

UNIVERSIDADE PRESBITERIANA MACKENZIE

Valmir Fernandes Junior

**Detecção e Rastreamento de Íris para
Implementação de Interface Homem-
Computador**

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

VALMIR FERNANDES JUNIOR

DETECÇÃO E RASTREAMENTO DE ÍRIS PARA
IMPLEMENTAÇÃO DE UMA INTERFACE HOMEM-
COMPUTADOR

Dissertação apresentada à Universidade
Presbiteriana Mackenzie como requisito
parcial para a obtenção do título de Mestre
em Engenharia Elétrica.

ORIENTADOR: PROF. MAURÍCIO MARENGONI

F363d Fernandes Junior, Valmir.
Detecção e rastreamento de íris para implementação de interface
homem-computador / Valmir Fernandes Junior – 2010.
51 f. : il. ; 30 cm

Dissertação (Mestrado em Engenharia Elétrica) – Universidade
Presbiteriana Mackenzie, São Paulo, 2010.
Bibliografia: f. 51.

1. Detecção. 2. Rastreamento. 3. Íris. 4. Interface. 5. Problemas
de mobilidade. I. Título.

CDD 005.8

AGRADECIMENTOS

Agradeço, sobretudo a **Deus** por todas as coisas e por me renovar quando me deixei fraquejar. Aos meus pais **Seu Valmir** e **Dona Julinda** pelo apoio incondicional. À minha esposa **Graça** pela paciência. À tia **Otília** pela força. Aos meus irmãos **Rodrigo** e **Juliana** pelo entusiasmo. Ao meu grande amigo **Luciano** pelas boas idéias. À meus enteados **André** e **Dani** e minha sobrinha **Taty** pela fé. Ao meu orientador **Professor Maurício** por não desistir nunca.

RESUMO

Este trabalho apresenta uma técnica de detecção e rastreamento de íris para ser utilizada em uma interface homem-computador que permita pessoas com mobilidade restrita, inclusive sem mobilidade nos ombros, possam controlar o cursor do mouse com o movimento dos olhos, sem a necessidade de adquirir equipamentos caros. A única entrada de dados utilizada é uma “webcam” simples sem auxílio de zoon ótico, iluminação especial ou fixação da face do usuário. A movimentação do mouse dar-se-á de maneira direta, ou seja, o ponteiro do mouse será direcionado para a região estimada pela técnica. Para a realização dos testes de detecção de íris foram utilizadas 60 imagens. Em 90.83% dos casos as íris foram encontradas corretamente, 4.17% dos casos não foram identificados e ocorreram 5% de falsos positivos (casos em que as íris foram estimadas no lugar errado). Com as imagens geradas diretamente pela webcam a identificação das íris ocorreu com sucesso em 87,50% dos casos, erros em 11,11 % e 1,39% de falsos positivos, o tempo médio entre o posicionamento e um clique é de cerca de 20 segundos.

Palavras-chave: Detecção e rastreamento de íris, interface homem-computador, problemas motores.

ABSTRACT

This paper presents a technique to iris detection and tracking that can be used in a human computer interface which allows people with mobility restricted, including no mobility above the shoulders, can control the mouse pointer only moving their eyes, without using expensive equipments. The unique data input used is an ordinary webcam without optical zoom, special lights or restricting user face mobility. The mouse displacement will occur in a straight way, in other words, the mouse cursor will be positioned at the place estimated by the technique. To the iris detection tests 60 images were used. 90.83% of the iris were identified correctly, there were 4.17 % of missing iris and 5% false positives (iris were estimated in a wrong place). Using images generated straight from the webcam the iris were found correctly in 87,5%, no iris were found in 11,11% and in 1,39% the technique found iris in wrong places, the average time between positioning and a click is about 20 seconds.

Keywords: Iris detection and tracking, human-computer interface, mobile restrictions.

SUMÁRIO

1. INTRODUÇÃO.....	8
2. FUNDAMENTOS TEÓRICOS E TRABALHOS RELACIONADOS.....	13
2.1. MODELOS FLEXÍVEIS – ACM/ASM/AAM.....	13
2.2. TÉCNICA DE PAUL VIOLA E MICHAEL J. JONES.....	18
2.3. DETECÇÃO DE OLHOS.....	24
3. MÉTODO PROPOSTO	29
3.1 IMPLEMENTAÇÃO.....	30
4. EXPERIMENTOS	37
4.1. POSICIONAMENTO DO CURSOR DO MOUSE	45
4. MELHORIAS E TRABALHOS FUTUROS.....	48
5. CONCLUSÃO	50

1. INTRODUÇÃO

A detecção e rastreamento de objetos é tarefa primordial em sistemas de visão computacional e, geralmente, é a primeira tarefa a ser executada neste tipo de sistema [1]. Devido à grande diversidade com que os objetos se apresentam em uma imagem, ainda hoje, este tipo de problema não possui uma solução definitiva para todas as situações possíveis. As variações a serem consideradas vão desde o tamanho e formato, passando pela cor, posição (translação e rotação nos três eixos), além de diferenças que não dependem do objeto, como condições de iluminação e do próprio equipamento utilizado para capturar a imagem (definição, taxa de amostragem, etc.). Quanto mais abrangente o algoritmo procura ser, maiores são as dificuldades encontradas. É fácil perceber que encontrar um quadrado de cor amarela em uma imagem em duas dimensões é mais simples do que encontrar um cubo de uma cor qualquer com qualquer rotação imaginável em um ambiente 3D. Objetos flexíveis apresentam uma gama de dificuldades ainda maior quando comparados a objetos rígidos, justamente por acrescentarem uma variação de formato, a identificação de características e formas humanas fazem parte desta última categoria de algoritmos.

Este trabalho apresenta uma técnica de detecção e rastreamento de íris que visa ser utilizada para criar uma interface homem-computador que permita pessoas com mobilidade restrita à face, inclusive sem mobilidade acima dos ombros, possam controlar o cursor do mouse de um computador com o movimento dos olhos.

Esse sistema pode auxiliar pessoas com diversos tipos de lesão. Alguns tipos de lesão apenas privam as pessoas de movimento enquanto que outras causam transtornos cognitivos. No primeiro caso, somente a utilização de uma interface que permite a movimentação do ponteiro do mouse será suficiente para a utilização de um computador. No segundo caso, o computador é utilizado também como uma ferramenta para auxiliar no reaprendizado de conceitos básicos como verbos, pessoas, lugares e objetos. Nesses, muitas vezes é necessário o acompanhamento especializado, bem como a utilização de aplicativos específicos para um único indivíduo, a Figura 1 apresenta um exemplo de um desses aplicativos.



Figura.1 Exemplo de aplicativo para reaprendizagem de conceitos básicos

A única entrada de dados utilizada na técnica aqui apresentada é uma “webcam” simples sem auxílio de zoom ótico, iluminação especial ou fixação da face do usuário. A detecção da íris possui ainda outras aplicações possíveis, como em sistemas de biometria,

ou até mesmo identificar sonolência de um motorista a partir do número de “piscadas” em um período de tempo.

A etapa inicial de identificação de face é realizada de forma automática pelo sistema desenvolvido por Cândido [2]. Em seu projeto Cândido desenvolveu um algoritmo de busca, detecção e rastreamento de faces de pessoas que estejam em frente ao monitor de um computador. Após a etapa de detecção, o algoritmo permanecerá rastreando e corrigindo a localização da face, até o momento que a pessoa saia do campo de visão da câmera, ou então o sistema não consiga mais detectar a face.

Há outros sistemas de “mouse óptico” com objetivo similar, inclusive com aplicações comerciais. Os sistemas desenvolvidos pela empresa Tobii utilizam uma luz infravermelha que reflete sobre as pupilas do usuário [3]. O sistema utiliza equipamentos com monitor, câmera e emissor de luz infravermelha integrados e possui mais de 5 anos de desenvolvimento. A Figura 2 mostra o modelo Tobii C12.



Figura.2 Modelo Tobii C12 com câmera, software e iluminação infravermelha incorporados.

Uma das possíveis aplicações deste equipamento é mapear, as regiões de uma página em um determinado site que mais chamam a atenção dos usuários, verificando onde o internauta está olhando no site, para adequar a distribuição de avisos publicitários.

O sistema desenvolvido por Takashiro Ishikawa para a Denso Corporation [4] utiliza técnicas de AAM (Active Appearance Models) [5] para identificar a face e a íris, verificando a direção um motorista está olhando. O sistema requer uma trabalho inicial onde são marcados manualmente pontos de interesse na imagem da face do usuário para gerar o modelo a ser utilizado, além de um processamento prévio em que são armazenadas diversas matrizes de erro entre a imagem e o modelo. Estas matrizes são utilizadas para direcionar a busca entre o modelo criado e a imagem. A Figura 3 ilustra o sistema em funcionamento. A “malha” em verde indica a identificação dos pontos previamente mapeados. Em vermelho tem-se a direção estimada dos olhos.

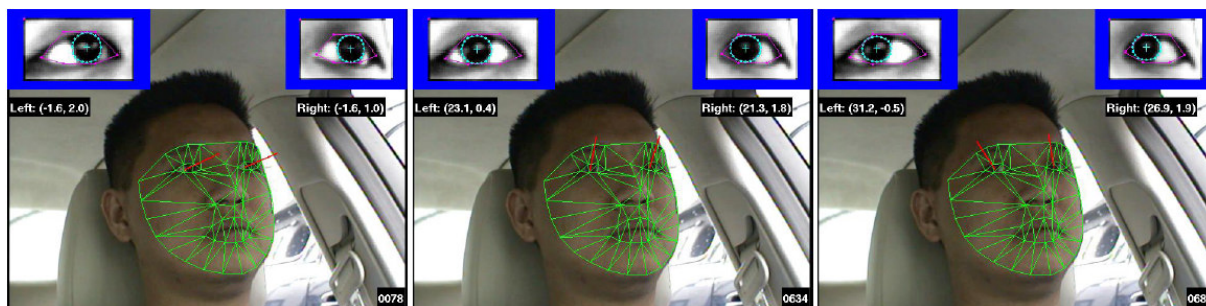


Figura 3: Sistema desenvolvido por Takashiro Ishikawa em três instantes de funcionamento [4].

Geralmente um processo de calibração mais simples implica em uma maior restrição na mobilidade do usuário, como no sistema desenvolvido por Dongheng Li e Ruqin Zhang [6]. Neste caso, é utilizado um suporte para apoio da cabeça e uma câmera de baixo



custo com zoom ótico para focar apenas a região das íris. A Figura 4 ilustra o suporte e o sistema sendo utilizado por um usuário.



Figura 4: Equipamentos utilizados no sistema apresentado em [6]. À esquerda suporte para apoio do rosto e à direita o sistema em funcionamento.

O sistema proposto neste trabalho se diferencia de outros descritos na literatura por utilizar como entrada apenas uma Webcam de baixo custo, baixa definição (640x480) com baixa velocidade de aquisição (15 FPS), além de não utilizar recurso algum para restringir os movimentos do usuário ou qualquer tipo de ampliação da imagem. Abaixo, na Tabela 1, observa-se um comparativo entre dois produtos comerciais e o sistema aqui desenvolvido.

Tabela 1 comparativo com produtos comerciais.

Modelo	Hardware	Preço	Usabilidade	Fabricante	Precisão	Imagem
Tobii CEye	Monitor, cameras, e luzes infravermelhas integradas	\$ 9.181,35	Única calibração para início de utilização. 50-70 cm	Tobii	0,5 cm	
DynaVox EyeMax	Monitor, cameras, e luzes infravermelhas integradas	\$ 13.115,32	Única calibração para início de utilização. 43-71 cm	Dynavox Technology	0,5 cm	
-	WebCam	\$ 15,00	Única calibração para início de utilização. 45-68 cm	-	8,0 cm	

Observa-se que a precisão alcançada pelos equipamentos comerciais é 16 vezes melhor que o apresentado pelo sistema no estágio atual. Entretanto considerando-se o preço e o aplicativo final a ser utilizado, que por vezes não necessita de grande precisão, o sistema proposto neste trabalho pode representar uma alternativa viável.

2. FUNDAMENTOS TEÓRICOS E TRABALHOS RELACIONADOS

Nesse capítulo são apresentadas as bases teóricas dos processos e métodos estudados, bem como as técnicas utilizadas no desenvolvimento deste trabalho.

2.1. MODELOS FLEXÍVEIS – ACM/ASM/AAM

ACM (active contour models/Snakes), ASM (active shape models/Smart Snakes) e AAM (active appearance models) [5], são técnicas baseadas em modelos flexíveis, o que quer dizer que não utilizam apenas “um modelo” aplicando apenas transformadas “simples” do tipo rotação, translação e variação do tamanho. As três técnicas têm em comum a possibilidade criar novos modelos a partir

de conhecimento prévio do objeto a ser identificado. De maneira simplificada um modelo do tipo ACM visa deformar uma curva (representada por um conjunto de pontos) de maneira a se “encaixar” às bordas de um objeto, este processo procura sempre minimizar o que os autores conceituam como “energia”. Esta energia é representada por duas medidas diferentes, uma denominada energia “interna” que é tanto menor quanto menor for a inclinação da curva e quanto mais suave forem seus contornos.

A Figura 5 ilustra a definição de energia interna, na parte superior observa-se a variação na inclinação da dos pontos que compõem o modelo, na parte inferior observa-se a variação na curvatura do modelo.

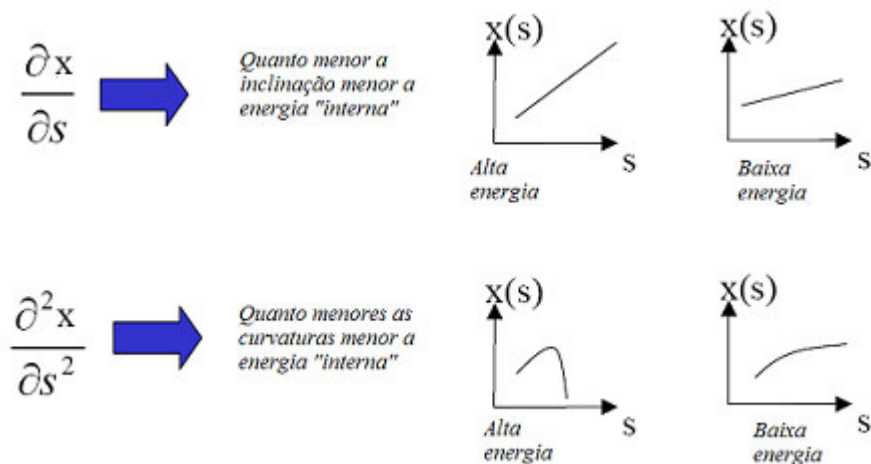


Figura 5: Representação da Energia Interna [5], a energia interna é tão menor quanto menos inclinados sejam os contornos e mais suaves as curvaturas.

O outro tipo de energia que se procura minimizar é chamado de energia “externa”, que é definida por quão bem os contornos se encaixam nas bordas encontradas na imagem. Assim, pontos que se situam sobre regiões com grande variação de intensidade possuem

menos energia que outros em regiões com pouca mudança de tonalidade, a Figura 6 ilustra esta definição.

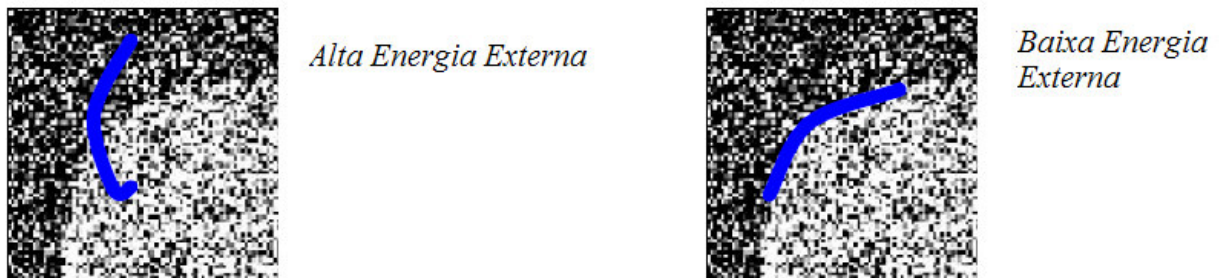


Figura 6: Representação da Energia Externa apresentada em [5].

O segundo dos algoritmos citados, ASM (Active Shape Models) é bastante similar ao modelo de “snakes” descrito em [5], que é um método clássico de rastreamento de contornos flexíveis. O ASM se diferencia por utilizar conhecimento prévio da variabilidade do objeto a ser identificado.

O primeiro passo é criar um modelo do objeto a ser procurado na imagem de entrada. Estes modelos são constituídos de uma seqüência de pontos, que representam o objeto a ser rastreado. Pontos de extremidades ou que possuam alta curvatura são preferencialmente utilizados por “representarem” melhor o objeto.

Na Figura 7 vemos um modelo simplificado de face que utiliza 15 pontos.

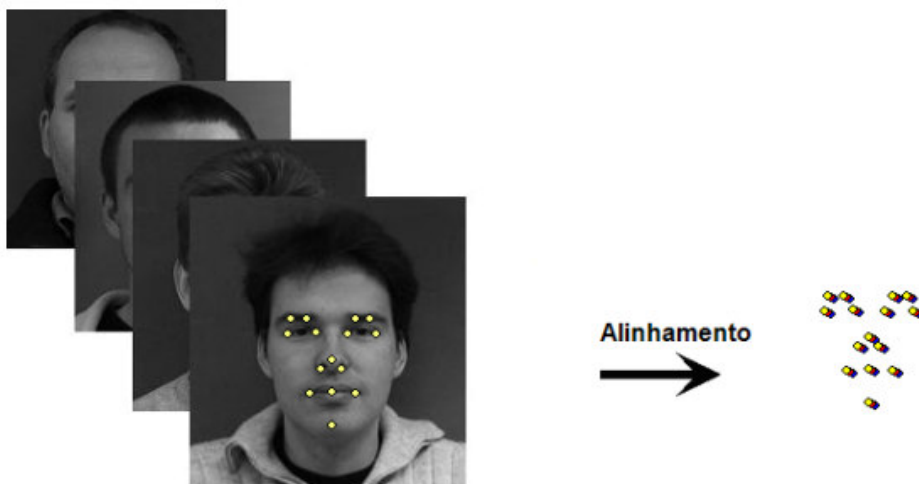


Figura 7: Modelagem de Pontos da Face para um modelo do tipo ASM.

A técnica prevê que os pontos sejam marcados em várias imagens previamente formatadas para terem tamanho e orientação similares. A partir deste mapeamento novos modelos podem ser criados a partir da relação mostrada na equação I.

$$X = \overline{X} + P_s B_s \quad (\text{I})$$

Onde X , o novo modelo gerado, é um vetor de pontos cartesianos x e y [$x_1 y_1 x_2 y_2 \dots x_n y_n$], \overline{X} é a média dos modelos utilizados, P_s é o conjunto de modelos utilizados para gerar o novo modelo, e B_s é o conjunto de pesos utilizados para gerar este novo modelo.

A definição de P_s é por si só uma tarefa complexa. Neste caso é utilizada uma técnica de análise de componentes principais que é definida pela extração dos auto vetores da matriz de covariância dos modelos utilizados. De forma simplificada, a análise de componentes principais permite que sejam identificados os eixos em que há uma maior variância na amostra utilizada, uma vez identificados estes eixos, pode-se descartar os eixos com menor variância, quanto maior o número de eixos utilizados mais próximo o modelo será do universo inicial. A definição do número de eixos utilizados é determinada pelo erro que é considerado “aceitável” para a aplicação.

Uma vez definido como gerar os modelos variáveis, a próxima etapa é encontrar o melhor modelo que se ajuste à imagem. O procedimento a seguir descreve os passos necessários:

1. Estimar um ponto de partida. Uma boa estimativa é essencial para o funcionamento do algoritmo, normalmente são utilizadas características de baixo custo computacional para esta tarefa, como

a identificação de uma cor predominante ou o conhecimento prévio da região mais provável para se encontrar o objeto.

2. Examinar a região em torno de cada um dos pontos do modelo para encontrar o ponto que melhor se adequa às condições previstas.

3. A partir dos novos pontos encontrados definir o melhor modelo que se adequa ao novo conjunto de pontos (este passo inclui encontrar os parâmetros \mathbf{P}_s) e fazer os ajustes necessários ao modelo, como rotação ou diferenças de tamanho que tenham sido geradas pelo passo 2.

4. Repetir os passos 2 e 3 até a convergência.

O terceiro e último algoritmo citado, AAM, se diferencia de ASM por utilizar modelos que considerem informações não apenas de formato e bordas, mas também da textura do interior do objeto. De forma semelhante ao ASM novos modelos são gerados a partir de padrões extraídos de uma amostra de testes. As relações utilizadas para se gerar os novos modelos podem ser sintetizadas pelas equações II e III.

$$X = \bar{X} + Q_s C \quad \text{(II)}$$

$$G = \bar{G} + Q_g C \quad \text{(III)}$$

Onde \bar{X} é a média dos formatos, \bar{G} é a média das texturas, Q_s e Q_g são matrizes com modos de variação derivados do grupo de treinamento, e C é um conjunto de parâmetros que se aplica sobre Q_s e Q_g , de maneira a gerar um novo modelo e, dependendo da implementação, C pode representar inclusive parâmetros como translação e rotação dos modelos.

Encontrar os valores que compõem C é uma tarefa complexa, trata-se de um problema de busca em um universo N dimensional. A solução utilizada é considerar que a partir do erro atual podem-se estimar novos valores que compõem o vetor C de maneira a minimizar o erro e executar esta tarefa iterativamente até o erro se tornar aceitável. O erro é computado pela diferença entre a imagem e o modelo gerado conforme a equação IV.

$$\partial I = I_i - I_m \quad (\text{IV})$$

Onde I_i e I_m são vetores com os valores de tons de cinza da imagem e do modelo gerado respectivamente. A partir de ∂I se estima a variação dos parâmetros em C para diminuir o erro, conforme indicado na equação V.

$$\partial c = A \partial I \quad (\text{V})$$

Para encontrar os valores que compõem o vetor A é feita uma regressão linear com múltiplas variáveis, o conjunto de pontos utilizado para este procedimento é gerado a partir de pequenos e conhecidos desvios em ∂c nas amostras de teste.

2.2.TÉCNICA DE PAUL VIOLA E MICHAEL J. JONES

Viola e Jones [7] introduziram uma nova maneira de procurar por “regiões de interesse” em imagens. A técnica foi inicialmente aplicada na detecção de faces, mas é aplicável à identificação de objetos em geral, desde que não existam muitas variações rotacionais. Os autores afirmam que o desempenho do algoritmo é superior aos demais dedicados a identificação de faces quando

comparado isoladamente. Os autores afirmam ainda que outros sistemas conseguem taxas de convergência similares, mas, para isto, utilizam uma combinação de algoritmos ou de outras informações, como diferenças entre frames ou cor, enquanto que o algoritmo proposto por Viola e Jones se baseia apenas em imagens em tons de cinza.

Originalmente, a técnica foi aplicada em imagens de 384x288 pixels, as faces são encontradas em vídeos a uma taxa de 15 frames por segundo em um computador com processador Pentium III de 700 MHz. A solução proposta se baseia na busca de “atributos” simples e na sucessiva busca por “atributos” mais detalhados. A decisão de se efetuar novas buscas é feita por uma variante do algoritmo AdaBoost introduzido por Freund e Schapire [8].

O modelo proposto por Viola e Jones utiliza apenas 4 atributos simples, estes são calculados com base na soma da intensidade dos pixels em regiões retangulares. A Figura 8 mostra os atributos utilizados. O valor de um atributo é dado pela soma dos pixels da região clara, menos a soma dos pixels da região escura.

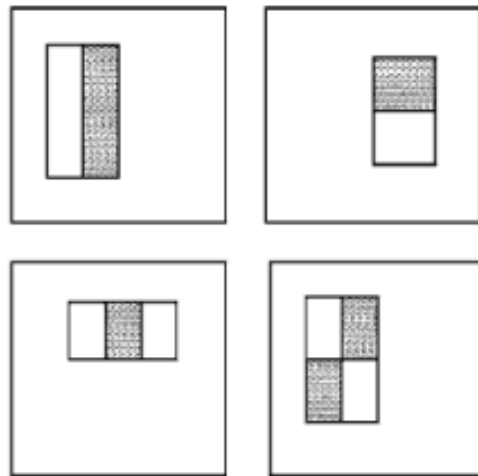


Figura 8: Representação geométrica dos “Atributos” usados no sistema de Viola e Jones

Viola e Jones introduziram uma maneira computacionalmente “barata” de calcular estes atributos, que é baseado no que denominaram “imagem integral”. O princípio da “imagem integral” é representar cada pixel pelo somatório dos pixels acima e à esquerda incluindo o pixel em questão. A equação VI apresenta este cálculo, onde $ii(x,y)$ é a imagem integral e $i(x,y)$ é a imagem original.

$$ii(x, y) = \sum_{x=0}^q \sum_{y=0}^p i(x, y) \quad (\text{VI})$$

A Figura 9 ilustra a área que é utilizada para o cálculo da imagem integral para as coordenadas x e y .

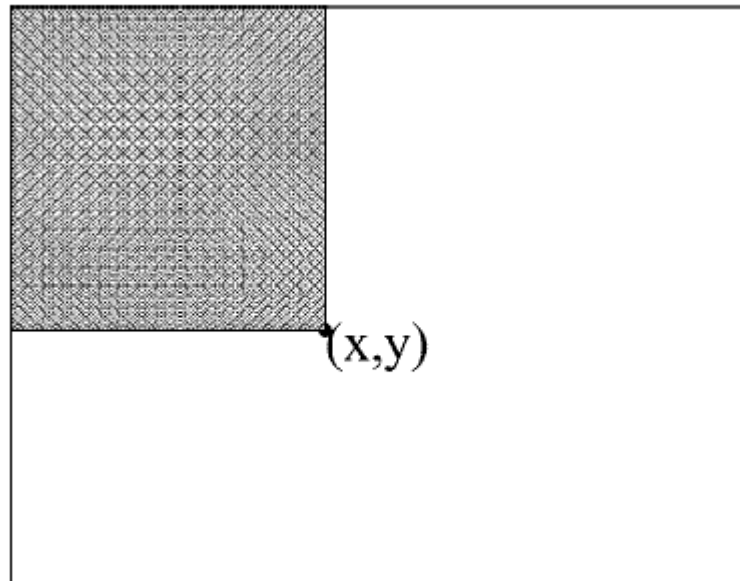


Figura 9: Representação da “imagem integral”.

O cálculo da equação VI pode ser obtido em apenas uma iteração com base no somatório do ponto imediatamente à esquerda e a soma

dos pontos na mesma coluna e acima do ponto a ser calculado, segundo as relações dadas nas equações VII e VIII.

$$s(x, y) = s(x, y-1) + i(x, y) \quad \text{(VII)}$$

$$ii(x, y) = ii(x-1, y) + s(x, y-1) \quad \text{(VIII)}$$

Onde $i(x,y)$ representa a intensidade da imagem em tons de cinza no ponto e $s(x,y)$ representa a soma dos pontos na mesma coluna e acima do ponto em questão, e $ii(x,y)$ representa o valor da “imagem integral”.

O cálculo do somatório dos pontos em uma região quadrangular é feito a partir do valor da “imagem integral” dos vértices da região. Assim para calcular o somatório dos pixels na região sombreada D da Figura 9 pode-se utilizar o valor da imagem integral nos pontos 1,2,3 e 4. O valor da imagem integral no ponto 1 é a soma dos pixels no retângulo A. O valor no ponto 2 é a soma dos retângulos A + B, na posição 3 é a soma das regiões A + C, e na posição 4 é A + B + C + D. Assim a soma na região D pode ser calculada por $ii(4) + ii(1) - ii(3) - ii(2)$ representada na Figura 10.

A utilização de atributos retangulares parece simples em relação a alternativas mais flexíveis, entretanto a velocidade com que é possível computar estas regiões parece compensar a sua simplicidade, pois possibilita a utilização de um número maior de atributos com um custo computacional viável.

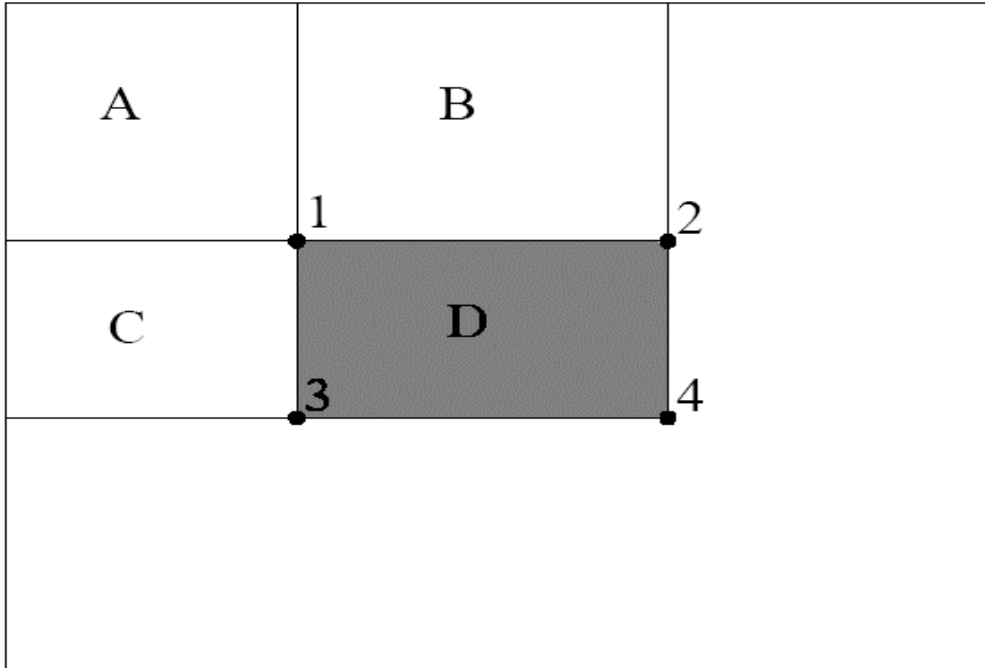


Figura 10: Cálculo do somatório de pontos de uma região a partir da Imagem integral.

A definição de quais atributos serão utilizados e quais terão um maior peso é feita por meio de uma variante do algoritmo “Adaboost”[8]. O Adaboost foi introduzido em 1996 e desde então ganhou grande popularidade, baseia-se em encontrar uma série de classificadores simples, que mesmo possuindo uma taxa de acerto medíocre (um classificador que possua uma taxa de acerto de 51% é considerado medíocre mas pode ser utilizado neste tipo de classificador) e associá-los de maneira a obter um acerto superior, o grande mérito do algoritmo é encontrar o peso utilizado para cada um dos classificadores.

A abordagem utilizada por Viola e Jones[7], se baseia em não calcular todos os classificadores simultaneamente, em vez disto são utilizados sucessivos refinamentos. Cada um dos classificadores utilizado pode descartar a região de interesse, caso a região não seja descartada é utilizado outro classificador mais detalhado, e por

consequência mais custoso. A Figura 11 ilustra o primeiro classificador utilizado que é formado por 2 atributos apenas.

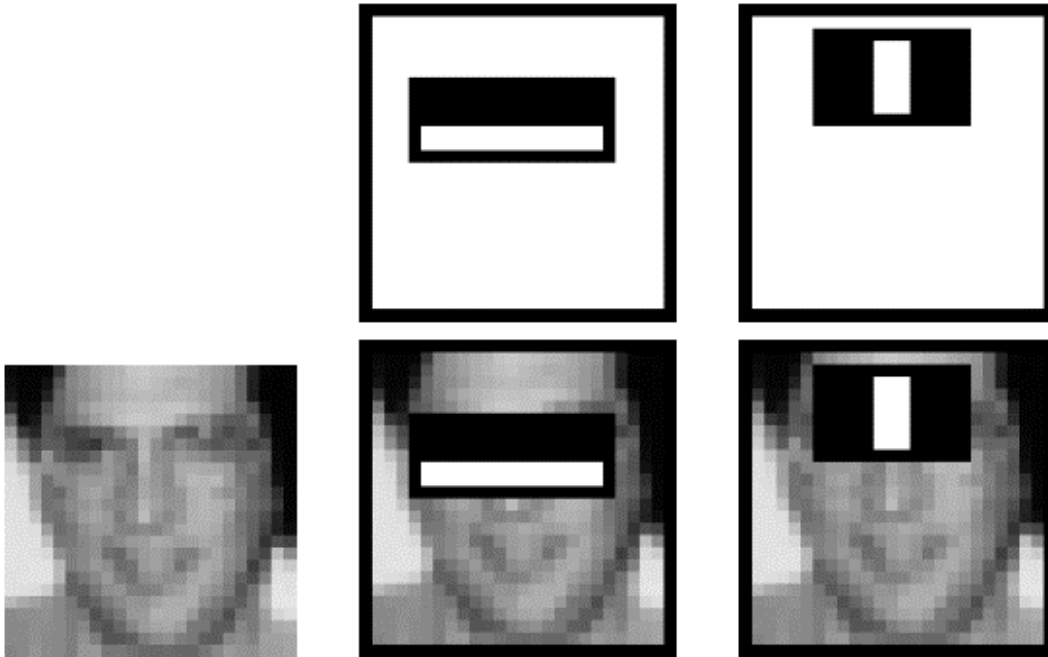


Figura 11: Representação dos dois primeiros atributos utilizados em Viola e Jones.

A Figura 12 ilustra como os classificadores são organizados de maneira a descartar a região de interesse ou detalhar mais a busca.

O algoritmo proposto por Viola e Jones[7] trata regiões de, no mínimo, 24x24 pixels, a busca é feita em 11 escalas diferentes, cada uma 25% maior que a anterior para compensar variações de escala nos objetos.

A estrutura final do detector é composta por 38 classificadores em cascata que usam no total 6060 atributos. O primeiro classificador usa apenas dois atributos e é capaz de rejeitar 50% das imagens não-face e aceitar perto de 100% das imagens face. O classificador seguinte usa dez atributos. Os próximos dois classificadores usam 25 atributos cada, e são seguidos por três classificadores com 50 atributos. Na seqüência, os classificadores usam quantidades variadas de atributos.

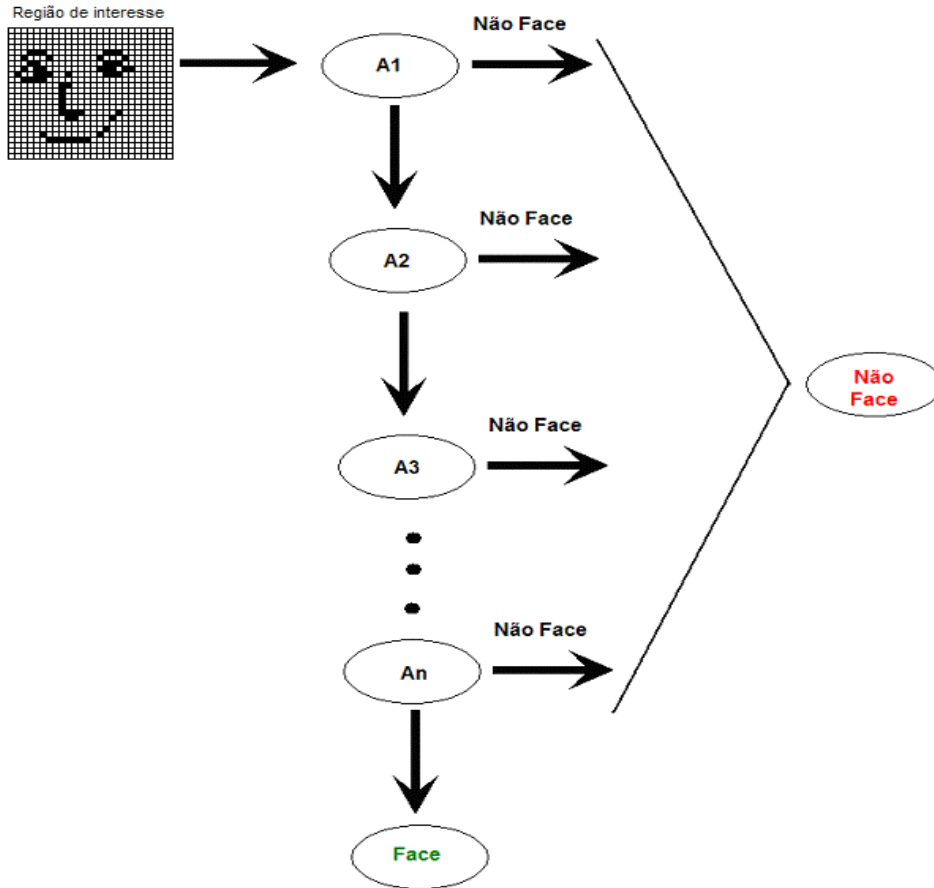


Figura 12: Representação da sequência de busca dos atributos da técnica baseada em Adaboost proposta por Viola e Jones.

2.3. DETECÇÃO DE OLHOS

Nos trabalhos realizados por Peng [9] e Dongheng [10], a identificação dos olhos parte da região da face já delimitada ou impõe uma restrição ao movimento dos usuários. Esta restrição pode ser mecânica, por meio de um apoio para a face, ou apontando a câmera para a região onde as íris deveriam estar. Note que se o usuário se movimentar para fora desta região o método não funcionará.

Na abordagem feita por Peng [9], uma vez delimitada a região da face, a partir do gradiente da imagem em tons de cinza consegue-

se restringir a região de busca dos olhos de maneira bastante efetiva, a Figura 13 ilustra este procedimento.

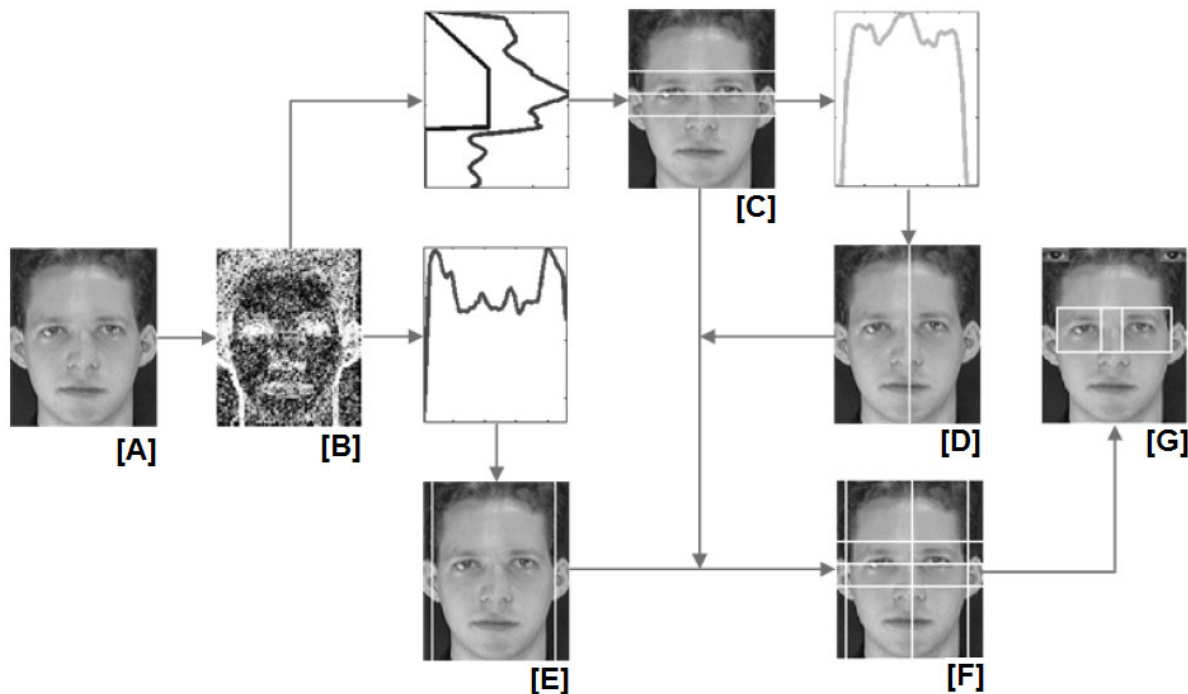


Figura 13: Esquema utilizado em [9] para identificação da região dos íris

O primeiro passo é calcular o gradiente da imagem (mostrado na Figura 13 [B]). Em seguida, aplica-se uma projeção horizontal para este gradiente. Sabendo-se que a região das íris possui grande variação de tons e bordas comparando com as outras partes da face, é de se esperar que o pico da projeção horizontal na parte superior identifique a reta horizontal que passa pelas íris (mostrado na Figura 13 [C]). Então é efetuada uma projeção vertical na região delimitada em desta imagem. O pico encontrado próximo à região central da imagem é definido como o centro da face (Figura 13 [D]). Ao mesmo tempo uma projeção vertical é feita para identificar as laterais da face (Figura 13 [E]). Combinando os resultados de apresentados na Figura 13 [C], [D] e [E] a imagem é segmentada como em mostrado na Figura 13 [F]. Finalmente, com base nesta

segmentação e na largura da face, delimita-se a região dos olhos, como apresentado na Figura 13 [G].

Ainda em [9], de posse de um modelo do olho, procura-se encaixar este modelo à imagem inicial. Para tal, utiliza-se o procedimento de reconhecimento de padrão, é utilizado ainda o conhecimento prévio do tamanho esperado para os olhos, restringindo assim as variações em escala a serem feitas no modelo.

Na Figura 14, observa-se no topo o modelo utilizado, no centro a imagem onde será efetuada a busca e na base da figura a diferença entre o modelo e a imagem original.



Figura 14: O modelo utilizado em [9], no topo tem-se o modelo em si, no meio a imagem do usuário e embaixo a diferença entre a imagem e o modelo.

A avaliação da qualidade do ajuste do modelo sobre a imagem é feita a partir das relações apresentadas nas equações IX e X:

$$C_{f,g}[i,j] = \sum_{k=1}^m \sum_{l=1}^n g[k,l]f[k+i,l+j] \quad (\text{IX})$$

$$M[i,j] = \frac{C_{f,g}[i,j]}{\left\{ \sum_{k=1}^m \sum_{l=1}^n f^2[k+i,l+j] \right\}^{1/2}} \quad (\text{X})$$

A equação IX representa a correlação entre o modelo e a imagem considerando a imagem a partir do ponto $[k,l]$, onde $g[k,l]$ representa a intensidade da imagem no ponto $[k,l]$ e $f[i,j]$ a intensidade do ponto $[i,j]$ no modelo. A equação X representa uma normalização para a região da figura que está sendo avaliada.

Em [10], a partir de um ponto no centro da íris, já identificado anteriormente, inicia-se uma busca radial de pontos de borda. A partir desses pontos, utiliza-se o algoritmo RANSAC [10], que é baseado em selecionar de maneira aleatória um grupo de pontos, verificando se a sequência aproxima-se do formato esperado, neste caso o formato de uma elipse. São feitas restrições quanto ao tamanho e excentricidade esperados, permitindo assim o descarte de grupos de pontos inadequados. A Figura 15 ilustra o procedimento.

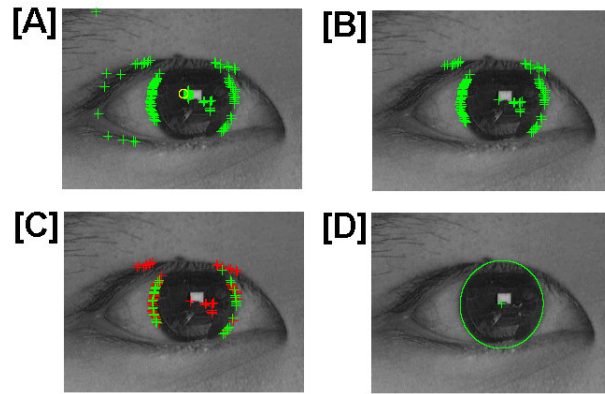


Figura 14 - Procedimento proposto em [10]

Na Figura 15[A] identifica-se o conjunto de todos os pontos encontrados (em verde). Em 15[B] os pontos restantes após a exclusão dos pontos distantes da média, 15[C] Pontos selecionados (em verde) e excluídos (em vermelho), e finalmente em 15[D] a melhor elipse.

3. MÉTODO PROPOSTO

O método proposto neste trabalho assume que a região da face do usuário já foi previamente encontrada. A partir de uma série de projeções do gradiente da imagem, identifica-se a região das íris, cria-se então uma malha de busca com passo inferior ao raio mínimo esperado (esta medida é inferida a partir da largura da face). Por fim, faz-se uma busca radial com vizinhança de 8 por pontos de borda da íris em cada um dos pontos da malha. Caso a busca radial retorne mais de 3 pontos de borda, identifica-se a circunferência que melhor se adequa a estes pontos, se o ajuste possuir um erro inferior ao limite então é feito um refinamento em que se buscam mais pontos de borda próximos à circunferência estimada. Este processo pode identificar mais do que uma região candidata à íris. É utilizada então a diferença de intensidade entre os pontos internos e externos à circunferência para definir qual das circunferências será considerada como a íris.

O sistema se caracteriza por duas etapas distintas, uma etapa inicial de calibração e outra etapa responsável pelo posicionamento do mouse e simulação do clique, a figura 16 ilustra o processo.

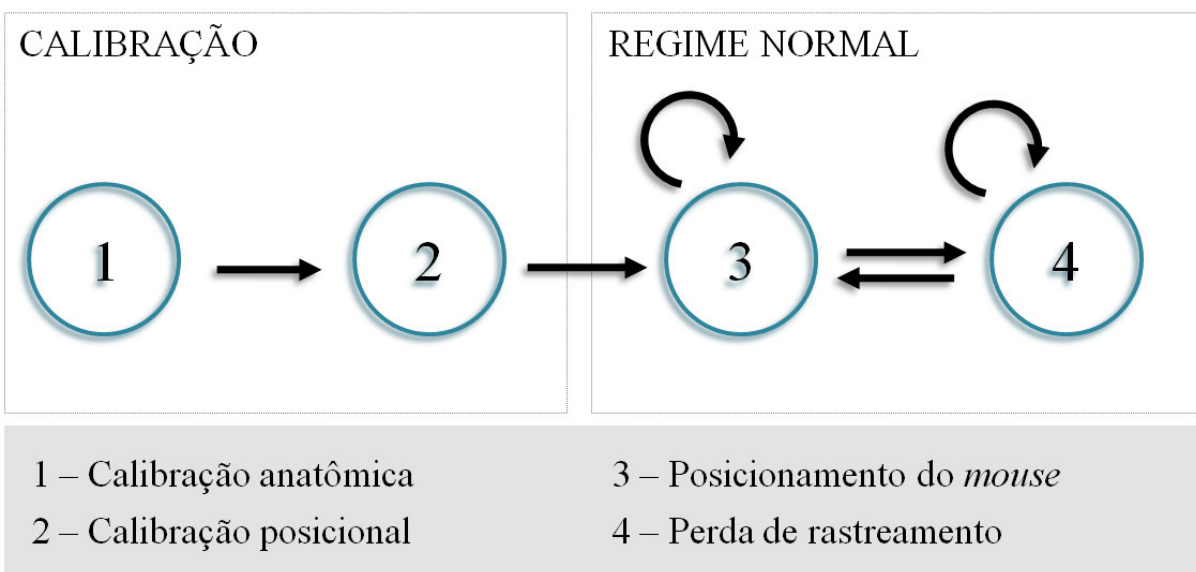


Figura 16 Etapas utilizadas pelo sistema.

No início da execução, etapa 1, o sistema se ajusta ao tamanho das íris do usuário, e à diferença de intensidade entre a íris e a esclera, esta calibração inicial restringe o universo de busca e favorece o algoritmo. Na etapa 2 o usuário é orientado a direcionar os olhos para as bordas da tela (cima, baixo, esquerda e direita), a partir destas informações, é criado um mapeamento da posição das íris e do posicionamento do ponteiro do mouse. A partir deste momento o sistema entra em seu regime normal, posicionando o mouse de acordo com a região determinada pela posição da íris do usuário, esta etapa é representada pela etapa 3 da figura 16. No momento em que as íris deixem de ser encontradas considera-se que os olhos estão fechados e inicia-se uma contagem de 8 ciclos de busca, um silvo é emitido para alertar o usuário que o sistema está interpretando a situação como se o olho estivesse fechado, representado pela etapa 4. Se a íris voltar a ser identificada por, pelo menos 8 ciclos o sistema emula um clique, e retorna para a etapa 3.

3.1 IMPLEMENTAÇÃO

Utilizando como entrada apenas uma “webcam” simples, o sistema aqui apresentado visa encontrar a íris do usuário e transferir o seu movimento para o ponteiro do mouse e diferente dos trabalhos observados em [6,9 e 10], sem a utilização de outros recursos como zoom, iluminação infravermelha ou fixação da face por um suporte.

Inicialmente, aplica-se um filtro do tipo gradiente sobre a imagem colorida. O gradiente da imagem ressalta regiões de grandes variações de cor e borda, esta imagem será utilizada para identificar a região dos olhos através de um procedimento semelhante ao

utilizado em [9], diferindo apenas na etapa extra [C], conforme mostrado na Figura 17.

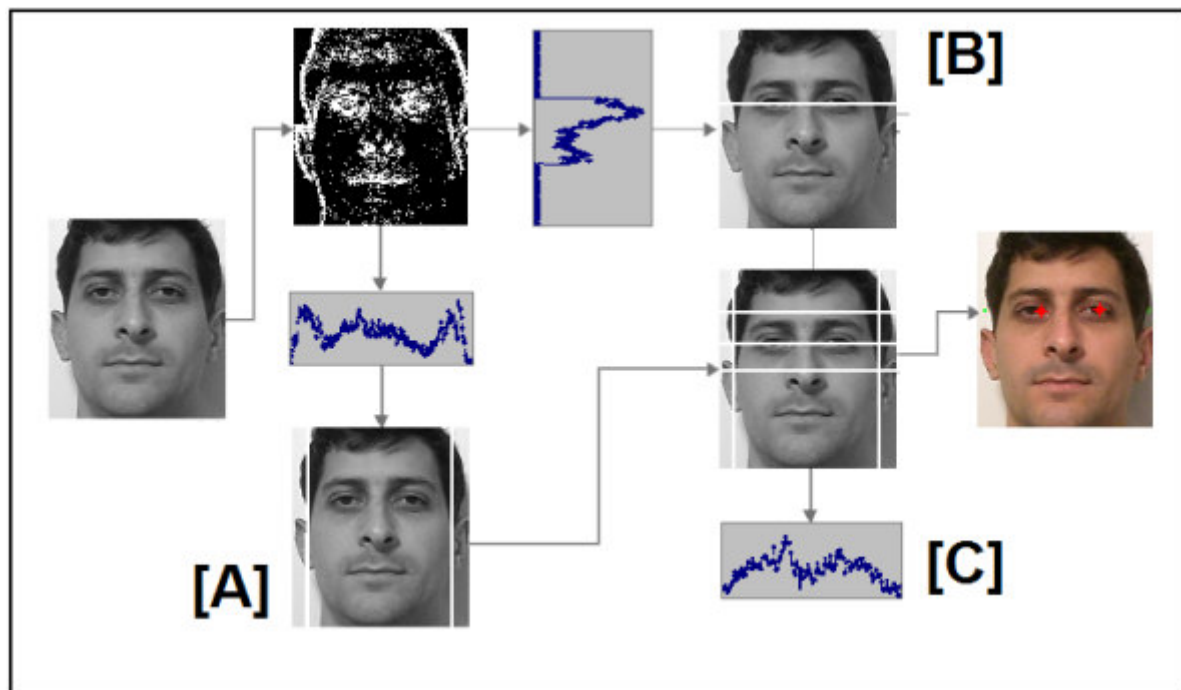


Figura 17 Procedimento utilizado na identificação da região dos olhos [10]

A Figura 17 ilustra o processo, na Figura 17[A], a partir da projeção horizontal do gradiente da imagem encontram-se dois pontos de máximo, um em cada metade da imagem. Estes pontos identificam as retas verticais que delimitam as laterais da face. Na Figura 17[B], no terço médio da face, limitado pelas retas verticais laterais, identifica-se o ponto de máximo da projeção vertical da região. Este ponto determina a reta horizontal que passa pelas íris. E, finalmente na Figura 17[C], na mesma região identificada no passo anterior, encontram-se dois pontos de máximo nas projeções horizontais, uma em cada metade da face. Estes pontos, juntamente com a reta horizontal que passa pelas íris, determinam os dois pontos onde se iniciam as buscas. Note que estes pontos estão próximos das íris do indivíduo.

A Figura 18 mostra exemplos dos pontos identificados pelo processo utilizando imagens de faces retiradas da IMM Face Database [11].



Figura 18 Exemplos de imagens com pontos iniciais de busca.

A busca da íris é feita a partir de pontos em uma malha de 18×6 pontos com centro no ponto identificado na etapa anterior que distam igualmente entre si de $0.017 * (\text{Largura da Face})$, esta constante foi inferida da seguinte forma: para as imagens utilizadas nos testes a menor proporção entre o diâmetro da íris e a largura da face foi de 0,0425 assim, utilizou-se ainda uma margem de 20% para o raio mínimo aceito que é de $0,034 * (\text{Largura da Face})$.

Para o passo da malha considerou-se a metade deste valor, garantindo que pelo menos um ponto da malha se inicie dentro da íris. A Figura 19 mostra a malha de busca sobreposta à imagem inicial.



Figura 19 Sobreposição da malha de busca sobre a imagem inicial.

A próxima etapa do processo é encontrar os contornos das íris do usuário. Para tanto, é realizado uma busca de pontos de bordas que sejam o mais próximo de uma circunferência dentro da região delimitada na etapa anterior. Para a identificação das bordas, aplica-se um filtro do tipo Canny [12] à imagem em tons de cinza, conforme ilustrado na Figura 18. São aplicados filtros com parâmetros distintos para cada uma das laterais da face com o objetivo de minimizar efeitos de iluminação desigual, conforme mostrado em [12] o filtro de Canny utiliza dois parâmetros que representam uma faixa de intensidades. Caso a intensidade da borda seja superior ao maior parâmetro, o ponto será considerado como sendo de borda, caso seja inferior ao menor, será desconsiderado.

Pontos com intensidade intermediária serão considerados caso participem de uma cadeia de pontos que possuam pelo menos um ponto com intensidade superior ao maior parâmetro.

A obtenção dos parâmetros do filtro de Canny é feita de forma iterativa. Inicialmente, aplica-se o filtro e identifica-se a proporção de bordas na região da malha de busca. Caso o número de pontos de borda seja inferior à faixa esperada, os parâmetros do filtro de Canny são diminuídos de forma a identificar mais pontos. Caso o número de pontos de borda seja superior à faixa aceita, os parâmetros são aumentados de maneira a minimizar a identificação de bordas.

A Figura 20 apresenta o resultado do processo de aplicação do filtro de Canny para diferentes parâmetros. Na Figura 20[A] observa-se o resultado do processo após a 1ª iteração, na Figura 20[B] após a 8ª iteração.

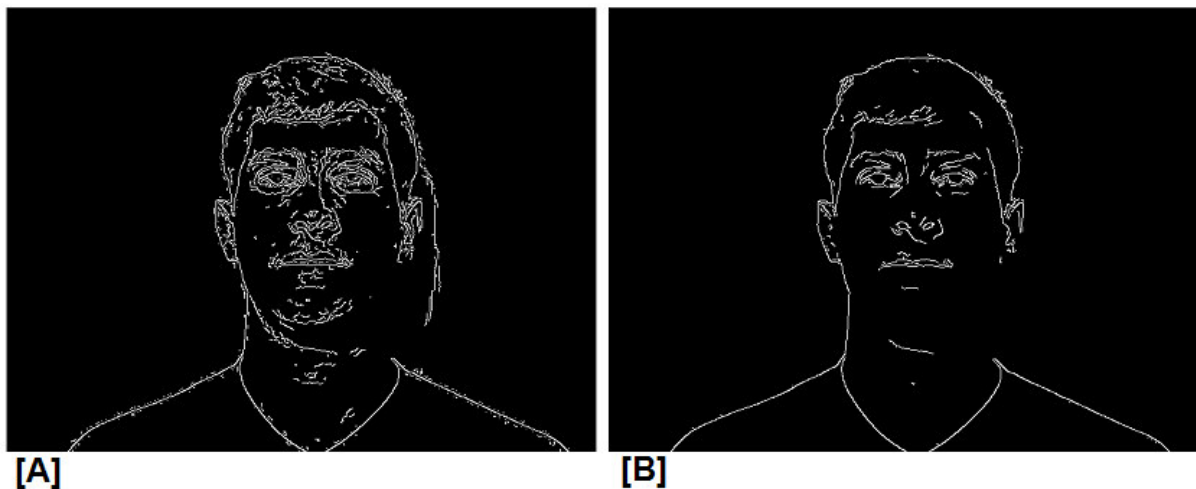


Figura 20 Resultados do filtro Canny para diferentes parâmetros em [A] 50 e 75 e em [B] 90 e 135

A partir de cada um dos pontos da malha identificada anteriormente, inicia-se uma busca radial com vizinhança 8. Caso

seja identificado um ponto de borda, faz-se uma busca local para verificar se este ponto faz parte de um conjunto maior, que possua a extensão horizontal ou vertical superior a 0.7 vezes o raio mínimo aceito, caso isto não ocorra o ponto é descartado. Este procedimento evita que se considerem pequenas manchas provenientes de reflexos ou sombras. A busca continua até que um ponto que atenda à restrição seja encontrado ou o número de tentativas seja superior a 15. Se o número de pontos encontrados for superior a 3 identifica-se o círculo que melhor se encaixe ao conjunto. Caso contrário este conjunto de pontos é descartado e a busca se reinicia a partir do próximo ponto da malha.

A identificação da melhor circunferência que se adeque aos pontos encontrados é feita conforme a técnica de Robert Israel [13]. A identificação do erro do ajuste dos pontos à circunferência é definida pela equação XI.

$$\mathcal{E} = \frac{\sum_{k=1}^m |\sqrt{(x_k - x_c)^2 + (y_k - y_c)^2} - R|}{m * R} \quad (\text{XI})$$

Onde R é o raio da circunferência encontrada, x_c e y_c são as coordenadas do centro da circunferência, x_k e y_k são as coordenadas de cada um dos pontos, e m é o número total de pontos. A equação XI representa a média ponderada das diferenças entre a distância dos pontos ao centro da circunferência estimada e o raio da circunferência estimado.

Caso o erro seja pequeno será feito um procedimento de refinamento da busca, inicialmente faz-se uma busca local partindo de cada um dos pontos originais em busca de outros pontos de borda que estejam próximos ao contorno da circunferência estimada. A

partir deste novo conjunto de pontos encontra-se a nova circunferência que melhor se adequa a este novo conjunto. Conjuntos de pontos que não se distribuam de maneira uniforme pelo contorno da circunferência serão descartados, o critério utilizado para definir esta uniformidade é o seguinte: divide-se a circunferência em 36 faixas (de 10 graus cada), quanto mais faixas possuírem pontos, melhor a distribuição dos pontos ao longo da circunferência. Conjuntos que não se distribuem por pelo menos 140 graus são descartados. A Figura 21 ilustra o processo de refinamento de busca em duas condições.

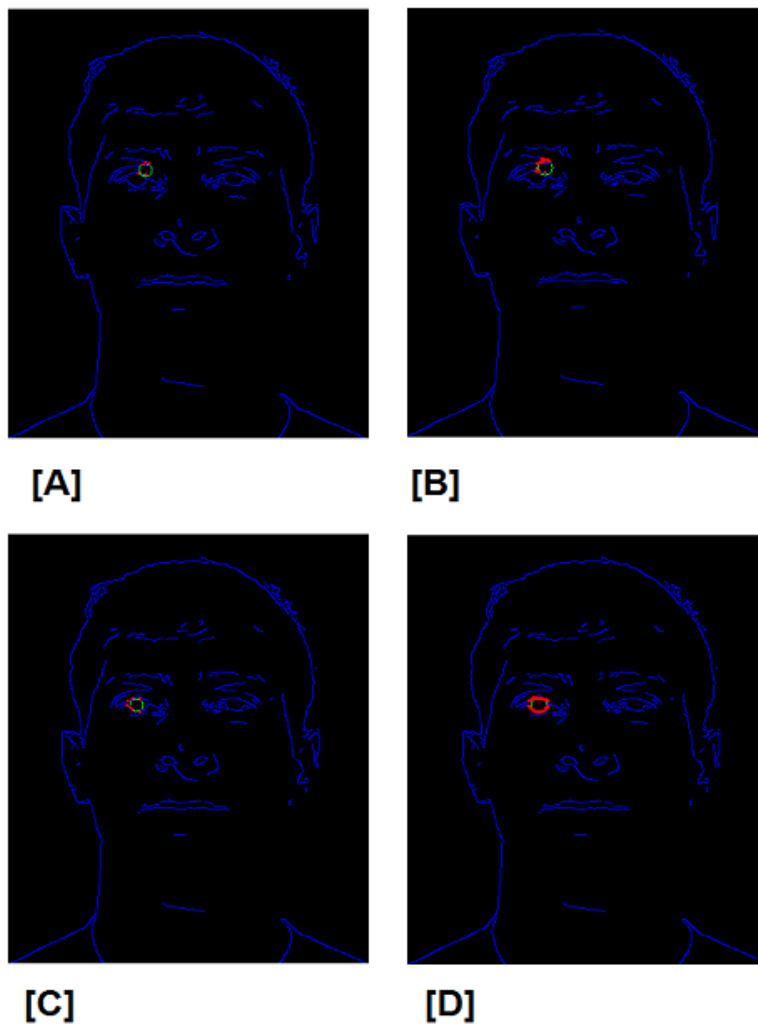


Figura 21: Refinamento de busca em duas condições

Na Figura 21 [A] tem-se a busca inicial em um ponto que atende às condições iniciais de diâmetro da circunferência, número de pontos e erro, na Figura 21[B] após o refinamento o conjunto é descartado, pois os pontos não se distribuem radialmente dentro da faixa esperada. Nas Figuras 21[C] e 21[D] observa-se um grupo em que o refinamento encontra um conjunto de pontos que atende aos requisitos mínimos do sistema. O processo inteiro é repetido para cada um dos pontos da malha. Os conjuntos de pontos que atenderam às restrições (erro mínimo, diâmetro da circunferência, distribuição de pontos) são armazenados. Dentre os conjuntos de pontos selecionados, aquele que possuir a maior diferença de intensidade entre pontos internos à circunferência estimada e imediatamente externos será considerado como a íris do usuário.

4. EXPERIMENTOS

Os experimentos foram divididos em duas partes, na primeira parte foram executados testes com imagens estáticas para verificar a capacidade do sistema na identificação de íris numa situação independente dos movimentos do usuário. No segundo grupo de testes foram utilizados usuários distintos que utilizaram uma interface simples para selecionar uma sequência de dígitos.

Para o primeiro grupo de testes foram utilizadas imagens do grupo da IMM Face Database [11]. Foram selecionadas apenas as imagens coloridas e com faces frontais totalizando 60 imagens. A intenção de utilizar uma amostra já disponível na internet é possibilitar que outros trabalhos possam ser comparados de forma simples e padronizada. A Figura 22 ilustra o resultado da busca numa amostra de 12 imagens das 60 utilizadas, em diferentes condições de iluminação e expressão facial.



Figura 22: Identificação das íris em 12 imagens da IMM Face Database[11].

A primeira e a segunda face foram selecionadas para ilustrar condições em que o algoritmo costuma falhar, no primeiro caso a pessoa está com os olhos ligeiramente fechados e no segundo caso a diferença de iluminação entre as laterais da face do indivíduo são muito acentuadas, para usuários com óculos a técnica oferece pouca ou nenhuma efetividade, de modo que os testes realizados não levaram em conta usuários com esta característica.

Para o grupo de imagens selecionadas, de um total de 120 íris possíveis, foram encontradas 109 íris (considerando cada íris existente na imagem individualmente como um acerto), em 5 casos as

íris não foram encontradas e ocorreram 6 falsos positivos. A Tabela 2 mostra os resultados obtidos no primeiro experimento:

Tabela 2 Resultados para testes com imagens estáticas.

Acertos	109	90.83%
Erros	5	4.17%
Falsos Positivos	6	5.00%
Total	120	100.00%

No segundo experimento as sequências de imagens apresentam um novo grupo de dificuldades. As imagens utilizadas anteriormente possuem ótima qualidade, as imagens obtidas diretamente da câmera, apesar de terem uma definição similar (480X640) e boas condições de iluminação, não têm a mesma qualidade que as extraídas a partir de uma câmera digital. O tempo necessário para a aquisição de uma imagem na webcam utilizada é cerca de 8 vezes maior do que uma câmera digital. Tipicamente em uma webcam, a velocidade de aquisição é de 15FPS (0,067s por imagem), enquanto que as câmeras digitais necessitam de apenas 1/125s (0,008s por imagem). Esta diferença no tempo de aquisição faz com que pequenos movimentos com a cabeça gerem imagens ligeiramente “borradas” na aquisição realizada pela webcam. Este borramento prejudica significativamente a detecção de bordas. A Figura 23 ilustra o equipamento e as condições de iluminação durante a realização dos testes e utilizou-se uma camera de marca Creative, modelo Creative Pro, um computador HP Pavilion dv8000 com 1.8 GHz, e iluminação direta proveniente de 2 lâmpadas fluorescentes de 40 Watts.

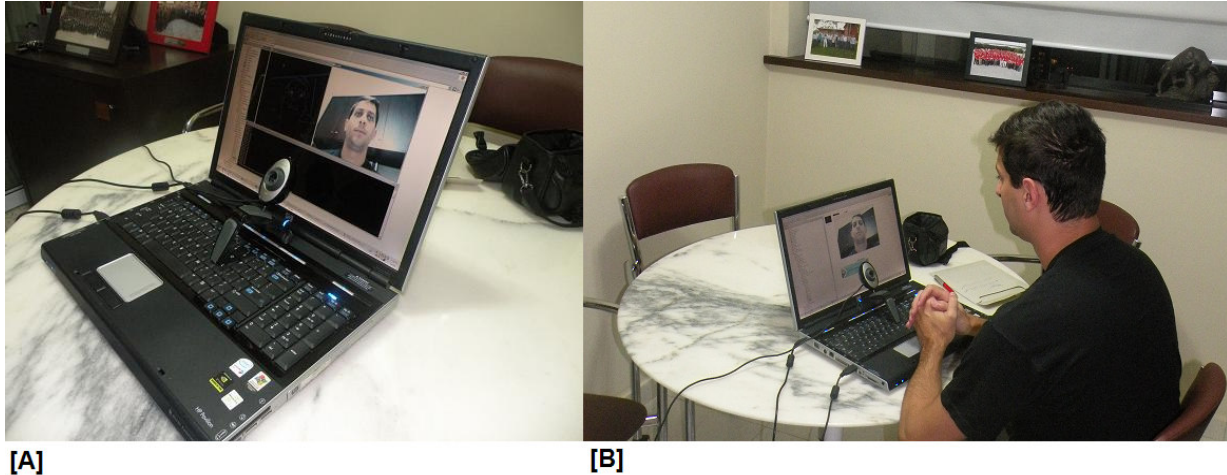


Figura 23: Em [A] visualiza-se a montagem com a webcam. Em [B] observa-se a qualidade da iluminação e a distância típica em que foram realizados os testes.

Outro fator que influencia no resultado é o posicionamento da câmera, que deve ser colocada abaixo da linha dos olhos para favorecer o algoritmo de detecção. O posicionamento da câmera em outra posição prejudica a detecção das íris. Quanto mais alta a posição da câmera, maior a sobreposição das pálpebras com relação à íris prejudicando o processo de busca por uma circunferência.

Foram realizados testes para identificar a proporção do tempo em que as íris eram encontradas em 4 pessoas com condições de iluminação favoráveis e com o posicionamento da câmera abaixo da linha dos olhos semelhante ao observado na Figura 23[B]. Foram armazenadas imagens de cada um dos usuários enquanto os mesmos direcionavam os olhos para cada uma das extremidades da tela, para as medianas das laterais e o centro da tela, repetiu-se este teste 2 vezes para cada usuário. Observou-se que, com frequência, o sistema “perdia” as íris enquanto os usuários mudavam a direção dos

olhos, porém, em seguida, as íris eram identificadas novamente. A identificação das íris ocorreu com sucesso em 87,5% dos casos, erros em 11,11% e 1,39% de falsos positivos, conforme indicado na Tabela 3.

Tabela 3 Resultados para testes com vídeo.

Acertos	63	87,50%
Erros	8	11,11%
Falsos Positivos	1	1,39%
Total	72	100,00%

Na Figura 24 observam-se algumas das imagens do sistema funcionando sobre imagens capturadas diretamente da webcam.



Figura 24: testes finais a partir de imagens extraídas diretamente da webcam.

Eventualmente, a etapa de detecção de bordas capta pequenas manchas devido a reflexos que prejudicam a identificação das íris. Na Figura 25[B] podemos verificar no detalhe os contornos identificados para o olho direito. Pode-se observar a formação de um número considerável de pontos de borda no interior da íris o que causa uma falha no sistema de detecção.

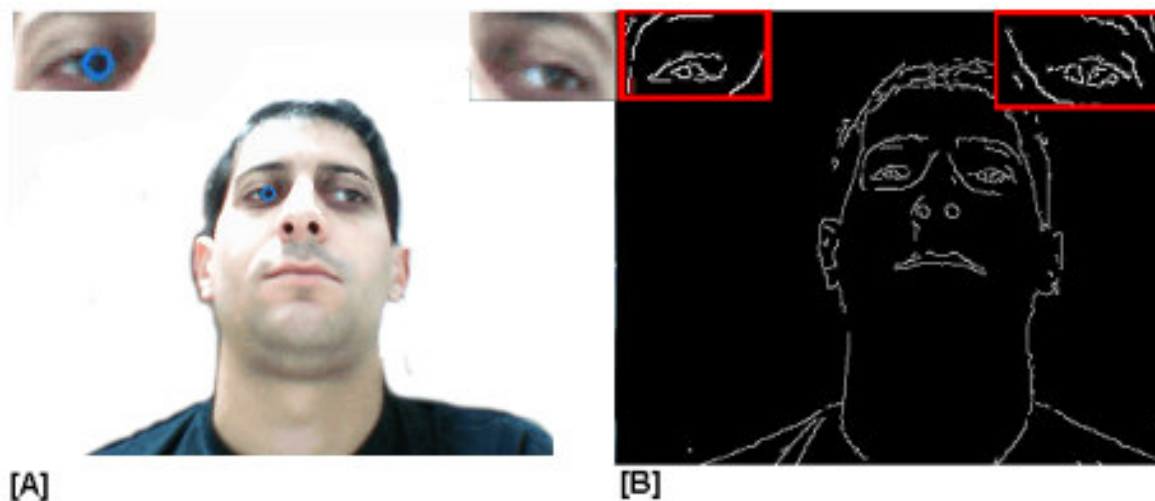


Figura 25: Em [A] observa-se uma ocorrência de erro no sistema. Em [B] visualiza-se a imagem gerada pelo detector de bordas.

Para as imagens geradas pela câmera utilizada e na posição/distância dos experimentos, a máxima amplitude do movimento das íris na direção horizontal foi por volta de 20 pixels e de 10 pixels na direção vertical. A Figura 26 ilustra a amplitude identificada. Os pontos em vermelho são determinados por uma malha de passo igual a 10 pixels.

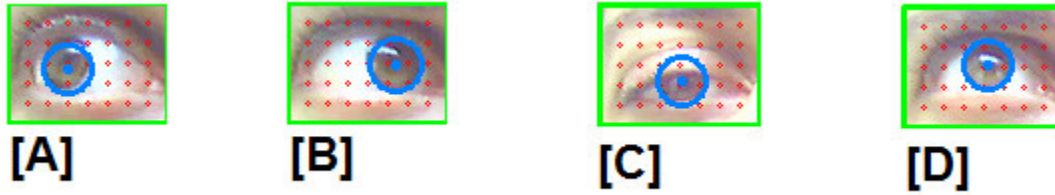


Figura 26: Em [A] observa-se a identificação da íris enquanto os olhos estão direcionados para a lateral esquerda da tela, em [B] quando estão direcionados para a direita, em [C] para a base da tela e em [D] para topo.

Observou-se ainda que, mesmo direcionando os olhos para um ponto fixo, há uma oscilação por volta 4 pixels em ambos os sentidos (cima/baixo e direita/esquerda), e que esta oscilação se mantém constante direcionando os olhos para qualquer região. Esta variação é atribuída a pequenos desvios na posição do rosto do próprio usuário e diferenças mínimas na iluminação. A Figura 27 mostra as pequenas variações na identificação da íris com o usuário direcionando os olhos para a lateral esquerda da tela.



Figura 27: Pequenas variações com o usuário direcionando os olhos para a lateral esquerda da tela.

Observa-se na Figura 28 a representação gráfica do movimento horizontal da íris apresentado na Figura 27 o eixo das abscissas indica individualmente cada uma das medidas realizadas no eixo das ordenadas a distancia em pixels.

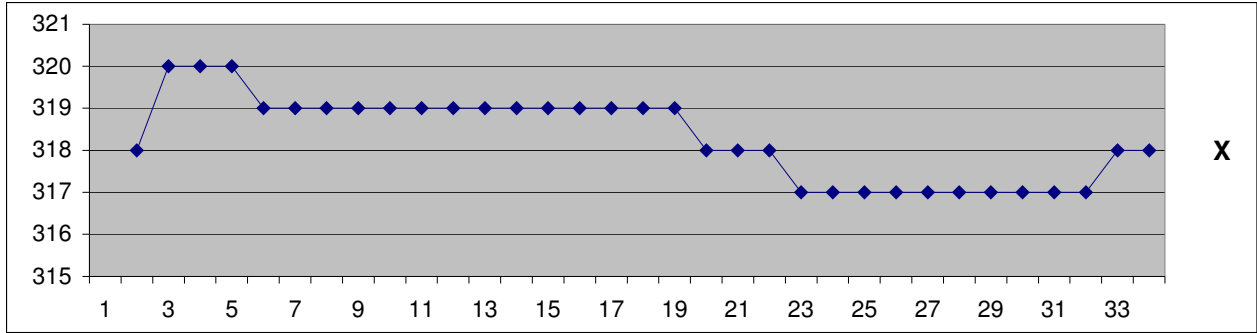


Figura 28: Variações na Horizontal da posição estimada da íris.

Na Figura 29, observa-se a representação gráfica do movimento vertical da íris na da íris apresentado na Figura 27 (com o usuário sempre direcionando os olhos para a lateral esquerda da tela)

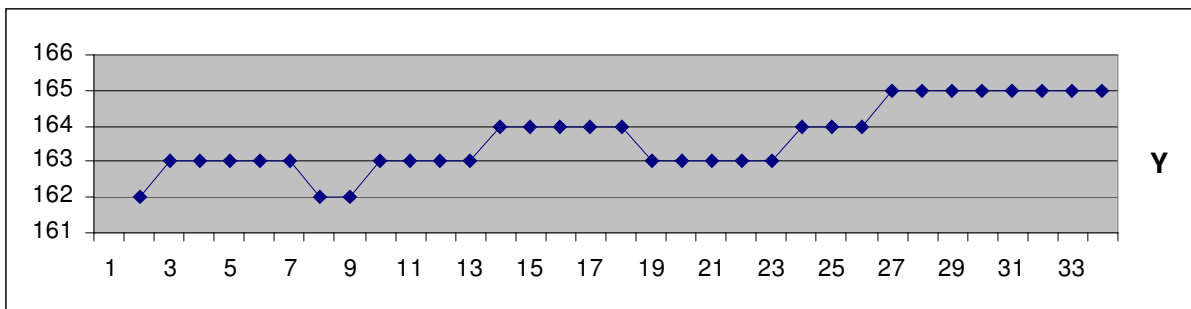


Figura 29: Variações na Vertical da posição estimada da íris.

A máxima amplitude alcançada é de 20 pixels e mesmo direcionando os olhos para um ponto fixo a variação identificada pela técnica é de 4 pixels, resultando em uma definição de apenas 5 regiões (20/4) na horizontal e 2 (10/4) regiões na direção vertical.

4.1. POSICIONAMENTO DO CURSOR DO MOUSE

Para avaliar a eficácia do posicionamento do ponteiro do mouse, criou-se uma fase inicial de calibração no sistema, onde o usuário deve direcionar os olhos para cada uma das bordas da tela e são armazenados os pontos de máximo e de mínimo nos eixos X e Y da posição da íris do olho esquerdo. A partir desses pontos são definidas 10 regiões nas quais o usuário poderá posicionar a íris de forma a mover o mouse. A implementação realizada mostra continuamente duas telas: uma com o rosto do usuário sobreposta com o “grid” correspondem (esta tela permanece fixa no centro para que, caso o usuário desloque o rosto, tenha um referência para reposicioná-lo). E, outra menor, apenas com o detalhe do olho, que acompanha o ponteiro de forma que o usuário sempre tenha uma referência. A Figura 30 ilustra as duas telas.

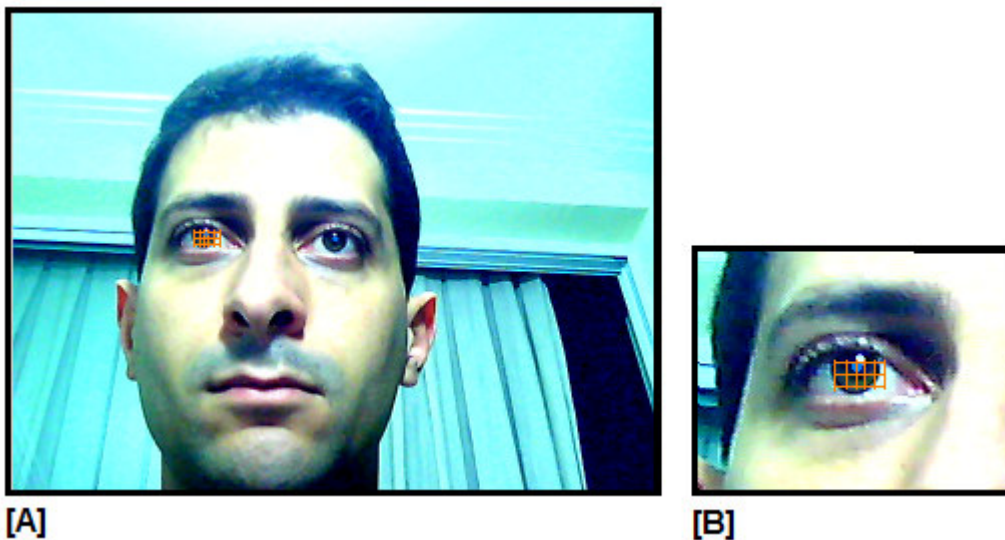


Figura 30: Em [A] tela com o rosto inteiro para facilitar o reposicionamento da face, Em [B] pequena tela que acompanha o mouse e fornece uma referência constante ao usuário.

O “clique” do mouse é registrado depois que a íris deixa de ser encontrada por 8 ciclos de busca (um ciclo completo gasta cerca de 300 ms).

Um aplicativo simples com 10 grandes botões foi desenvolvido com o objetivo de cobrir todo o monitor. Cada botão corresponde a uma das 10 regiões mencionadas anteriormente. O usuário deve clicar nos 10 botões em uma ordem fixa: 7523618049, foi escolhida uma sequência aleatória que possuísse todos os dígitos onde o usuário tivesse que fazer movimentos dos olhos atravessando a tela do computador na vertical, horizontal e diagonal. O tempo total para clicar em todos os botões foi computado. A Figura 31 ilustra uma imagem capturada do aplicativo.



Figura 31: tela aplicativo utilizado para computar o tempo médio.

Foram realizados testes com 15 voluntários. Para as condições utilizadas (iluminação fluorescente frontal e distância do monitor de cerca de 50 cm), o tempo médio foi de 20 segundos entre o posicionamento e o clique para cada um dos botões. Adicionalmente pode-se observar que a medida que o usuário se “acostuma” com o

sistema há uma redução no tempo gasto para clique. A Tabela 4 apresenta os resultados obtidos.

Tabela 4 Resultados de tempo por clique.

	1º teste	2º teste	Média
Indivíduo 1	18,94	18,77	18,86
Indivíduo 2	34,38	21,27	27,83
Indivíduo 3	18,95	15,78	17,37
Indivíduo 4	17,27	19,20	18,24
Indivíduo 5	24,82	20,73	22,78
Indivíduo 6	16,98	16,38	16,68
Indivíduo 7	20,26	-	20,26
Indivíduo 8	17,31	24,23	20,77
Indivíduo 9	15,51	15,10	15,30
Indivíduo 10	31,89	30,26	31,07
Indivíduo 11	22,43	19,56	21,00
Indivíduo 12	14,93	-	14,93
Indivíduo 13	12,23	-	12,23
Indivíduo 14	18,47	18,31	18,39
Indivíduo 15	19,45	18,78	19,12
Média	20,25	19,86	20,08

A maior dificuldade é, sem dúvida, o posicionamento do mouse que, com alguma frequência, se direciona para as regiões vizinhas e eventualmente, mesmo com os olhos fechados, o algoritmo “encontra” as íris no contorno das sombrancelhas e acaba por deslocar a posição do mouse. A Figura 32 mostra os 15 voluntários que se submeteram aos testes.



Figura 32: Voluntários que se submeteram ao teste para aferição do tempo médio necessário para um clique

4. MELHORIAS E TRABALHOS FUTUROS

O tempo médio por clique de 20 segundos é sem dúvida restritivo para a utilização cotidiana, entretanto, algumas melhorias poderiam ser criadas para favorecer a utilização, o posicionamento do mouse poderia, por exemplo, ser incremental, desta forma a identificação de apenas 5 regiões seriam suficientes como: central, direita, esquerda, acima e abaixo conforme a Figura 33[A]. Isto gera outro problema a ser contornado, imagine que o ponteiro está localizado na extrema esquerda da tela, e que o usuário deseje movê-lo apenas um pouco para a direita, seria necessário direcionar os olhos para a outra extremidade, e, neste caso, não seria possível ver o ponteiro dificultando um posicionamento mais refinado do mouse. Uma solução seria utilizar telas menores, que indicariam a posição do mouse em relação à tela (Figura 33[B]) e outra com um “recorte” da região onde o mouse se encontra conforme a Figura 33[C]. Estas

telas deveriam ser flutuantes e se posicionar de acordo com a região sugerida pelo algoritmo de maneira que o usuário poderia inferir a posição do ponteiro sem olhar para o mesmo.



Figura 33: Em [A] a restrição para apenas 5 regiões, em [B] a tela resumo com a identificação da posição do mouse, em [C] o detalhe da região do mouse.

Outro ponto a ser melhorado é a identificação da “piscada” a técnica é voltada para a identificação de íris e por vezes encontra falsos positivos, além de tomar 2,4 segundos (8 ciclos X 300ms) apenas para interpretar que o olho está fechado

É também necessário o desenvolvimento de uma interface amigável que favoreça o usuário comum na definição de parâmetros que aumentariam a eficácia do método (proporção entre olhos e face e distancia entre olhos) além de, se possível, ajustar os parâmetros da câmera com o sistema em execução o que permitiria a identificação de melhores parâmetros para aquela situação, além da possibilidade de se salvarem estes parâmetros para uso futuro.

5. CONCLUSÃO

A técnica proposta identifica as íris em 87% dos casos, considerando-se boas condições de iluminação proveniente de luz fluorescente e imagens predominantemente frontais. Os parâmetros da câmera de contraste e brilho também têm grande impacto na identificação das íris.

Como exposto anteriormente, optou-se por dar o máximo de liberdade ao usuário e não se exigir nenhum outro recurso como, por exemplo: zoom, iluminação infravermelha, câmeras com alta resolução ou fixação da face. Dadas estas condições, a amplitude máxima que se consegue captar no movimento das íris é de cerca de 20 pixels na horizontal e 10 pixels na vertical (Figura 26) Com os olhos fixos em uma posição, a variação é de cerca de 5 pixels (Figura 27). Assim, o número máximo de faixas sem superposição é de $5(20/4)$ na horizontal e $2(10/4)$ na vertical, o que limita a apenas $10(2*5)$ regiões no total.

Mesmo aceitando-se a restrição de 10 regiões no total, o tempo médio de 20 segundos alcançado pela aplicação parece ser restritivo para o uso cotidiano, desta forma é necessário o desenvolvimento de melhorias para permitir uma utilização abrangente.

6. REFERÊNCIAS BIBLIOGRÁFICAS

- [1] SCHNEIDERMAN, Henry; KANADE, Takeo. A Statistical Method for 3D Object Detection Applied to Faces and Cars. International Conference on Computer Vision, 2000
- [2] CANDIDO, Jorge; MARENGONI, Mauricio. Enhancing Face Detection Using Bayesian Networks, *IASTED International Conference on Signal and Image Processing – SIP2006*, v. 1, n. 1, 2006.
- [3] Site do fabricante de equipamentos de acessibilidade <http://www.tobii.com/>
- [4] Takahiro Ishikawa. Passive Driver Gaze Tracking With Active Appearance Models, Research Laboratories, DENSO CORPORATION, 2004
- [5] COOTES, Tim F.; TAYLOR, Chris J.; Statistical Models of Appearance for Computer Vision, Imaging Science and Biomedical Engineering, University of Manchester 2004
- [6] Dongheng Li, Ruqin Zhang. Eye Typing Using A Low-Cost Desktop Eye Tracker, Iowa State University 2005
- [7] VIOLA, Paul; JONES, Michael J. Robust Real-Time Face Detection. *International Journal Computer*, p.137-154, 2004.
- [8] FREUD, Y; SCHAPIRE, R.E. *Experiments with a New Boosting Algorithm, in Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 148-156, 1996.
- [9] Kun Peng, Liming Chen, Su Ruan and Georgy Kukharev. A Robust Algorithm for Eye Detection on Gray Intensity Face without Spectacles, LIRIS, Département MI, Ecole Centrale de Lyon, France 2005.
- [10] Dongheng Li; Derrick Parkhurst, ,Low-cost eye-tracking for human computer interaction, Iowa State University, Ames, Iowa 2006.
- [11] Biblioteca de Imagens IMM face database www2.imm.dtu.dk/
- [12] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8:679-714, 1986.
- [13] Fitzgibbon, A.; Pilu, M.; Fisher, R.B. “Direct least square fitting of ellipses” *Pattern Analysis and Machine Intelligence*, IEEE Transactions on Volume 21, Issue 5, May 1999 Page(s):476 – 480 Digital Object Identifier 10.1109/34.765658

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)