Lista de Figuras

1	Imagem Digital, adaptado de [1]	p. 17
2	Histograma	p. 19
3	Exemplo de limiarização	p. 20
4	Operação de Subtração entre imagens. Extraído de [2]	p. 21
5	Aplicação da Desfocagem	p. 22
6	Distorção em barril	p. 22
7	Distorção em palheiro	p. 23
8	Imagens ilustrando correção da distorção da imagem	p. 23
9	Modelos do olho humano	p. 26
10	Pupila iluminada por uma fonte de luz infravermelha	p. 27
11	<i>Glint: reflexos da luz na córnea</i>	p. 28
12	Vetor <i>glint</i> -pupila	p. 28
13	Gerenciamento do projeto	p. 30
14	Ciclo de Desenvolvimento	p. 31
15	Atividades do desenvolvimento do <i>Harware</i>	p. 32
16	Sequência de ações utilizadas para determinar a configuração do sistema	p. 35
17	Circuito de sincronismo	p. 37
18	Imagens do equipamento	p. 38
19	Passo adicional na sequência de obtenção de imagens	p. 39
20	Componentes do sistema	p. 40
21	Diagrama de classes do sistema - A	p. 42
22	Diagrama de classes do sistema - B	p. 43

23	Diagrama de sequência do software de rastreamento	p. 44
24	Diagrama de estados da detecção da pupila	p. 45
25	Passos da detecção da pupila: pupila ainda não detectada	p. 46
26	Passos da detecção da pupila: Pupila já detectada anteriormente	p. 47
27	Etapas da detecção do glint	p. 48
28	Imagens da tela de calibração	p. 49

Lista de Tabelas

- 1 Resultado de detecção da pupila p.51
- 2 Resultado de detecção do *glint* p.51

Lista de abreviaturas e siglas

- LED Light Emissor Diode
- RGB Red, Green, Blue
- UART Universal Asynchronous Receiver/Transmitter
- USB Universal Serial Bus
- CMOS Complementary metaloxidesemiconductor
- OpenCV Open Source Computer Vision Library

Sumário

1	Introdução	p. 12
1.1	Objetivo	p. 13
1.2	Organização do Texto	p. 14
2	Revisão Bibliográfica	p. 15
3	Tratamento de Imagens Digitais	p. 17
3.1	Imagem Digital	p. 17
3.1.1	Imagens em escala de cinza e imagem binária	p. 18
3.1.2	Histograma de uma imagem em escala de cinza	p. 19
3.2	Operações sobre imagens	p. 20
3.2.1	Limiarização	p. 20
3.2.2	Subtração entre imagens	p. 21
3.2.3	Desfocagem	p. 21
3.3	Correção de distorções da lente	p. 22
3.4	Considerações Finais	p. 24
4	Rastreamento do Olhar	p. 25
4.1	Modelo do Olho	p. 25
4.2	Rastreamento	p. 26
4.3	Obtenção do centro da pupila	p. 27
4.4	Ponto Fixo de Referência: <i>Glint</i>	p. 27
4.5	Considerações Finais	p. 28

5	Metodologia	p. 29
5.1	Pesquisa	p. 29
5.2	Gerenciamento	p. 29
5.3	Desenvolvimento do <i>Software</i>	p. 30
5.4	Desenvolvimento do <i>Hardware</i>	p. 32
5.5	Programação dos Testes	p. 32
5.6	Considerações Finais	p. 33
6	Projeto e Implementação	p. 34
6.1	Pesquisa e Ambiente de Desenvolvimento	p. 34
6.2	Provas de Conceito de Detecção	p. 35
6.3	Processo de teste	p. 35
6.4	<i>Hardware</i>	p. 36
6.5	Calibração da Câmera	p. 39
6.6	<i>Software</i> e Integração	p. 39
6.6.1	Diagrama de Classes do Sistema	p. 41
6.6.2	Diagrama de Sequência do Sistema	p. 41
6.6.3	Estados da Detecção da Pupila	p. 45
6.6.4	Algoritmo de Detecção da Pupila	p. 45
6.6.5	Algoritmo de Detecção do <i>Glint</i>	p. 47
6.7	Calibração do Sistema	p. 48
6.8	Considerações Finais	p. 49
7	Conclusão	p. 50
7.1	Resultados e Viabilidade do Sistema	p. 50
7.2	Limitações	p. 52
7.3	Trabalhos Futuros	p. 53

7.4	Considerações Finais	p. 54
	Referências	p. 55
	Apêndice A – Suporte para os LED's	p. 57

1 Introdução

As formas de interfaceamento humano-máquina mais comuns consistem em dispositivos de entrada tais como *mouse*, teclado, *touchpad* e tela sensível ao toque. Além destas, a utilização de outras formas de interação como voz[3], pés [4], movimento [5, 6] aumenta o grau de interatividade com os sistemas computacionais, facilita a execução de tarefas e melhora a experiência imersiva em ambientes virtuais. Um método de interação ainda pouco difundido é a interação pelo olhar, o qual tem grande potencial para satisfazer as qualidades mencionadas anteriormente.

Em um computador, sempre que se digita um texto, seleciona-se um ícone ou se interage em um ambiente virtual como um jogo, deposita-se a atenção sobre o objeto virtual com o qual se deseja interagir através da visualização de tal objeto. Em [7] é apresentado um estudo sobre o uso de dispositivos de rastreamento do olhar em jogos eletrônicos que demonstra um ganho na imersão e naturalidade das ações segundo a opinião dos participantes dos testes.

Na medicina, o rastreamento do olhar pode auxiliar o estudo do comportamento dos olhos e dos elementos que chamam a atenção numa cena. Na área de *marketing* e propaganda o uso de tal conhecimento auxilia na determinação de melhores políticas para formulação de propagandas e *layouts* de páginas *web*, revistas e sessões de jornais. Novas formas de interação com jogos eletrônicos também se beneficiariam de tal recurso, talvez não como substituto direto de dispositivos atuais - uma vez que nem sempre isso agrada aos usuários [7] - mas como auxiliares. Outra aplicação seria a disponibilização de uma forma de acesso potencialmente mais confortável a portadores de necessidades especiais.

1.1 Objetivo

Este trabalho tem como objetivo realizar o projeto e implementação de um sistema de interação humano-máquina que servirá como alternativa ao movimento realizado pelo *mouse* em um computador. Para isso, o sistema realizará o rastreamento do foco de atenção do usuário na tela, sendo capaz de detectar em que local da mesma em que o usuário olha. O sistema não será responsável pela atuação sobre os elementos dispostos na tela, sendo essa função delegada à teclas existentes no teclado do sistema computacional utilizado. Portanto, o foco do sistema está em usuários comuns que consigam operar normalmente um teclado e não se aplica à portadores de necessidades especiais.

Deseja-se que o sistema não exija que o usuário utilize equipamentos adicionais acoplados à cabeça ou qualquer parte de seu corpo, tornando a interação mais natural. Além disso, o equipamento utilizado deve possuir um custo baixo de componentes, de modo que o custo do equipamento não seja significativamente alto frente ao custo do próprio computador. Para isso, será explorada a utilização de uma *webcam*, um dispositivo que está disponível em muitos computadores pessoais de mesa e em *laptops*.

1.2 Organização do Texto

O documento está organizado de modo que no início tenhamos uma pequena apresentação dos principais trabalhos relacionados na área e pesquisados para o projeto. No capítulo 3 são apresentados os principais tópicos relacionados a tratamento de imagens digitais do qual este trabalho faz uso, desde conceito de imagem digital até operações complexas, como a remoção da distorção causada pela câmera na imagem.

Após essa parte conceitual, segue-se no Capítulo 4 com a apresentação do modelo de rastreamento do olhar utilizado, onde se faz uso das operações mencionadas anteriormente. No Capítulo 5 é discutida a metodologia empregada pela equipe para o desenvolvimento do projeto. Essa sessão abrange desde como se realizou as pesquisas na área, passando pelos procedimentos de testes até chegar nas provas de conceito.

O capítulo de **Projeto e Implementação** descreve o desenvolvimento do *hardware*, *software* e a integração de ambos. É comentado também como foi o processo de calibração da câmera e do sistema.

Os resultados e a análise de viabilidade do sistema são apresentadas logo em seguida, finalizando com as limitações e trabalhos futuros que podem melhorar o sistema.

2 Revisão Bibliográfica

Nos inícios dos trabalhos a pesquisa não restringiu em nenhuma maneira as fontes investigadas. Graças a esta postura pode-se conhecer diversas formas de aplicações para os resultados do rastreamento ocular. Existem, por exemplo, empresas como a Enquiro [8] que se beneficiam de tais sistemas para prover informações úteis de design para o site de seus clientes. Já existem também soluções comerciais de sistemas de rastreamento como o Tobii [9] que já permitem o acesso a tecnologia de rastreamento à institutos, pessoas com necessidades especiais e demais indivíduos interessados.

Mas apesar de investidas comerciais como estas, a maioria de pesquisas na área tem como finalidade os mais diversos interesses acadêmicos. Tendo seus primórdios enraizados pela simples curiosidade do saber, tais como os trabalhos do oftalmologista francês Louis Émile Javal [10], logo inúmeros sistemas de rastreamento foram desenvolvidos para estudos patológicos e fisiológicos, como os descritos nas pesquisas [11] e [12] publicadas por R. S. Allisson. Mas trabalhos como o de Byrne e Anderson [13] na utilização de tais sistemas para uso de interfaceamento em sistemas computacionais que realmente motivou a equipe em prosseguir nas pesquisas para o desenvolvimento deste trabalho.

Assim que o objetivo de desenvolver um sistema desta natureza foi consolidado, o foco em pesquisa de sistemas com diferentes configurações foi traçado. Muitas alternativas foram encontradas e analisadas, como comprova a bibliografia referenciada. Entre elas existem alguns métodos utilizados para realizar o rastreamento do olhar que podem ser classificados a partir de uma divisão básica: métodos que exigem que o usuário utilize algum dispositivo auxiliar e métodos que não o exigem.

- Como exemplos do primeiro caso têm-se a utilização de óculos contendo *LED's* infravermelhos [12], utilização de lentes de contato como as citadas em [11] e outros apetrechos.
- A segunda maneira, sem exigência de contato do usuário com dispositivos extras fazem uso de câmeras que tentam capturar algumas características da cabeça, face e

olho, sejam elas naturais como as apresentadas em [14] ou artificialmente induzidas pela utilização, por exemplo, de *LED's* auxiliares, como apresentado em [15].

Características naturais geralmente escolhidas são o posicionamento da cabeça e a inclinação do eixo que conecta os olhos [14] e a localização da pupila. Dentre estas, é de grande difusão a aplicação de técnicas que fazem uso de *LED's* infravermelhos para ajudar na detecção da pupila e da formação de pontos de referência dentro do olho [16], fazendo uso dos conceitos apresentados nos itens 4.3 e 4.4. Como o rastreamento por métodos sem contato do usuário com o dispositivo é feito majoritariamente por meio da captura de imagens, tais métodos ficam sujeitos às condições de iluminação do ambiente em que estão inseridos. A utilização das fontes de luz infravermelha tem portanto duas grandes justificativas:

- por emitir luz em faixas de frequência não visíveis ao olho humano, não causando incômodo ao usuário durante sua operação.
- as características proporcionadas pela iluminação das fontes de luz facilitam a detecção de elementos desejados na imagem capturada.

Dentre os trabalhos pesquisados dois chamaram a atenção pelos seus métodos empregados, ambos por serem bastante diferentes na abordagem e grau de complexidade.

Um deles trata-se da publicação de E. D. Guestrin de 2006 [17], onde é apresentada a dedução de um sistema de rastreamentos com bastante liberdade para o usuário utilizando-se apenas de uma câmera simples e alguns pontos luminosos de referência. Além de este trabalho estar bastante detalhado ele se enquadra com os desejos iniciais da equipe.

O outro trabalho trata-se de uma publicação de C. H. Morimoto, com várias publicações e patentes na área de rastreamento ocular e trabalhos relacionados. Neste *paper* em especial é apresentado um método relativamente simples para a detecção do olhar do usuário quando este se apresenta numa localização fixa[18].

3 Tratamento de Imagens Digitais

Essa seção apresenta algumas conceitos e definições de técnicas de tratamento de imagem utilizadas no projeto. Os conceitos apresentados neste capítulo foram adaptados de [1] e [19].

3.1 Imagem Digital

Uma imagem digital é basicamente uma matriz (Figura 1). A cada uma das coordenadas dessa matriz dá-se o nome de *pixel* e o conjunto dos *pixels* define o domínio espacial da imagem. Os valores mapeados podem representar grandezas como valor de cor (RGB)¹, luminância e crominância (YCbCr)² [1].

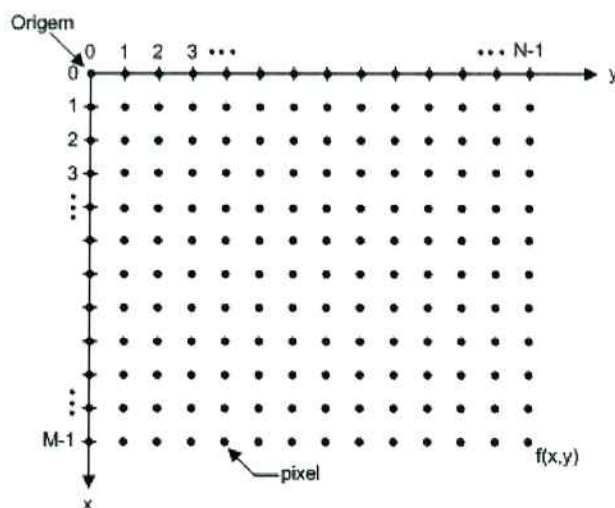


Figura 1: Imagem Digital, adaptado de [1]

¹Sistema de cores aditivo formado por Vermelho (Red), Verde (Green) e Azul (Blue).

²Luminância e crominância são definidas como:

$$\text{Luminância: } Y = 0.299R + 0.587G + 0.114B$$

$$\text{Crominância Vermelha: } C_r = R - Y$$

$$\text{Crominância Azul: } C_b = B - Y$$

Onde **R,G,B** são os valores de *pixel* referentes ao sistema **RGB**

O tamanho da imagem corresponde ao número de valores discretos presentes em cada linha e coluna da mesma. Desta forma, as coordenadas x e y de uma imagem são tais que:

$$\begin{cases} 0 \leq x < M \\ 0 \leq y < N, \quad M, N \in \mathbb{Z} \end{cases}$$

Assim, uma imagem pode ser vista como uma matriz $M \times N$ onde cada elemento a_{ij} com $i = 0, \dots, M - 1$ e $j = 0, \dots, N - 1$ desta matriz representa um valor de uma função f :

$$f(x, y) = \begin{pmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{pmatrix} \quad (3.1)$$

3.1.1 Imagens em escala de cinza e imagem binária

Em imagens em escala de cinza, cada valor a_{ij} da função f representa o tom de cinza de cada *pixel*. A representação deste valor é realizada através da utilização de k bits, $k \in \mathbb{Z}$ e portanto, o número de tons de cinza representáveis é:

$$L = 2^k \quad (3.2)$$

Em que os níveis discretos assumem valores no intervalo $[0, L - 1]$, sendo que quando $a_{i,j} = 0$ o *pixel* representa a cor preta e quando $a_{i,j} = L - 1$, representa a cor branca. Outro caso da Equação (3.2) ocorre para imagens binárias. Neste caso utiliza-se $k = 1$ bit para representação dos níveis de intensidade, obtendo-se $L = 2$ e tem-se somente a representação dos dois extremos: preto e branco. Uma das maneiras de se obter uma imagem binária é através de um processo de limiarização aplicado sobre uma imagem originalmente representada em escala de cinza.

3.1.2 Histograma de uma imagem em escala de cinza

O histograma de uma imagem em escala de cinza com níveis de representação dos tons variando no intervalo $[0, L - 1]$ corresponde a uma função discreta $h : Z \rightarrow Z$, tal que

$$h(r_k) = nk \quad (3.3)$$

onde r_k corresponde ao k -ésimo valor de *pixel* ($r_k \in [0, L - 1]$) presente na imagem e nk o número de vezes que este valor aparece na imagem. O histograma de uma imagem é útil para verificar a composição da imagem em termos dos tons de cinza (no caso da imagem em escala de cinza). Na Figura 2 é apresentado um exemplo de histograma gerado a partir de uma imagem. Pela Figura 2b, nota-se uma predominância de valores de *pixel* elevados.

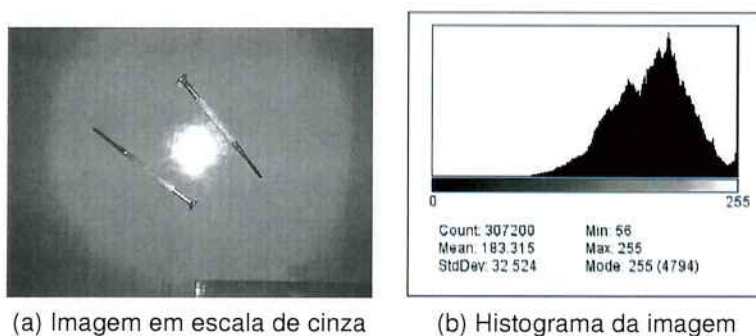


Figura 2: Histograma

Uma utilidade de se analisar o histograma de uma imagem é na escolha de parâmetros para segmentação da imagem, como é o caso da escolha do valor de limiar t para a operação da limiarização, operação descrita na Seção 3.2.1.

3.2 Operações sobre imagens

A seguir serão descritas algumas operações sobre imagens. O escopo da utilização destas operações será limitado à sua aplicação sobre imagens cujos valores são representados em escala de cinza ou sobre imagens binárias.

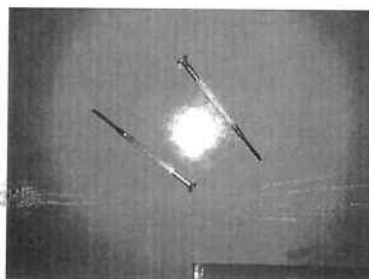
3.2.1 Limiarização

A limiarização (*thresholding*) sobre imagens em escala de cinza é uma operação na qual se escolhe um valor $T \in Z, 0 \leq T < L - 1$ e aplica-se uma função $t : Z \rightarrow Z$ sobre cada *pixel* da imagem tal que:

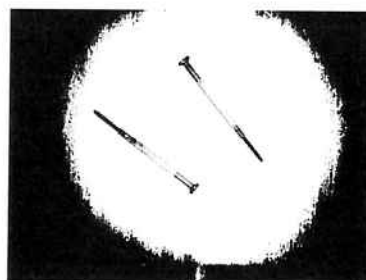
$$t(x, y) \begin{cases} \text{valor de objeto} & \text{se } f(x, y) > t \\ \text{valor de fundo} & \text{caso contrario} \end{cases}$$

onde $f(x, y)$ é o valor do *pixel*, i.e. um valor no intervalo $[0, L - 1]$ representando o tom de cinza valor de objeto e valor de fundo correspondem a um dos valores binários.

Esta operação é especialmente útil para realizar segmentação de imagens quando seus histogramas puderem ser particionados facilmente devido a diferença entre grupos de valores de interesse, escolhendo-se um valor t que separe dois grupos representados por picos no histograma. As Figuras 3a e 3b mostram respectivamente uma imagem original em escala de cinza e sua resultante após aplicação da operação de limiarização para $t = 180$.



(a) Imagem em escala de cinza



(b) Resultado da limiarização para $t = 180$

Figura 3: Exemplo de limiarização

3.2.2 Subtração entre imagens

Dadas duas imagens $f(x, y)$ e $h(x, y)$, a diferença $g(x, y)$ delas é definida como:

$$g(x, y) = f(x, y) - h(x, y) \quad (3.4)$$

onde o valor de cada par (x, y) é obtido pela subtração entre os pares correspondentes de f e g .

A utilização da subtração permite a segmentação de imagens onde as duas imagens originais diferem em apenas alguns componentes. A Figura 4 exemplifica a operação:

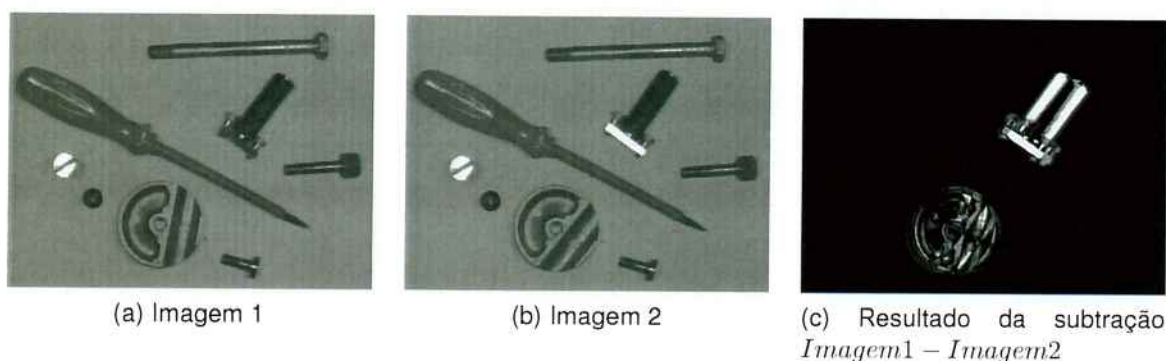


Figura 4: Operação de Subtração entre imagens. Extraído de [2]

3.2.3 Desfocagem

Esta operação é responsável pela uniformização de bordas e outros elementos de uma imagem que possuam variações bruscas de valor de *pixel*. Pode ser utilizada como ferramenta para eliminação de pequenos ruídos na imagem. O efeito visualmente observável é a diminuição da nitidez da imagem. A desfocagem pode ser utilizada para eliminar pequenos ruídos na imagem. A Figura 5 mostra como a eliminação de ruídos pode ser realizada através desta operação. Na Figura 5b, resultante da aplicação da limiarização sobre a Figura 5a, existem pequenos pontos brancos que foram introduzidos devido às condições de iluminação da cena. Ao aplicar o desfocamento, a imagem é suavizada, fazendo com que o ruído desapareça.

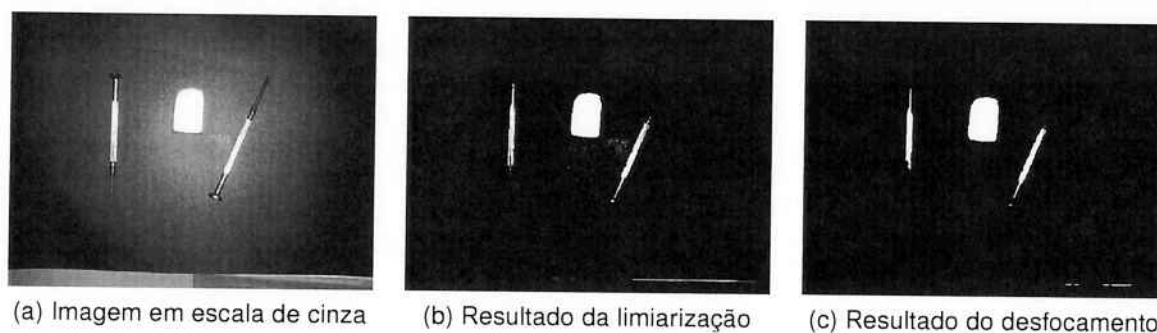


Figura 5: Aplicação da Desfocagem

3.3 Correção de distorções da lente

Na ótica geométrica clássica, e nos modelos desenvolvidos sobre ela, como nos casos usados em [17] e em [18], todas as projeções de imagens, inclusive no caso de câmeras de orifício (*pinhole cameras*)³, são supostas retilíneas, ou seja, todas as linhas retas na cena devem se manter retas e em proporção na imagem gerada. No entanto, no caso real câmeras fotográficas e de vídeo produzem uma distorção radial, aproximadamente simétrica, devido a simetria curvilínea de suas lentes. Estas aberrações óticas, como são conhecidas, podem ser basicamente de dois tipos:

- Distorção em barril: neste caso as linhas das imagens tendem a se encontrar nas bordas dela, como se esta fosse deformada ao projetá-la sobre a superfície externa de uma esfera (figura 6).

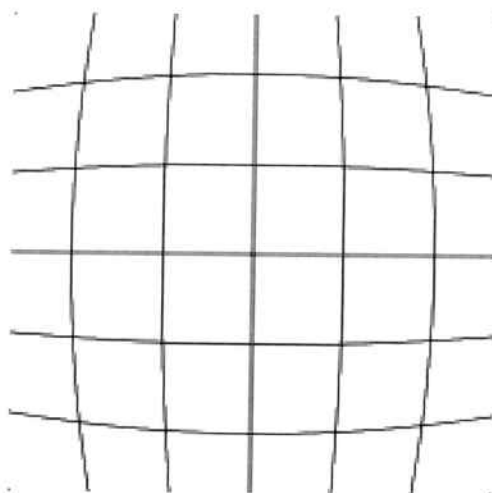


Figura 6: Distorção em barril

³No modelo de câmera de orifício, a imagem é projetada através de um pequena orifício, não existindo uma lente entre a cena e a imagem. Desta forma, a imagem é uma representação fiel da cena observada. Apenas a invertendo nos dois eixos, respeitando a ótica geométrica.

- Distorção em palheiro (do inglês *pincushion*): neste caso as linhas da imagem tendem a se afastar uma das outras ao se distanciar do centro da imagem, como se esta fosse projetada sobre a superfície interna de uma esfera (figura 7).

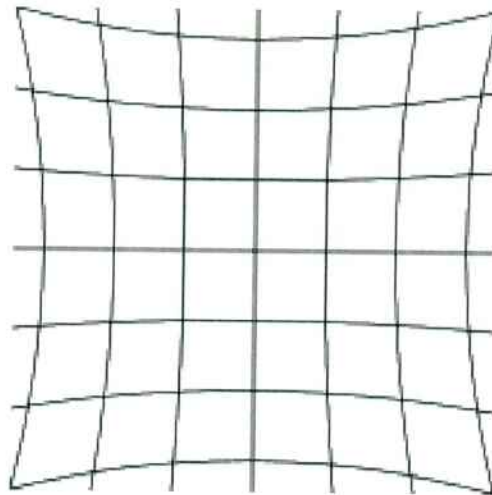
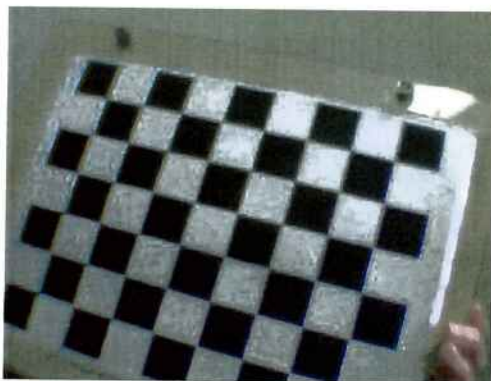
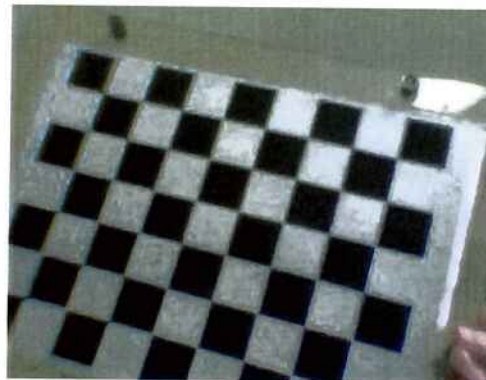


Figura 7: Distorção em palheiro

Desta maneira, a câmera utilizada no projeto, - assim como a grande maioria das *webcams* disponíveis no mercado - produz figuras com distorções do primeiro tipo citado anteriormente. Assim, é necessário remover tais distorções para garantir a fidelidade às dimensões reais dos objetos. Para tanto deve-se identificar todos os níveis das diferentes distorções presentes numa determinada lente de modo que sejam atenuadas ou removidas por completo. Essa remoção é realizada através da calibração da câmera, que consiste em obter os coeficientes de distorção radial e tangencial, denominados parâmetros intrínsecos⁴ da câmera. Esses parâmetros são utilizados para realizar a normalização da imagem[20].



(a) Imagem com distorção



(b) Imagem sem distorção

Figura 8: Imagens ilustrando correção da distorção da imagem

⁴Os parâmetros intrínsecos de uma câmera não dependem da resolução utilizada ou da cena capturada.

As Figuras 8a e 8b mostram imagens obtidas antes da câmera ser calibrada e depois da calibração. Pode-se observar que as linhas do tabuleiro de xadrez estão levemente curvadas na primeira imagem, correspondendo à distorção introduzida pela câmera.

3.4 Considerações Finais

Foram apresentados neste item conceitos de imagem digital e tratamento de imagem. Em especial, foi definido o sistema de referência de uma imagem (figura 1) e representação de seus valores. Também foram abordados os conceitos de limiarização, subtração de imagens e desfocagem, os quais são úteis para o desenvolvimento dos algoritmos apresentados ao longo deste documento.

4 Rastreamento do Olhar

O rastreamento por meio da utilização de câmeras pode ser realizado como apresentado em [18]. Nas seções seguintes será apresentado como é realizado o rastreamento do olhar através deste método, iniciando por uma descrição do modelo considerado para o olho humano, seguida pela explicação de como o rastreamento é realizado a partir do modelo e pré-condição adotada. Então será mostrado como podem ser obtidas as informações necessárias ao funcionamento do rastreamento.

4.1 Modelo do Olho

O olho humano pode ser representado por uma esfera [21], como ilustrado na Figura 9a. Nesta imagem, a esfera exterior, representa o globo ocular, o qual rotaciona a medida em que o olhar é desviado de um ponto a outro numa cena 3D. A esfera menor representa a modelagem da córnea através de um tronco da mesma. Também é ilustrada a localização da pupila, da retina, da fóvea e da íris.

A direção do olhar pode ser estimada através da obtenção do eixo visual do olho. O eixo visual é definido como a reta que conecta a fóvea ao ponto nodal do olho¹, o qual corresponde ao centro de curvatura da córnea na Figura 9a. O ponto em uma cena em três dimensões em que uma pessoa deposita sua atenção é a intersecção deste eixo com a cena. No caso de uma tela de computador, é a intersecção do eixo visual com o plano frontal da tela. O eixo visual e o eixo óptico do olho possuem diferença de direção no espaço de apenas alguns graus, podendo-se obter um a partir do outro[17].

Uma simplificação do modelo (Figura 9b) considera que o olho seja uma esfera uniforme e o centro da pupila esteja localizado na superfície desta esfera. Deste modo, ao rotacionar o globo ocular, o centro da pupila é rotacionado com mesmo ângulo.

¹Ponto do eixo óptico no qual ocorre a intersecção de todas as retas conectando pontos do objeto com seus respectivos pontos na imagem formada no olho[17]

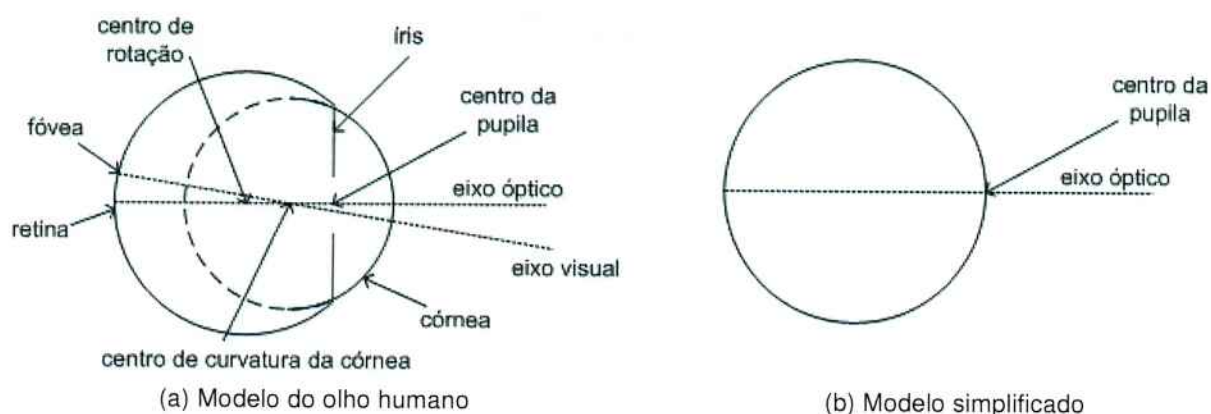


Figura 9: Modelos do olho humano

4.2 Rastreamento

A partir do modelo simplificado da Figura 9b, pode-se implementar um sistema de rastreamento do olhar fazendo-se duas considerações adicionais:

- a existência de um ponto de referência fixo no olho
- posição do usuário do sistema é fixa

Dadas estas pré-condições, não é necessário que se conheça o eixo visual como mencionado na Seção 4.1 para que se estime o ponto na tela em que o usuário olha. Também elimina-se a necessidade de conhecer a distância do usuário à tela do computador. O rastreamento portanto, pode ser realizado a partir da obtenção de imagens do olho do usuário e da extração de duas informações a partir destas imagens: o centro da pupila e a posição do ponto de referência fixo.

De posse destas duas informações, calcula-se a posição do centro da pupila em relação ao ponto de referência fixo, obtendo-se a distância no sistema de coordenadas da imagem. Esta distância ($E = (x_e, y_e)^t$) é convertida para o sistema de coordenadas na tela de um computador ($S = (x_s, y_s)^t$) através de duas equações de segunda ordem:

$$x_e = a_0 + a_1x_s + a_2y_s + a_3x_sy_s + a_4x_s^2 + a_5^2 \quad (4.1)$$

$$y_e = a_6 + a_7x_s + a_8y_s + a_9x_sy_s + a_{10}x_s^2 + a_{11}^2 \quad (4.2)$$

Onde $a_i, i = 0, \dots, 11$ são os coeficiente dos polinômios de grau 2 (lado direito das Equações 4.1 e 4.2). Estes coeficientes são obtidos através de um processo de calibração

antes de cada utilização, obtendo os valores das distâncias $E_i = (x_{e_i}, y_{e_i})^t$ para 9 pontos $S_i = (x_{s_i}, y_{s_i})^t$ distintos na tela ($i = 1, \dots, 9$), que possibilita a resolução de dois sistemas superdeterminados (6×9), um para a Equação 4.1 e outro para a Equação 4.2.

4.3 Obtenção do centro da pupila

Uma fonte de luz infravermelha, a ser posicionada próxima ao eixo óptico² de uma câmera, causa um efeito de realce da pupila ao ser dirigida ao olho, semelhante ao efeito de olho vermelho em fotografias. Isto ocorre devido a retro-refletividade dos olhos [18]. O efeito pode ser observado na Figura 10.



Figura 10: Pupila iluminada por uma fonte de luz infravermelha

Esse efeito da iluminação têm um importante significado para o rastreamento, uma vez que ele destaca a pupila dos demais elementos da imagem, deixando os *pixels* que preenchem sua área com alto valor. Este destaque ajuda na busca pelo centro da pupila, necessário ao rastreamento.

4.4 Ponto Fixo de Referência: Glint

Ao ser posicionado em um ponto fora do eixo óptico de uma câmera, uma fonte pontual de luz infravermelha gera um ponto de reflexão sobre a córnea denominado *glint*. Ao se considerar a captura do reflexo em uma imagem digital, o *glint* é caracterizado por possuir um alto valor de *pixel* em comparação com o restante dos *pixels* da imagem, atingindo na maioria das vezes o topo da escala dos níveis de intensidade de cor. A Figura 11 apresenta um exemplo de *glint*.

²Eixo que passa pelo centro da lente e pelo seu foco



Figura 11: *Glint*: reflexos da luz na córnea

Como a fonte que gera o *glint* é fixa e o usuário também possui posição fixa (por hipótese), pode-se utilizar o ponto gerado na imagem como referência para aplicação do método de rastreamento apresentado na Seção 4.2. O vetor com origem no centro do *glint* até o centro da pupila é denominado vetor *glint*-pupila. A Figura 12 ilustra a representação deste vetor:



Figura 12: Vetor *glint*-pupila

4.5 Considerações Finais

A técnica de rastreamento utilizada por Morimoto em [18] é uma técnica simples que utiliza poucas operações de processamento de imagens e cálculo numérico, exigindo pouco poder de processamento. Apesar de depender de elementos adicionais à câmera (fonte de luz infravermelha), a própria utilização desses elementos torna a detecção de características menos dependentes do ambiente de operação, se comparados a métodos como os empregados em Stiefelhagen [22] e Wang2002 [14].

5 Metodologia

A execução do projeto pode ser dividida nas etapas: pesquisa de trabalhos relacionados, realização de provas de conceito para a detecção da pupila e do *glint*, provas de conceito do método empregado, testes de detecção, testes de rastreamento e avaliação dos resultados. Para a execução de cada uma das etapas foram determinadas iterações semanais para uma divisão em tarefas gerenciáveis. Abaixo são descritas metodologias usadas para o projeto.

5.1 Pesquisa

As pesquisas realizadas tem como base o estudo de *papers*, livros e *sites* sobre o assunto relacionado. A finalidade da pesquisa está na verificação de trabalhos anteriormente executados para que sejam extraídos modelos já conhecidos para que o projeto seja executado. Também tem finalidade de construir a base teórica para que se possa reproduzir trabalhos anteriores e projetar possíveis melhorias.

5.2 Gerenciamento

Para realização do gerenciamento foram desenvolvidas duas atividades: planejamento e reuniões semanais, de acordo com o mostrado na Figura 13. O planejamento consiste em elaborar metas de longo e curto prazo através da elaboração do cronograma do projeto e divisão de tarefas a cada semana. As reuniões semanais servem como meio de avaliar se as metas propostas foram atingidas e definir as próximas ações de acordo com o estágio do projeto que foi alcançado. As iterações do projeto têm base semanal.

As reuniões semanais são divididas em dois tipos de reuniões: com orientador e entre os membros da equipe de trabalho. A primeira é realizada com a finalidade de gerar uma realimentação ao orientador e estabelecer metas para a próxima iteração. Essa periodicidade



Figura 13: Gerenciamento do projeto

dade e regularidade ajuda a manter uma linha de andamento do projeto e ter uma opinião do orientador antes que o projeto se distancie muito do foco. O segundo tipo de reunião - entre os membros do grupo de trabalho - consiste em analisar detalhes técnicos que precisam ser abordados na iteração e dividir internamente tarefas a serem feitas para a próxima iteração.

5.3 Desenvolvimento do Software

Não foi utilizado nenhum método formal de desenvolvimento, tal como *XP* (*Extreme Programming*)[23] ou *RUP* (*Rational Unified Process*)[24], mas um método próprio adaptado às necessidades da equipe. O desenvolvimento do projeto exigiu que fossem realizadas diversas provas de conceito para verificar o funcionamento das técnicas apresentadas na literatura para que se pudesse aplicá-las ao projeto. Deste modo, um método ágil de desenvolvimento fez-se necessário. A metodologia utilizada é baseada na geração de um protótipo inicial que é complementado com funcionalidades adicionais a cada iteração com o objetivo de se atingir o produto final ao término do projeto. Essa metodologia consiste em 4 etapas, sendo a primeira etapa executada uma única vez e as demais executadas a cada iteração de forma cíclica (Figura 14):

- **Prototipagem Inicial:** Realização do protótipo inicial do *software*, contendo funções básicas de acesso ao *hardware* e manipulação dos dados obtidos.
- **Análise:** A cada iteração, cada tarefa é analisada para definir o que deve ser mo-

dificado no protótipo para atingir os objetivos da mesma. Desta análise resulta um projeto simplificado que consiste em diagramas ilustrando a estrutura das classes do projeto após a alteração e como é realizada a interação com os novos componentes para orientar na etapa de codificação. Quando a tarefa a ser realizada não envolve criação de novos componentes, apenas alteração dos existentes sem que haja mudanças nas interfaces existentes, não existe a necessidade de geração desses artefatos.

- **Codificação e Integração:** A codificação é realizada individualmente, com base nas tarefas divididas nas reuniões entre os membros do grupo de trabalho. A Cada elemento do grupo é dada a liberdade de realizar a implementação do jeito que for mais conveniente, desde que os demais sejam informados em caso de mudanças não planejadas na etapa de Análise. Toda codificação é realizada sobre o protótipo inicialmente elaborado, incrementando suas funcionalidades.
- **Teste:** Nesta etapa, as funcionalidades e alterações são testadas a partir de sua integração ao protótipo, realizando a verificação da obtenção dos resultados esperados e possíveis melhorias.

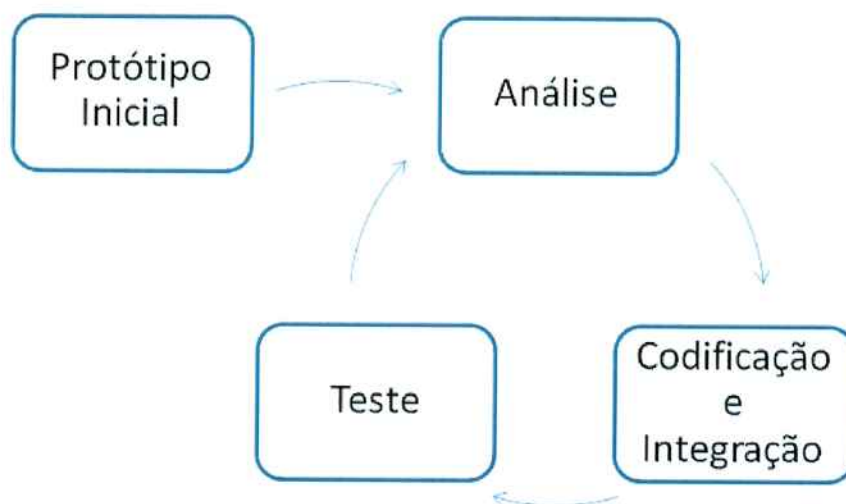


Figura 14: Ciclo de Desenvolvimento

5.4 Desenvolvimento do Hardware

Para o *hardware*, a prototipação é a opção de metodologia, pois resulta naturalmente da obtenção de uma prova de conceito. A construção de protótipos e o melhoramento deles à medida que os resultados esperados forem saindo é um dos meios mais confiáveis de se evitar falhas no projeto. Deste modo, o desenvolvimento do *hardware* é dividido em três atividades, ilustradas na Figura 15:



Figura 15: Atividades do desenvolvimento do *Hardware*

Onde:

- **Preparação inicial:** Todos os componentes de *hardware* necessários ao funcionamento do sistema são montados e um equipamento inicial é obtido.
- **Ajustes:** São realizados ajustes no equipamento para condicionamento do mesmo às condições impostas ao projeto. Esses ajustes são realizados tanto com a ajuda do próprio *software* projetado para o sistema como de *software* de apoio (*i.e.*, mais simples que o sistema) codificado apenas com esta finalidade.
- **Integração:** É realizada a integração do equipamento com o *software* implementado, afim de validar a interoperabilidade entre ambos.

5.5 Programação dos Testes

Os testes são executados a cada iteração para validar o funcionamento do sistema de acordo com as metas determinadas na mesma. Deste modo, está intimamente ligado à integração do *software* com o equipamento. Além dos testes a cada interação, são programados testes para verificação da taxa de detecção alcançada para as características buscadas e precisão do sistema no rastreamento.

5.6 Considerações Finais

Neste capítulo foi apresentada a metodologia utilizada no desenvolvimento do trabalho. A metodologia utilizada visa um desenvolvimento ágil, tanto na parte de *software* como na montagem e configuração do equipamento. Também valoriza a comunicação entre os membros do grupo de trabalho, uma vez que é baseada em iterações de intervalo de tempo pequeno e reuniões frequentes.

6 Projeto e Implementação

6.1 Pesquisa e Ambiente de Desenvolvimento

A etapa inicial abordou a construção da base teórica sobre os métodos utilizados e técnicas de processamento de imagens. A análise das referências bibliográficas seguiu dois critérios para a escolha do modelo a ser aplicado no projeto:

- Procedimento descrito de forma mais detalhada possível;
- Imposição de um número menor de restrições ao usuário.

Da análise, destacou-se dois modelos[17][18]. O primeiro modelo, descrito por Guestrin em [17], apresenta um modelo desenvolvido com base em conceitos de geometria analítica e óptica geométrica. Tal modelo possibilita pequenas movimentações do usuário ($\pm 3cm$ lateral, $\pm 2cm$ vertical e $\pm 4cm$ para frente/trás). Porém, para determinar o ponto na tela em que o usuário olha é necessária a resolução de 3 sistemas não-lineares de graus 3, 4 e 12. O número de equações não lineares e a dificuldade em se obter convergência de valores na sua resolução fez com que fosse optado por um método mais simples, que não permite a movimentação do usuário, descrito no Capítulo 4.

Também nesta etapa, outra frente de trabalho consistiu na preparação das ferramentas a serem utilizadas, definição dos equipamentos necessários. O ambiente de desenvolvimento utilizado consiste na plataforma *Windows*, o *IDE Eclipse* adaptada para suporte a *C++* e o sistema de controle de versão *Subversion*.

6.2 Provas de Conceito de Detecção

Para definir a sequência de passos a serem executados na detecção das pupilas e dos *glints*, realizaram-se provas de conceito para definir a configuração ideal do equipamento e sobre as técnicas de detecção para selecionar a sequência de passos mais adequado. As ações tomadas para atingir a configuração desejada do sistema seguiram a sequência observada da Figura 16.



Figura 16: Sequência de ações utilizadas para determinar a configuração do sistema

A utilização de imagens inicialmente foi aplicada para simular de modo rústico a presença do realce da pupila e do *glint* no olho enquanto o equipamento não estivesse corretamente configurado. Uma vez configurado, o próximo passo foi a realização dos testes de detecção utilizando vídeos, o que possibilitou um paralelismo maior do trabalho, permitindo testar implementações diferentes para detecção de modo a buscar a que melhor funcionasse para obtenção dos centros da pupila e *glint*. Finalmente, também é necessário realizar provas de conceito para o sistema quando opera capturando imagens e imediatamente realiza o rastreamento, para assim verificar o desempenho real do sistema. Para o caso da utilização de vídeos e da captura direta, foram utilizados graus de iluminação do ambiente distintos para verificação de sua influência sobre a detecção. Outro ponto analisado é a correta asserção ao sistema sobre o estado da detecção, isto é, sobre quais elementos o sistema havia detectado comparado com o que realmente ocorreu, para auxiliar na identificação dos casos de falso positivo (indicação errônea de detecção da pupila e *glint*) e falsos negativos para que pudessem ser corrigidos.

6.3 Processo de teste

O processo empregado classifica-se em 3 categorias principais, onde cada uma delas seguiu uma estratégia de testes, enumeradas a seguir:

- Testes de modelo: nesta categoria se encaixam todas as classes e métodos desenvolvidos para os diferentes passos de execução do sistema. Cada classe, ou conjunto destas quando aplicável (caso em que o encadeamento de classes se faz

necessário para a realização de uma determinada tarefa, onde a simplificação do problema em partes auxilia sua resolução), são testadas com dados forjados de entrada com valores em faixas plausíveis do que se espera obter na realidade. Após esta etapa preliminar usada essencialmente na depuração de erros grosseiros, se faz os testes com dados obtidos de situações reais.

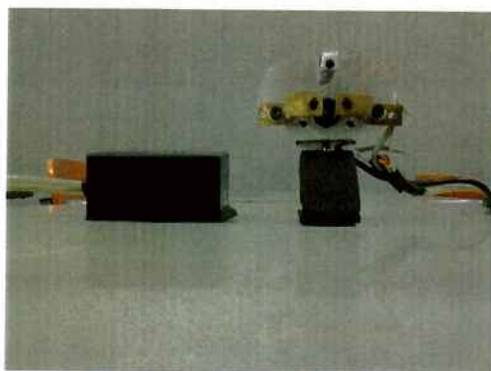
- Testes de detecção: medição das taxas de acerto, falsos negativos e falsos positivos na detecção da pupila e do *glint*.
- Testes de precisão: após a obtenção dos primeiros resultados finais de rastreamento foram feitos teste manuais para se determinar o grau de precisão do ponto determinado em tela. Este teste consiste em se dividir a continuamente em regiões retangulares de mesmo tamanho, comparando-se o resultado obtido como o real.

Os resultados dos testes de detecção e precisão são apresentados na Seção 7.1.

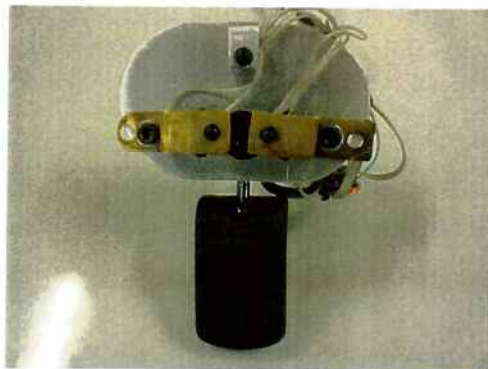
6.4 Hardware

O equipamento utilizado no projeto foi desenvolvido para auxiliar o método de detecção dos olhos e do rastreamento do olhar. De fato o equipamento realiza papel indispensável nos métodos utilizados para ambos os procedimentos. Ele consiste de uma placa fixada junto da *webcam* que contém *LED's* localizados em posições bem definidas com relação ao eixo da lente da câmera (Figuras 18a,18b e 18c). Esses por sua vez, estão ligados através de um cabo à uma placa onde se localiza o circuito integrado FT232RL [25] (Figura 18d), que realiza o interfaceamento entre portas USB (*Universal Serial Bus*) e portas UART (*Universal Asynchronous Receiver/Transmitter*). Através de um programa sincronizado com a obtenção dos quadros de vídeo da *webcam*, é feito o acionamento sequencial e ritmado dos *LED's*, de forma a se obter um quadro iluminado por um conjunto e o seguinte iluminado pelos demais. Na Figura 17 é apresentado o esquema da placa de sincronismo projetada.

Toda a estrutura de suporte dos *LED's* na câmera foi feita em alumínio e com tamanho reduzido para simultaneamente propiciar um suporte resistente e leve, proporcionando fácil manuseio ao usuário. Além disso, os *LED's* utilizados são emissores de luz no espectro do infravermelho próximo (comprimento de onda próximo a 850nm), tornando essa radiação invisível ao olho humano, não incomodando o usuário durante a utilização do sistema. Devido a utilização de *LED's* infravermelhos é necessário utilizar uma câmera cujo filtro



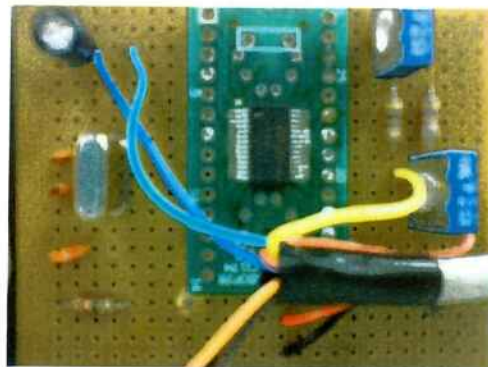
(a) Equipamento



(b) Vista Frontal



(c) Vista Lateral



(d) Circuito de sincronismo

Figura 18: Imagens do equipamento

6.5 Calibração da Câmera

Para diminuir a influência das distorções introduzidas na captura da imagem pela câmera, de modo que as medidas a serem obtidas (centro da pupila e *glints*) tenham melhor precisão, é necessário calibrar a câmera utilizada. A calibração da câmera consiste na obtenção de parâmetros intrínsecos e no uso destes parâmetros para remover as distorções introduzidas pela lente na imagem. Tais distorções correspondem às distorções em barril e palheiro (*pincushion*), apresentadas na Seção 3.3.

O processo de calibração consiste na aquisição de dados via uma coleção de imagens de onde se extraem pontos de um objeto com dimensões conhecidas. Esses dados são utilizados para obter os parâmetros desejados do equipamento.

A biblioteca *OpenCV* possui um conjunto de funções preparadas para tratar esse. Uma vez obtidos os parâmetros da câmera, basta encadear um passo após a obtenção das imagens, como é ilustrado na Figura 19. O processo de calibração só precisa ser executado uma única vez.



Figura 19: Passo adicional na sequência de obtenção de imagens

6.6 Software e Integração

O *software* desenvolvido é responsável pela realização do sincronismo entre a obtenção da imagem e os *LED's* auxiliares, tratamento e análise das imagens obtidas pela câmera e cálculo da coordenada da tela através da aplicação do modelo estudado. Um dos componentes de *software* externos utilizados consiste na biblioteca *OpenCV*. O *OpenCV* é uma biblioteca multiplataforma de código aberto, contendo funções para processamento de imagens, originalmente desenvolvida pela Intel. Também possui funções úteis para resolução numérica de sistemas lineares, podendo ser utilizada para resolver o sistema desenvolvido em [18]. Outro componente externo se trata de uma extensão ao *OpenCV*, a *cvBlobsLib*, responsável pela detecção de componentes conectados na imagem. Os componentes do sistema são os ilustrados na Figura 20:

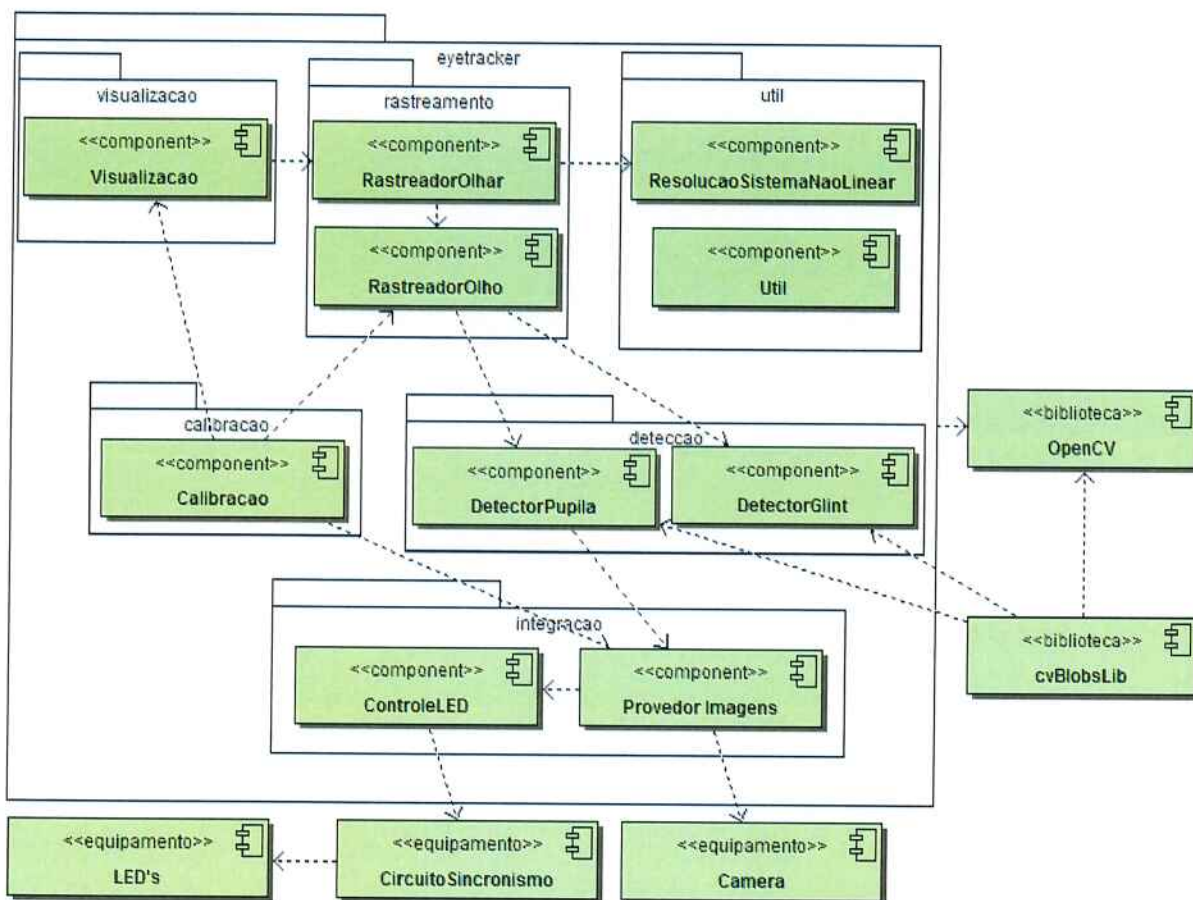


Figura 20: Componentes do sistema

- **Detecção:** Agrega os componentes responsáveis pela detecção do centro das pupilas e do *glint* a partir das imagens obtidas.
- **Rastreamento:** Responsável pelo cálculo das coordenadas na tela a partir dos resultados obtidos.
- **Integração:** Contém componentes de acesso e controle dos equipamentos de *hardware*.
- **Calibração:** Componentes da calibração da câmera, responsável pela eliminação das distorções existentes na imagem introduzidas pela câmera.
- **Visualização:** Este pacote agrega componentes de apresentação dos resultados do rastreamento, isto é, pela visualização da coordenada em que foi detectado o foco do usuário.
- **Util:** Composto por um componente que agrega funções utilitárias utilizadas por diversos outros componentes do sistema e também componentes utilizados para a resolução de sistemas não-lineares. Foi implementado o módulo para resolução de sistemas não lineares utilizando o método de Newton [28].
- **Bibliotecas externas:** Referentes ao *OpenCV* e ao *cvBlobsLib*, bibliotecas utilizadas para auxiliar na captura, tratamento, análise e exibição de imagens.
- **Equipamentos:** Correspondem aos equipamentos físicos necessários para a implementação do projeto: câmera digital, circuito de sincronismo e *LED's* infravermelhos.

6.6.1 Diagrama de Classes do Sistema

Nas figuras 21 e 22 são mostradas como as classes do sistema estão relacionadas.

6.6.2 Diagrama de Sequência do Sistema

O diagrama de sequência da Figura 23 mostra o processo de obtenção das coordenadas da tela a partir da utilização das duas imagens. Por simplificação, foram omitidas mensagens reflexivas.

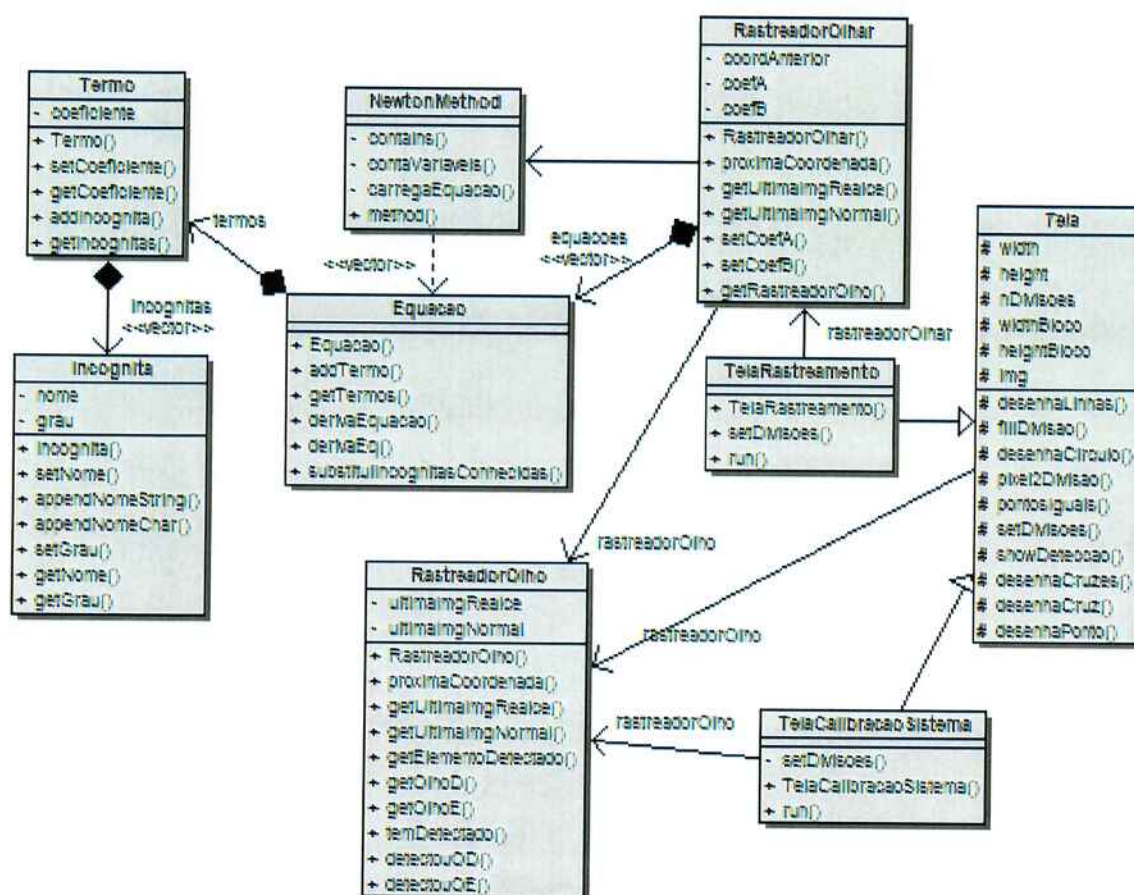


Figura 21: Diagrama de classes do sistema - A

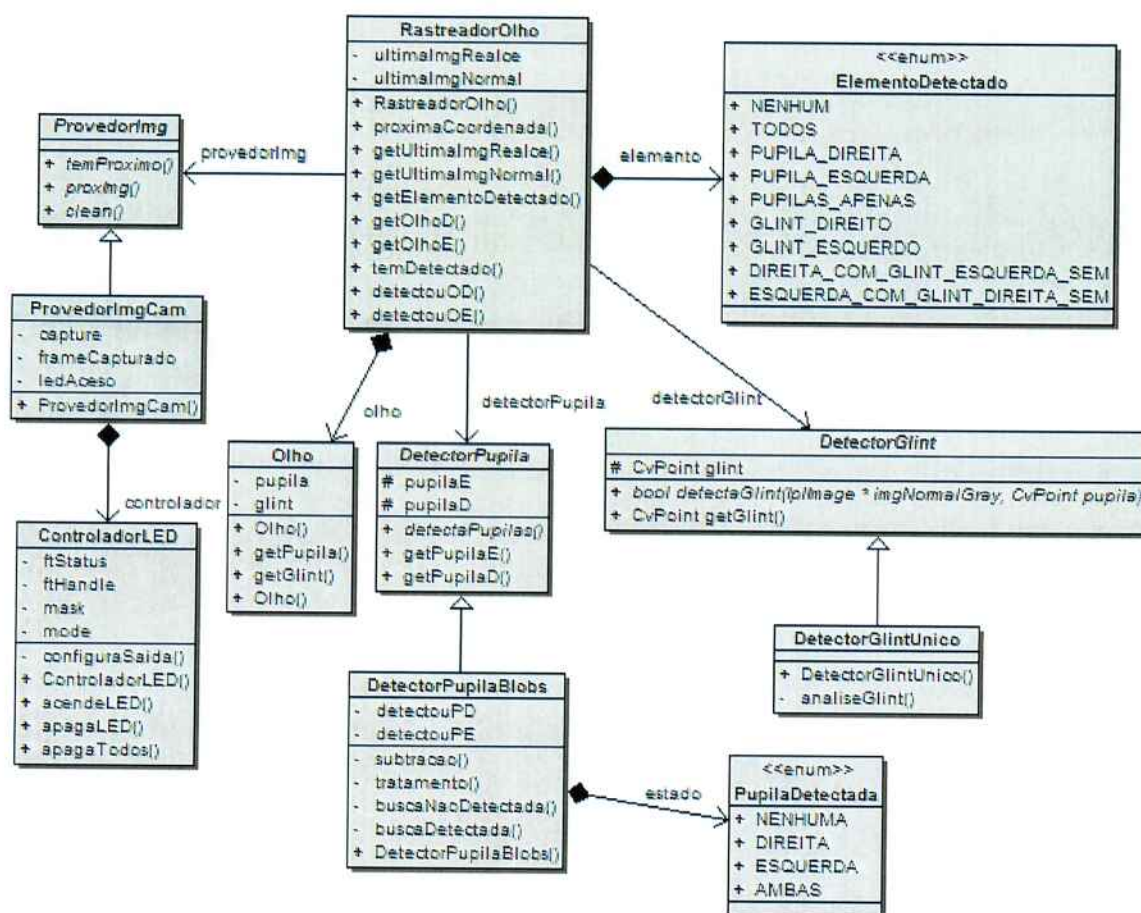


Figura 22: Diagrama de classes do sistema - B

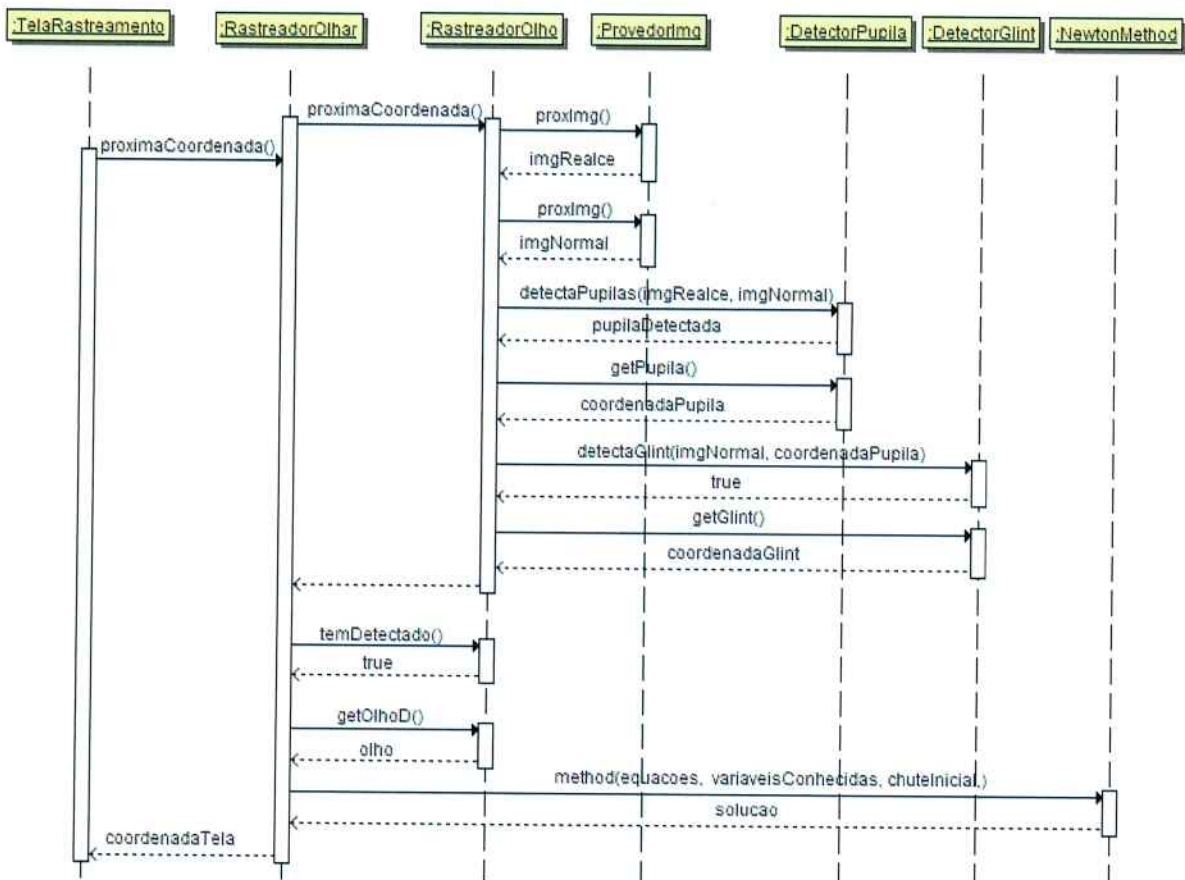


Figura 23: Diagrama de seqüência do software de rastreamento

6.6.3 Estados da Detecção da Pupila

A detecção da pupila ocorre inicialmente realizando a busca na imagem obtida após todos os tratamentos mencionados no item 6.6.4. Ao conseguir detectar pelo menos uma das pupilas, guarda-se a informação dos centros das pupilas que foram encontradas. Desta forma, a próxima busca de cada pupila encontrada é realizada somente em uma pequena região da imagem ao redor do último centro. Ao final dessa segunda busca, compara-se o valor do centro atual e o centro anteriormente detectado. Caso a diferença entre eles seja maior que um certo valor configurável manualmente, considera-se que a pupila não foi corretamente encontrada, voltando-se ao estado inicial. Esta análise é útil para tratar casos em que a detecção não detecta corretamente, recuperando-se rapidamente na busca seguinte. O diagrama de estados da detecção pode ser observado na figura 24.

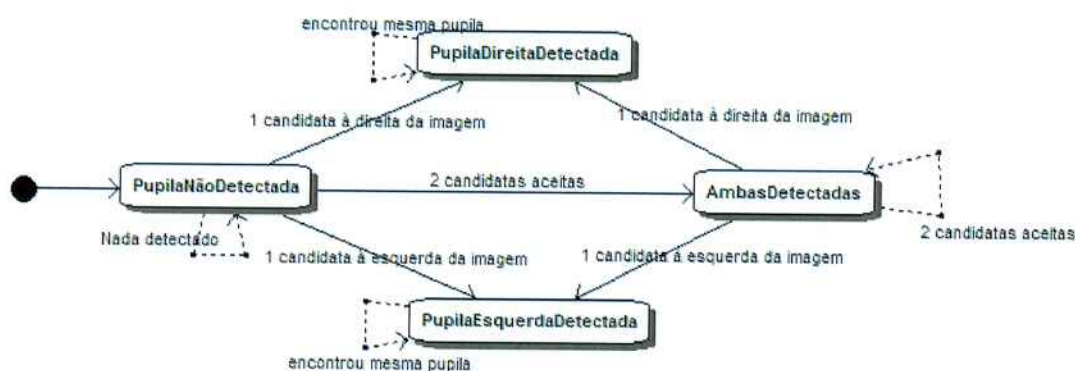


Figura 24: Diagrama de estados da detecção da pupila

Vale notar que em qualquer estado, caso não seja detectada nenhuma das pupilas volta-se para o estado inicial e o algoritmo volta a procurar pelas pupilas na imagem inteira.

6.6.4 Algoritmo de Detecção da Pupila

O algoritmo de obtenção do centro da pupila utiliza o conceito apresentado no item 4.3, seguindo os passos descritos nos itens seguintes:

- **Conversão para escala de cinza:** A captura de imagens realizada pela câmera retorna as imagens em formato de imagem colorida. Porém, as operações e análise aplicadas pelo algoritmo apenas necessitam de uma imagem em escala de cinza. A conversão adequa a imagem para os passos seguintes e elimina *overhead* no armazenamento e processamento.

- **Subtração:** A operação de subtração é realizada entre a imagem contendo a pupila com realce e a imagem com a pupila sem o realce. O objetivo deste passo é fazer com que a imagem resultante contenha valores de *pixel* próximos do zero exceto na localização da pupila.
- **Limiarização:** O objetivo desta operação é particionar os valores obtidos pela subtração, de modo que a pupila seja caracterizada por uma região contínua de valores de pixel máximo.
- **Detecção de elementos conectados:** Neste passo, todos os elementos conectados da imagem resultante da limiarização são obtidos para análise posterior.
- **Análise:** É feita a comparação dos elementos, verificando candidatos que melhor satisfazem as condições impostas de área na imagem, distância horizontal entre os eixos. Estes parâmetros são passíveis de ser adquiridos através da calibração. Após encontrar os candidatos adequados, obtém-se seu centro na imagem.

A ocorrência da presença de elementos indesejáveis pode ser observada na sequência de imagens da Figura 25. Observa-se que logo após a limiarização ainda existem elementos que não correspondem à pupila, mas realizando-se a detecção dos componentes conexos da imagem e filtragem pela área, estes elementos são eliminados completamente. Do mesmo modo, elementos maiores em área do que a pupila seriam eliminados.

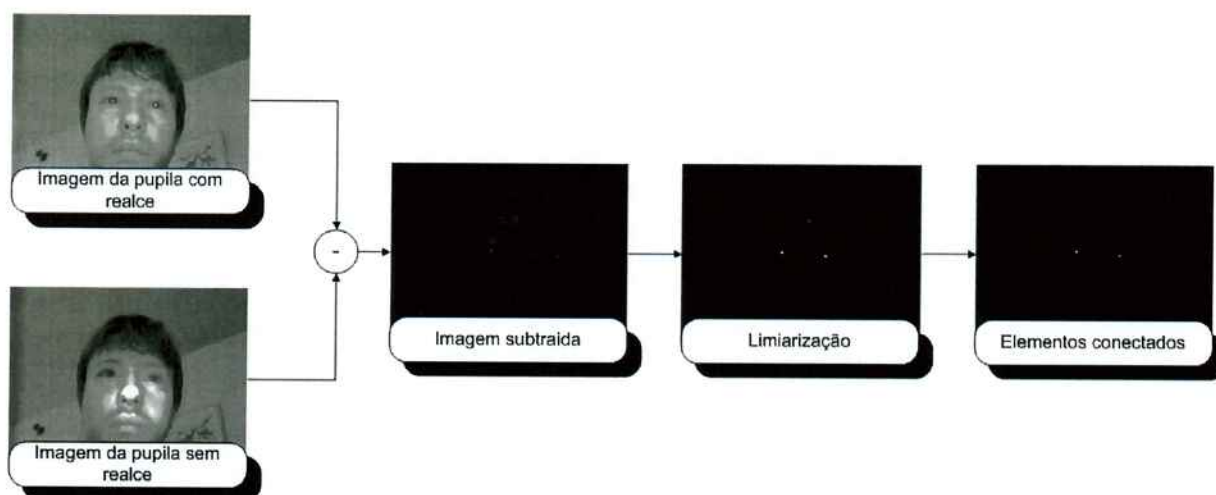


Figura 25: Passos da detecção da pupila: pupila ainda não detectada

A Figura 25 ilustra os passos realizados para detecção da pupila no caso em que nenhuma pupila foi previamente encontrada. Isto ocorre na primeira vez que o *software* realiza a busca e quando a localização da pupila não é detectada em alguma dupla de

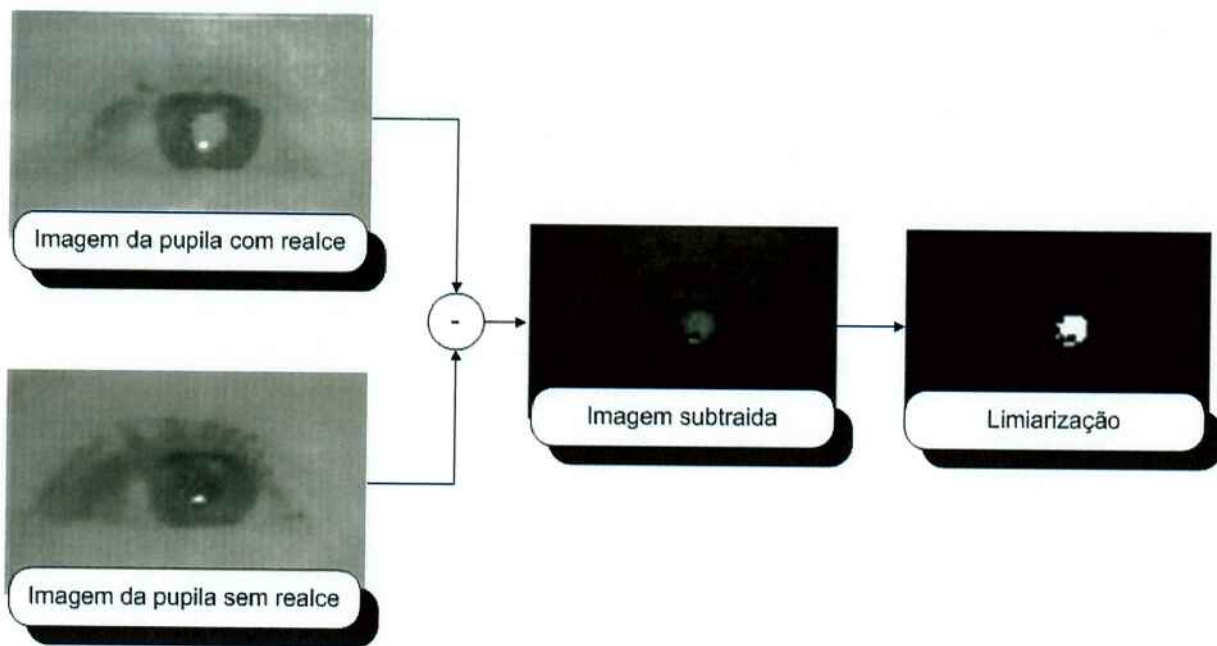


Figura 26: Passos da detecção da pupila: Pupila já detectada anteriormente

quadros (com realce e sem realce da pupila). A Figura 26 mostra a área em volta da qual é feita a busca da pupila nos casos em que esta foi encontrada anteriormente.

6.6.5 Algoritmo de Detecção do Glint

O *glint* pode ser localizado a partir da imagem com a pupila não iluminada. Pela imagem do *glint* presente na Figura 27 pode-se observar que o *glint* é mais claro que o resto da imagem. A detecção do *glint* tem como pré-requisito a detecção da pupila. Assim, pode-se limitar a busca do *glint* na área da imagem ao redor das coordenadas encontradas da pupila. Para facilitar a detecção - buscando a eliminação de ruídos - aplica-se a operação de desfocagem sobre a imagem contendo o *glint*, onde cada ponto da imagem é substituído pelo valor da mediana da região em volta dele. A idéia é diminuir a intensidade do *glint* nesta imagem para que, quando for feita a subtração (imagem original e imagem desfocada), os mesmos se destaquem do resto da imagem. Como a área perto do olho que possui níveis elevados devido à iluminação da face é muitas vezes maior do que a área ocupada por um ponto apenas, a influência do borramento sobre ela é mínima e na subtração essa área tende a se sobressair menos.

Após a subtração das duas imagens, é realizado o processo de limiarização da imagem, o que resulta no isolamento do *glint*, como pode ser visto na sequência. Seguindo a limiarização, é realizada a detecção de elementos conectados. Com esse passo adicional, é possível realizar uma análise mais precisa da imagem, eliminando eventuais ruídos que

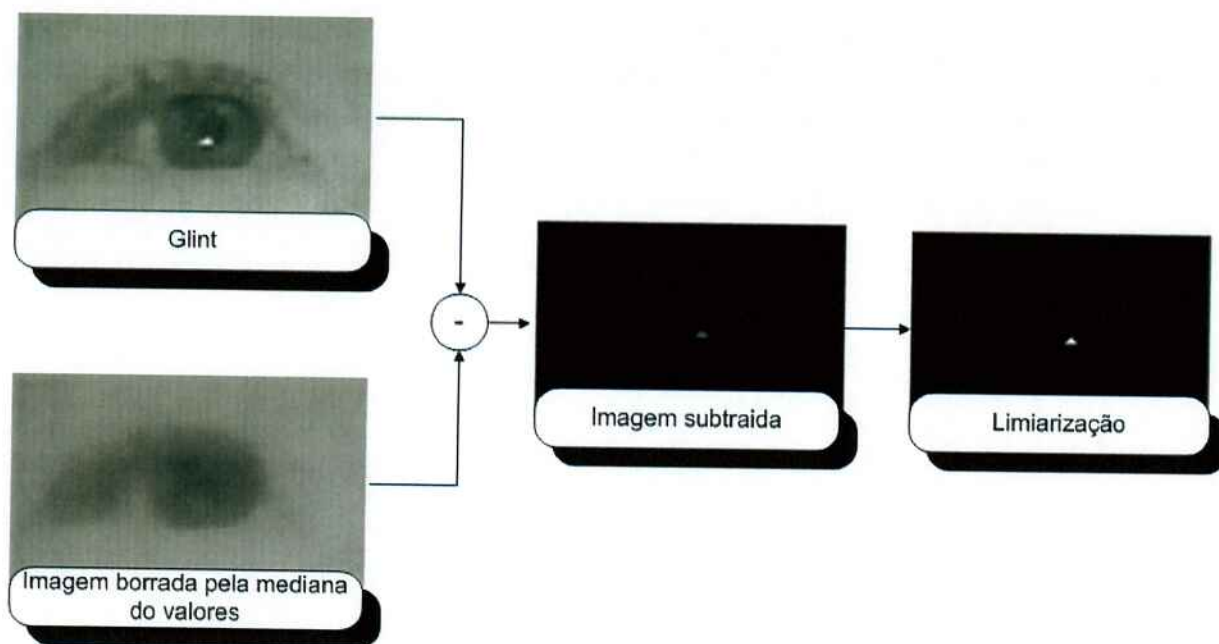


Figura 27: Etapas da detecção do glint

podem aparecer principalmente devido a problemas de iluminação.

A análise posterior à detecção de componentes conexos consiste em verificar a localização dos candidatos a *glint*. Verificou-se que geralmente, quando existe mais de um candidato a *glint*, o ponto realmente buscado é o que se localiza mais perto da pupila. Deste modo, na análise dos candidatos a *glint*, considera-se primeiro os candidatos localizados mais próximo do centro da pupila.

6.7 Calibração do Sistema

A calibração do sistema consiste na exibição de uma tela em que são dispostos sequencialmente nove pontos Figura 28. Para cada um destes pontos, obtém-se as coordenadas do centro da pupila (x_p, y_p) e do *glint* (x_g, y_g) , calculando-se o vetor *glint-pupila* definido na seção 4.4. De posse do valor do vetor *glint-pupila* e das coordenadas em que foi exibido o ponto observado pelo usuário, utiliza-se as equações 4.1 e 4.2 para se obter um sistema super-determinado com 9 equações e 6 incógnitas. A resolução deste sistema permite obter os coeficientes $a_i, i = 0, \dots, 11$ de 4.1 e 4.2. Para que se tenha certeza de que os pontos detectados estão corretos é solicitada a confirmação do usuário a cada coordenada obtida. As figuras 28a e 28b ilustram a tela utilizada para realizar a calibração.

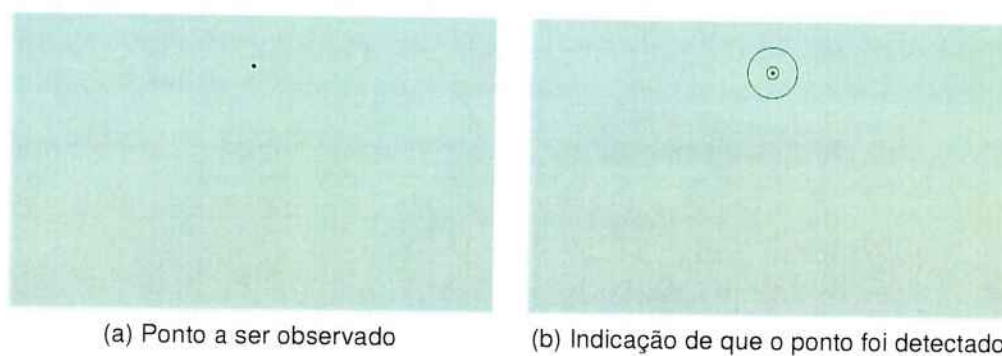


Figura 28: Imagens da tela de calibração

6.8 Considerações Finais

Neste capítulo foram abordados detalhes dos procedimentos adotados na implementação, bem como a estrutura do *software* projetado e do equipamento utilizado. Também explicitou-se como é realizado o procedimento de calibração do sistema.

7 Conclusão

Serão apresentadas as conclusões sobre este trabalho, analisando a viabilidade da implementação de um sistema deste tipo, as limitações do sistema implementado e trabalhos futuros a serem realizados para melhorar o sistema de rastreamento.

7.1 Resultados e Viabilidade do Sistema

Para a realização dos testes de rastreamento, foram utilizados monitores de 15" *widescreen* 16:10. O *software* foi executado em uma máquina Intel Core 2 Duo T5550 com 3GB de RAM DDR2@667Mhz rodando o sistema operacional Windows Vista.

Ao final da redação deste documento, os resultados obtidos para a detecção das pupilas do usuário foram satisfatórios para situações em que a iluminação ambiente não fosse muito intensa ou proveniente da luz solar. Os testes realizados indicam uma taxa de acerto na detecção delas superior a 60%. No entanto, a detecção do *glint* apresenta resultados muito abaixo do esperado, embora os testes não tenham sinalizado uma taxa de acerto constante. Razões para a ocorrência deste cenário provém da baixa resolução e qualidade das imagens obtidas da câmera e da precisão das técnicas de tratamento e análise de imagens.

Onde as detecções ocorrem da maneira correta, o método utilizado para o rastreamento respondeu bem aos dados de entrada. No entanto, mesmo este rastreamento não pôde ter seu grau de precisão avaliado devido à influência das incertezas embutidas nos pontos obtidos nos passos anteriores. A saber, a localização das pupilas e *glints*. Deste modo, os resultados ficaram abaixo da expectativa da equipe, sendo que a viabilidade do sistema na situação atual não é válida. O número de quadros em que o sistema fica inativo devida à falhas de detecção das características buscadas, o torna inadequado para operação à taxas interativas.

Na tabela 1, temos os resultados obtidos da detecção da pupila, e na tabela 2, temos

Pupila							
Vídeos	Total	Encontrou	Falso +	Falso -	Não Encontrou	%Encontrado	%Acerto
teste1	50	37	-	-	13	74,00%	100,00%
teste2	50	45	-	-	5	90,00%	100,00%
teste3	50	17	12	-	21	34,00%	76,00%
teste4	50	14	18	-	18	28,00%	64,00%
teste5	50	5	1	-	44	10,00%	98,00%
teste6	50	43	-	-	7	86,00%	100,00%
teste7	50	43	-	-	7	86,00%	100,00%
teste8	50	42	-	-	8	84,00%	100,00%

Tabela 1: Resultado de detecção da pupila

Glint							
Vídeos	Total	Encontrou	Falso +	Falso -	Não Encontrou	%Encontrado	%Acerto
teste1	37	8	26	-	3	21,62%	29,73%
teste2	45	36	4	-	5	80,00%	91,11%
teste3	17	6	9	1	1	35,29%	41,18%
teste4	14	1	13	-	0	7,14%	7,14%
teste5	5	4	-	-	1	80,00%	100,00%
teste6	43	42	1	-	0	97,67%	97,67%
teste7	43	30	10	-	3	69,77%	76,74%
teste8	42	26	12	-	4	61,90%	71,43%

Tabela 2: Resultado de detecção do *glint*

os resultados da detecção do *glint*. O significado de cada campo é dado abaixo:

- Longe: casos em que o usuário fica a uma distância maior da câmara, de aproximadamente 60cm;
- Perto: casos em que o usuário fica a uma distância menor da câmara, de aproximadamente 20cm;
- OK: número de vezes em que a pupila ou *glints* foram detectados corretamente;
- Falso +: número de vezes em que o sistema acusou ter detectado erroneamente;
- Falso -: número de vezes em que o sistema acusou não ter detectado erroneamente;
- Não encontrou: número de vezes em a pupila ou *glints* não foram detectados;
- Encontrado: porcentagem de detecção correta
- Acerto: porcentagem de acerto na avaliação da detecção (se foi conseguido detectar ou não)

7.2 Limitações

O sistema proposto possui algumas limitações decorrentes do método de rastreamento utilizado em conjunto com o equipamento projetado. A técnica empregada, extraída de [18], parte do princípio que o usuário possui sua localização fixa com relação à câmera utilizada no sistema (e conseqüentemente o centro de rotação dos olhos), pois leva em consideração a variação da distância do centro da pupila ao centro do *glint* do *LED* de referência do sistema, e supondo-se que a reflexão deste varia muito pouco, já que a córnea segue uma curvatura próxima da do centro de rotação do olho e, portanto, mantém a superfície "idêntica" a cada movimento, pode-se inferir a variação angular associada a variação do ponto observado na tela. Este mapeamento obtido com uma amostragem de pontos conhecidos no momento da calibração, e com a resolução de um sistema de equações que modelam o sistema é estimada a localização do ponto observado para qualquer movimento do olho, sendo que apenas um é suficiente, uma vez que ambos fitam o mesmo ponto.

Desta forma é necessário que o usuário limite seus movimentos ao máximo, visto que a calibração se faz necessária a cada nova inicialização do sistema ou mudança de postura do usuário. É recomendado que se utilize um suporte, tal como uma cadeira com encosto para que se consiga manter uma postura constante de forma confortável por longos períodos intervalos de tempo.

A outra limitação do sistema está ligada ao equipamento e parte dos módulos e técnicas utilizadas nas etapas preliminares de processamento de imagem para se encontrar o centro da pupila e *glints*, já que é feita a suposição que o usuário esteja num ambiente na ausência de outras fontes de luz infravermelha, faixa de frequência emitida pelos *LED's* usados. Assim fontes *glints* provocados por outros objetos ou mesmo devido a utilização de lentes de contato ou óculos pelo usuário não foram levados em consideração no projeto, sendo este último um dos maiores desafios encontrados em qualquer trabalho de rastreamento do olhar. Também é altamente recomendado que se o usuário utilize o equipamento num cômodo que não seja fortemente iluminado por luz ambiente, principalmente que possua janelas com o sol focando diretamente o equipamento, pois a radiação infravermelha deste irá saturar as imagens capturadas, uma vez que a câmera utilizada teve seu filtro removido por necessidade de detectar os *LED's* sem afetar a utilização pelo usuário.

7.3 Trabalhos Futuros

O projeto realizado ao longo deste ano sofreu diversas alterações desde sua idealização e primeira formulação apresentada na primeira entrega oficial anual. Muitas delas se devem a reavaliação de riscos e metas devido as dificuldades que foram surgindo ou foram sendo previstas ao longo do desenvolvimento. Uma destas modificações diz respeito ao método de rastreamento do olhar utilizado. Inicialmente pretendia-se reduzir ao máximo as restrições impostas ao usuário, criando-se um sistema que permite o livre movimento da cabeça do usuário, permitindo que este adquira qualquer postura e mova-se sem qualquer restrição de direção ou comportamento. Em [17] é apresentado um método que satisfaz todas essas restrições. No entanto, a complexidade de obtenção da informação desejada através do modelo apresentado, pois este faz uso de sistemas de equações não lineares de ordem até 6 vezes maior que o descrito em [18], o que o torna mais sensível à variações nas amostras obtidas. Deste modo, a opção foi por um método mais limitado. Como trabalho futuro seria necessário aprimorar o modelo utilizado de modo que este passe a permitir movimentações do usuário, sem que seja introduzida muita complexidade matemática.

Outro item importante que ficou de fora desta versão, mas que inicialmente figurava como um dos objetivos a ser perseguido na versão preliminar foi à possibilidade do uso do sistema por pessoas usuárias de óculos. No entanto as pesquisas realizadas na primeira iteração não apresentaram um método preferido e com bons resultados, pelo menos para sistemas que permitam a livre movimentação da cabeça, que consiga se livrar dos ruídos e *glints* espúrios inseridos pelos reflexos das lentes. Para a implementação atual até seria possível a utilização de um método para se contornar as dificuldades impostas pelos óculos, como o apresentado em [16], no entanto ele exige o posicionamento sempre no mesmo local dos olhos, uma vez que o sistema de rastreamento se monta sobre a cabeça ou com esta sobre um suporte. O sistema atual permite que o usuário defina sua posição fixa durante a calibração, porém essa posição pode variar a cada utilização.

Por fim, o aprimoramento dos passos de tratamento de imagem e classificação das figuras resultantes seriam modificações importantes uma vez que os resultados atuais ainda são razoavelmente dependentes de variações ambientais como luminosidade provenientes de fontes externas tais como luz solar e lâmpadas fluorescentes.

7.4 Considerações Finais

O projeto de um sistema de rastreamento do olhar baseado em câmera exige especial atenção à qualidade das imagens capturadas. Como qualidade da imagem entende-se a resolução da mesma e a eliminação de ruídos adicionados através de fontes de luz diversas (sol, lâmpadas, etc.) ou movimentação do usuário¹.

¹A movimentação do usuário introduziria ruídos por causa da operação de subtração, uma vez que as imagens teriam mais elementos diferentes em uma mesma região da imagem

Referências

- [1] GONZALEZ R. C.; WOODS, R. E. *Digital Image Processing*. 2. ed. [S.l.]: Prentice Hall, 2001.
- [2] FISHER, R. *Image Processing Learning Resources*. Disponível em: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hipr_top.htm>.
- [3] MICROSOFT Vista Speech Recognition. Disponível em: <<http://www.microsoft.com/windows/windows-vista/features/accessibility.aspx?tabid=2&catid=1#speech>>.
- [4] ANDAMIRO Pump it Up. Disponível em: <<http://www.andamiro.com/en/products/amusement00b.html>>.
- [5] NINTENDO Wii. Disponível em: <<http://www.nintendo.com/wii/what>>.
- [6] EYETOY. Disponível em: <<http://www.eyetoy.com/index.asp?pageID=18>>.
- [7] SMITH J. D.; GRAHAM, N. Use of eye movements for video game control. p. 8, 2006.
- [8] ENQUIRO Research. Disponível em: <<http://www.enquireresearch.com/eye-tracking-examples.aspx>>.
- [9] TOBII. Disponível em: <<http://www.tobii.com>>.
- [10] KISHI A. K.; FANG, C. Ferramenta para análise de informações do olhar. p. 35, 2007.
- [11] ALLISON R.; EIZENMAN, M. T. R. D. S. J. F. R. C. Head and eye tracking for study of the vor during natural head motion. p. 2, 1994.
- [12] ALLISON R. S.; EINZENMAN, M. C. B. S. K. Combined head and eye tracking system for dynamic testing of the vestibular system. *IEEE Transactions on Biomedical Engineering*, v. 43, n. 11, p. 10, Novembro 1996.
- [13] ANDERSON J. R.; DOUGLASS, S. M. M. B. M. D. Eye tracking the visual search of click-down menus. p. 8, Maio 1999.
- [14] WANG J. G.; SUNG, E. Study on eye gaze estimation. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, v. 32, n. 3, p. 19, Junho 2002.
- [15] JI Q.; ZHU, Z. Non-intrusive eye and gaze tracking for neural human computer interaction. 2003.
- [16] EBISAWA, Y. Improved video-based eye-gaze detection method. *IEEE Transactions on Instrumentation and Measurement*, v. 47, n. 4, p. 8, Agosto 1998.

- [17] GUESTRIN, E. D. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering*, v. 53, n. 6, p. 10, Junho 2006.
- [18] MORIMOTO C. H.; KOONS, D. A. A. F. M. Frame-rate pupil detector and gaze tracker. p. 5.
- [19] MORFOLOGIA Matemática. Disponível em: <<http://www.mmorph.com/html/index.html>>.
- [20] A method for Camera Calibration. Disponível em: <<http://www.softnet.tuc.gr/adeli/Calib.html>>.
- [21] MEYER A.; BÖHME, M. M. T. B. E. A single-camera remote eye tracker. p. 4.
- [22] STIEFELHAGEN R.; YANG, J. W. A. Tracking eyes and monitoring eye gaze. In: . [S.l.: s.n.], 1997. p. 3.
- [23] EXTREME Programming: A Gentle Introduction. Disponível em: <<http://www.extremeprogramming.org/>>.
- [24] RATIONAL Unified Process. Disponível em: <<http://www-01.ibm.com/software/awdtools/rup/?STACT=105AGY59&SCMP=WIKI&ca=dtl-08rupsite>>.
- [25] FDTICHIP FT232. Disponível em: <<http://www.ftdichip.com/Products/FT232R.htm>>.
- [26] SONIX SN9C202. Disponível em: <[http://www.sonix.com.tw/sonix/product.do?p=SN9C202%20\(Phase%20out\)](http://www.sonix.com.tw/sonix/product.do?p=SN9C202%20(Phase%20out))>.
- [27] OPENCV. Disponível em: <<http://www.sourceforge.net/projects/opencvlibrary>>.
- [28] MIT Solving Systems of Nonlinear Algebraic Equations using Newton's Method. Disponível em: <<http://web.mit.edu/10.001/Web/CourseNotes/Newton2.htm>>.

APÊNDICE A – Suporte para os LED's

