colloquium exactarum

MOUSE OCULAR PARA PESSOAS COM MOVIMENTOS LIMITADOS CAUSADOS POR PROBLEMAS CERVICAIS E/OU CEREBRAIS

EYE-TRACKING MOUSE FOR PEOPLE WITH LIMITED MOVEMENT CAUSED BY CERVICAL AND / OR CEREBRAL PROBLEMS

Vinicius Bitencourt dos Santos[1]; Mario Augusto Pazoti[1]; Francisco Assis da Silva[1]; Danillo Roberto Pereira[1]

[1]Faculdade de Informática – FIPP, Universidade do Oeste Paulista – UNOESTE
e-mail: vinibiten@hotmail.com, {mario, chico, danilopereira}@unoeste.br

RESUMO - Pessoas com movimentos limitados causados por problemas cervicais ou cerebrais muitas vezes não conseguem fazer o uso de um computador, em razão da dificuldade em movimentar os membros superiores. A solução apresentada neste trabalho busca prover acessibilidade para essas pessoas a fim de que possam se comunicar por meio de um computador a um baixo custo e com o mínimo de intrusão possível. Para tanto, foi desenvolvido um sistema de visão computacional, a partir da aplicação das técnicas Viola e Jones (Adaboost) e Hough-Circles, capaz de realizar funções de um mouse e um teclado virtual por meio da detecção dos movimentos dos olhos e pálpebras do usuário. Com o sistema proposto, uma pessoa com limitações físicas é capaz de utilizar funções simples de um computador movimentando olhos e pálpebras, sem o uso de sensores, suportes ou marcadores que pudessem ser intrusivos e aumentar o custo da solução.
Palavras-chave: Mouse Ocular; Acessibilidade; Tecnologia Assistiva.

ABSTRACT - People with limited movement caused by cervical or brain problems are limited on the use of a computer because of the difficulty in moving their arms. The solution presented in this paper aims to provide accessibility for these people so that they can communicate using a computer at a low cost and with minimal intrusion possible. Thus, a computer vision system was developed from the application of the techniques Viola and Jones (AdaBoost) and Hough-Circles, capable of performing functions of a mouse and a virtual keyboard by detecting eye movements and eyelids of the user. With the proposed system, a person with physical limitations is able to use simple functions of a computer just moving the eyes and eyelids, without using sensors, brackets or markers that could be intrusive and raise the cost of the solution.
Keywords: Eye-Tracking Controller; Assistive Technology; Accessibility.

Recebido em: 25/05/2015
Revisado em: 20/06/2015
Aprovado em: 15/07/2015

1 INTRODUÇÃO

Muitas pessoas com problemas cervicais e danos cerebrais, como, por exemplo, tetraplegia, pseudocoma e esclerose lateral amiotrófica (ELA), perdem praticamente todos os movimentos, ficando restritas ao leito e até mesmo à fala (CASSEMIRO; ARCE, 2004).

Neste projeto foi desenvolvido um sistema de visão computacional, por meio da análise do movimento dos olhos e das pálpebras do usuário e sem o uso de marcadores ou sensores, capaz de simular as funções de um mouse e utilizar o teclado virtual, possibilitando o simples manuseio de um computador. Dessa forma, o sistema é capaz de fornecer acessibilidade para as pessoas com alguma das limitações citadas. Analisando ferramentas similares à proposta deste trabalho, constatou-se que para o seu uso fazem-se necessários alguns objetos para posicionar a câmera. Como exemplo, pode-se citar o uso de um boné, como apresentado no trabalho de Zampirolli e Cuzziol (2011) ou até mesmo de sensores que captam a movimentação dos olhos e pálpebras, conforme relatado no trabalho de Cassemiro e Arce (2004).

O uso desses objetos, acoplados ao usuário, tendem a aumentar o custo da solução e limita o uso em pessoas com problemas vertebrais, pois elas não podem sofrer movimentos bruscos, para se instalar algum sensor ou vestir um boné (SCRAMIN; MACHADO, 2006).

Com base nisso, o presente trabalho propôs uma solução baseada na aplicação de técnicas de visão computacional, para captura e análise do movimento dos olhos por meio de uma webcam. A partir disso, buscou-se mover o cursor do mouse e acessar as teclas de um teclado virtual a fim de proporcionar acessibilidade para pessoas que apresentam as limitações citadas anteriormente. A partir da solução desenvolvida neste trabalho, uma pessoa com tais limitações será capaz de realizar algumas funções apenas movimentando os olhos e as pálpebras, como, por exemplo, leitura de um texto, podendo controlar a barra de rolagem lateral, escrever uma mensagem com o teclado virtual, acessar um aplicativo, como bloco de notas, entre outras ações.

Este artigo é composto por cinco seções, organizadas da seguinte maneira: uma breve introdução, contendo a motivação e o objetivo do trabalho, foi apresentada na Seção 1; em seguida, são descritos os materiais e métodos utilizados para o desenvolvimento do projeto na Seção 2; na Seção 3 são detalhadas as etapas do desenvolvimento da solução; os resultados obtidos são apresentados na Seção 4, bem como os problemas identificados durante sua

implementação; por fim, na Seção 5, são apresentadas as considerações finais do presente artigo, apresentando as contribuições do projeto, bem como as possibilidades de melhoria e continuidade por meio da proposta de trabalhos futuros.

2 MATERIAIS E MÉTODOS

Para o desenvolvimento do projeto foi necessária uma câmera de alta resolução (720p) acoplada a um notebook. A partir do estudo de técnicas de visão computacional, foram definidos métodos para aquisição da imagem do usuário, seleção da região de interesse, pré-processamento da imagem e identificação dos olhos. Para aplicação das técnicas foram usadas funções da biblioteca OpenCV, por meio da camada Emgu CV (MARENGONI; STRINGHINI, 2009) e para desenvolvimento do protótipo da aplicação foi utilizada a linguagem C#. As técnicas adotadas para processamento dos quadros do vídeo capturado pela webcam e detecção dos movimentos dos olhos são detalhadas na subseções a seguir.

2.1 Viola-Jones

Viola e Jones (2004) desenvolveram uma técnica para reconhecimento de face, a qual se baseou na representação das características extraídas com os filtros de Haar (BRKIĆ, 2010). Para isso foi montado um classificador utilizando o AdaBoost, que a partir das características significantes de uma face, selecionadas na fase de treinamento, realiza a comparação dessas características com as características extraídas de cada quadro do vídeo analisado (ARAUJO, 2010).

AdaBoost é um método que aprimora o desempenho de um algoritmo de aprendizado, aplicado juntamente a outras formas de aprendizado, como por exemplo, redes neurais. Seu objetivo é fazer com que classificadores fracos se tornem classificadores fortes ajustando a classificação de acordo com os erros encontrados durante a classificação anterior (CHAVES, 2012).

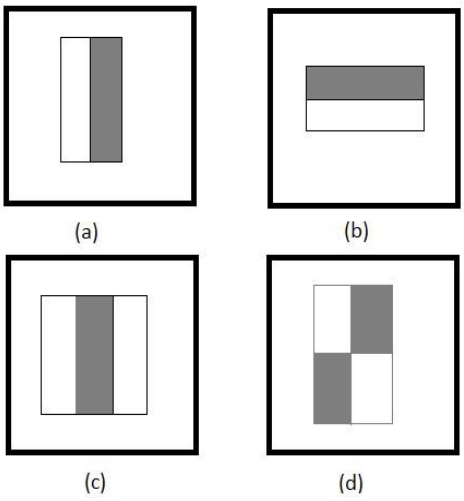

Figura 1. Características Básicas de Haar.
Fonte: Adaptado de Viola e Jones (2004)

O conjunto de características de Haar, utilizado por Viola e Jones (2004), é composto por características de dois, três ou quatro retângulos, como mostrados na Figura 1, em que (a) representa o conjunto de dois retângulos com adjacência vertical, (b)

representa o de dois retângulos com adjacência horizontal, (c) representa o conjunto de três retângulos e, por fim, (d) representando o de quatro retângulos.

Para obter o valor da característica, subtrai-se o somatório da intensidade dos pixels pertencentes aos retângulos brancos do somatório da intensidade dos pixels pertencentes aos retângulos cinza.

A partir do valor da característica calculado é possível realizar a comparação com o valor do limiar, que é obtido por meio do treinamento. Se este valor estiver dentro dos valores do treinamento, a região é considerada como uma área de interesse na imagem.

A aplicação das características de Haar em uma imagem examina os contrastes naturais da face, como exemplificado na Figura 2. Note que nessa figura é mostrada a relação entre as características e o rosto, cuja subtração dos somatórios usando os dois conjuntos apresenta valores superiores ao limiar estabelecido no treinamento.



Figura 2. Relação entre as características de Haar e a imagem do rosto.

2.2 *Smooth*

Algoritmo usado no pré-processamento para suavização da imagem. Ao aplicar a suavização, o reconhecimento e segmentação da imagem se tornam mais fácil, uma vez que bordas que não estão bem definidas na imagem irão desaparecer conservando apenas as bordas que são importantes para o reconhecimento (MANZI, 2007). A suavização é feita por meio da convolução de um filtro determinado para cada aplicação, já que quanto maior o filtro, maior o grau de suavização. Na Figura 3 é possível observar um filtro de convolução e na Figura 4(b) o resultado da aplicação deste filtro (FARIA, 2005).

$$\frac{1}{273}.\begin{bmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{bmatrix}$$

Figura 3. Aproximação discreta da função Gaussiana.
Fonte: (Faria, 2005)



Figura 4. Resultado da aplicação da suavização.
Fonte: (Faria, 2005).

Como visto, o resultado da aplicação da suavização com filtro gaussiano é uma imagem borrada, com a diminuição de ruídos que poderiam ser considerados bordas pelo detector.

2.3 Canny

Algoritmo utilizado para detecção de bordas que, segundo Vale e Dal Poz (2002), é um algoritmo que visa buscar um ótimo detector de bordas. Por meio de um valor limiar predefinido, esse método consegue chegar a um detector de bordas capaz de diferenciar o que é um preenchimento do que é uma borda.

Canny (1986) estabeleceu que um ótimo algoritmo detector de bordas deve conter uma baixa taxa de erro e a localização da borda detectada deve estar o mais próximo possível de onde a borda se encontra na imagem. E para isso, ele desenvolveu um detector que minimiza os ruídos e aperfeiçoa a localização das bordas (SILVA et al, 2004).

Na Figura 5 é mostrado o resultado da aplicação do detector Canny na região de interesse correspondente ao olho de uma pessoa.

2.4 Hough-Circles

A transformada de Hough-Circles visa encontrar o centro de uma circunferência de raio *R* em uma imagem. Essa transformada cria uma matriz com a mesma dimensão da imagem, que armazenará números inteiros inicializados com zero (PISTORI; PISTORI; COSTA, 2005).

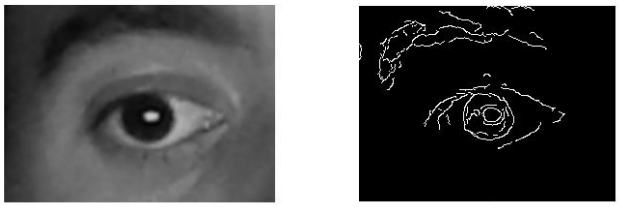

Figura 5. Aplicação do detector de bordas proposto por Canny.

Depois de criada a matriz, todos os pixels da imagem são percorridos e ao encontrar um pixel em destaque, branco no caso da imagem com fundo preto (ou preto no caso da imagem com fundo branco), é feito o incremento na matriz de todos os pixels que pertencem a um círculo de raio R com o ponto central em que o pixel destaque foi encontrado. Os pixels pertencentes a esse círculo de raio R e centro no pixel destaque podem ser encontrados por meio das Equações 1 e 2 em que *Xc* e *Yc* são as coordenadas do pixel destaque, *X* e *Y* são as coordenadas dos pixels pertencentes ao círculo e *R* o raio, com θ iniciando em 0 e variando até 2π. (PISTORI; PISTORI; COSTA, 2005).

$$Y = Yc + R * \sin(\theta) \quad (1)$$

$$X = Xc + R * \cos(\theta) \quad (2)$$

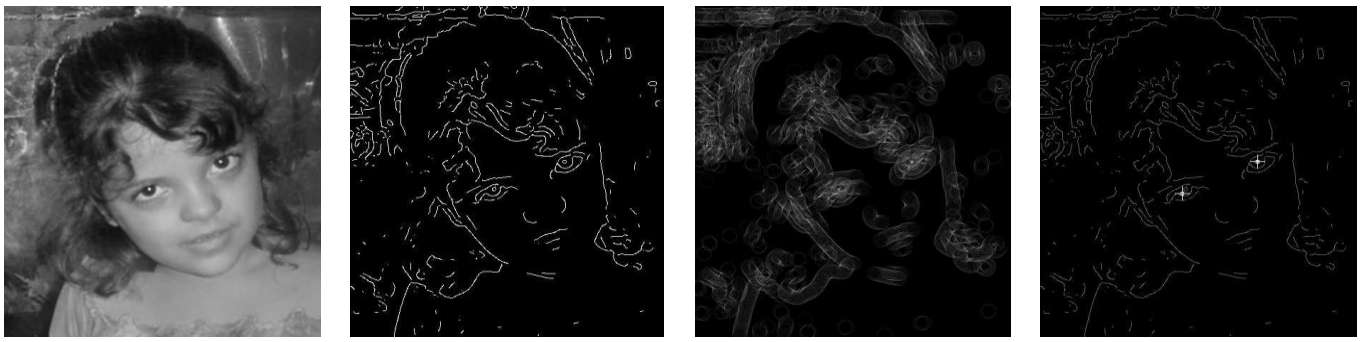

Figura 6. Processo da aplicação da técnica de Hough-Circles.
Fonte: (PISTORI; PISTORI; COSTA, 2005).

Ao terminar de percorrer todos os pixels da imagem e fazer o incremento na matriz, os prováveis pixels centrais de círculos de raio *R* terão um valor maior na matriz e serão considerados centros de possíveis círculos (PISTORI; PISTORI; COSTA, 2005).

Na Figura 6 são mostradas as etapas da técnica Hough-Circles, em que a imagem inicial (a) é a imagem de uma menina em escala de cinza; a imagem (b) representa o resultado a aplicação da técnica de detecção de bordas; a imagem (c) representa a matriz de incremento após a aplicação da Transformada de Hough-Circles em que a região do olho da criança possui maior intensidade com base nos valores de incremento; e, por fim, na imagem (d) destaca-se a centro do olho da criança devido à alta concentração de incrementos naquela região.

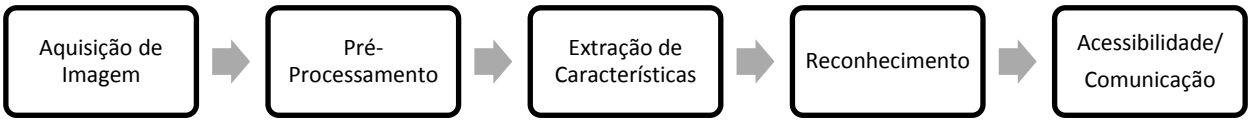

Figura 7. Diagrama de um sistema de visão computacional para identificação do movimento ocular.

3 DESENVOLVIMENTO

A ferramenta desenvolvida consiste em um sistema de visão computacional com o intuito de replicar as funções de um mouse e teclado virtual, por meio da identificação do movimento dos olhos e pálpebras do usuário. O sistema foi projetado de acordo com as etapas mostradas na Figura 7. A aquisição de imagem consiste na obtenção de um quadro que contém o rosto do usuário, já em escala de cinza, por uma *webcam* HD (720p) acoplada a um notebook. Após a aquisição de imagem é feito o Pré-Processamento, que consiste na identificação da região de interesse (ROI - *Region Of Interest*), que neste caso é a região dos olhos, por meio da

técnica proposta por Viola e Jones. Posteriormente, é aplicada a técnica de Smooth para remoção de possíveis reflexos na íris. Para a identificação da ROI foram utilizados os arquivos XML com os parâmetros de Haar gerados a partir do treinamento para detecção de regiões da face e disponibilizados pela biblioteca do OpenCV. A ROI é utilizada para que os cálculos sejam feitos apenas na região que realmente possui algum significado para o sistema, diminuindo a quantidade de falsos objetos na imagem. Após definida a ROI é aplicada a técnica de Hough-Circles para identificar o centro dos círculos formados pelas íris do usuário. A partir da localização do centro da íris, o sistema captura o posicionamento da íris a cada quadro para determinar a direção do olhar do usuário e, com isso, mover o mouse ou acessar o teclado virtual, proporcionando a acessibilidade para as pessoas com problemas cervicais e ou cerebrais.

A transformada de Hough-Circles é aplica da por meio da função cvHoughCircles com o seguintes parâmetros:

-A dimensão da matriz acumuladora é quatro vezes menor que a dimensão da ROI.

- a distância mínima entre os centros dos círculos detectados é igual à metade da altura da ROI.

- o maior limiar para aplicação do Canny é de 70.

- o limiar para a detecção do centro foi definido com o valor de 90.

- o menor raio a ser encontrado é de 10.

- O maior raio a ser encontrado é de 40.

Os parâmetros foram definidos empiricamente por meio de testes preliminares.

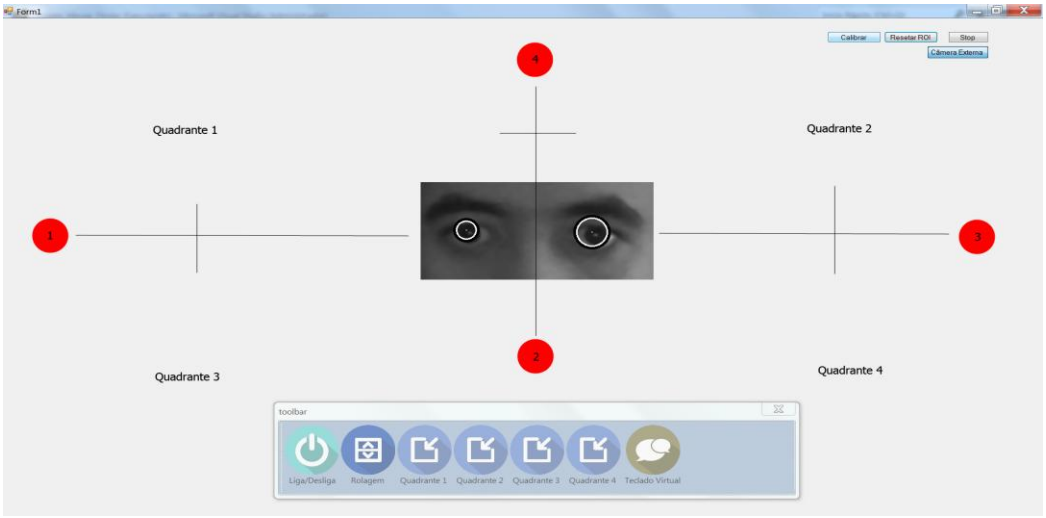

Figura 8. Destaque dos marcadores para calibragem, barra de ferramentas e quadrantes.

Entretanto, um processo de calibragem é necessário para a adequação da ferramenta para diferentes pessoas ou diferentes distâncias entre o usuário e a câmera. Sendo assim, quando o usuário olhar para o centro da tela, o sistema irá comparar os dados obtidos na calibragem com os atuais e entenderá que o olho do usuário está voltado para o centro.

Isso é possível, pois na calibragem é solicitado para que o usuário olhe para pontos fixos da tela por um período de três segundos para cada ponto, conforme indicados na Figura 8 por círculos vermelhos enumerados de 1 a 4 e, durante esses três segundos, faz-se a média do ponto central da íris, representado pelo P_{centro} na Figura 9.

Os pontos médios também estão representados na Figura 9, em que o $P_{esquerda}$ representa o ponto médio calculado ao olhar para o círculo de número 1, o P_{baixo} representa o ponto médio calculado ao olhar para o círculo de número 2, o $P_{direita}$ representa o ponto médio calculado ao olhar para o círculo de número 3 e o P_{cima} representa o ponto médio calculado a o olhar para o círculo de número 4.

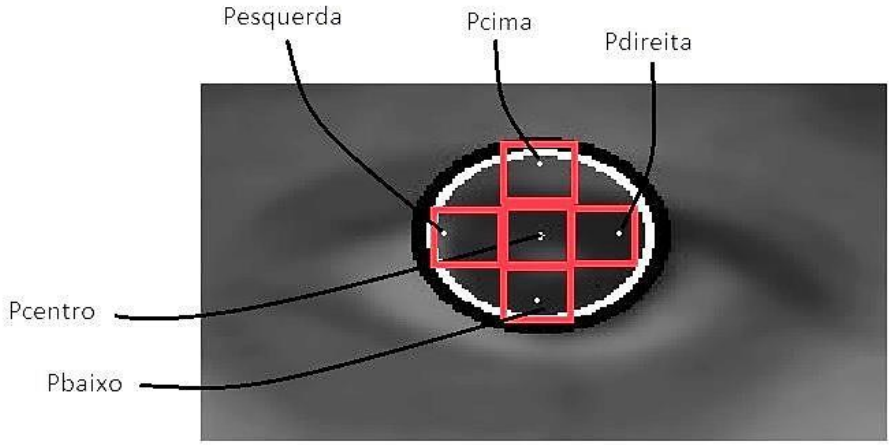

Figura 9. Imagem do olho com a marcação das regiões e pontos médios.
Fonte: Gerado pelo autor.

Depois de calculados os pontos médios, definem-se as regiões, indicadas na Figura 9, pelos quadrados em vermelho. Essas regiões são calculadas da seguinte forma:

- Calculam-se as distâncias entre os pontos médios na vertical (*Dist_Vertical*) e horizontal (*Dist_Horizontal*) por meio da Equações 3 e 4.

$$Dist_Vertical = P_{cima}.Y - P_{baixo}.Y \quad (3)$$

$$Dist_Horizontal = P_{direita}.X - P_{esquerda}.X \quad (4)$$

- Os limites, X mínimo e X máximo das regiões horizontais e Y mínimo e Y máximo das regiões verticais, são calculados por meio das Equações 5 a 12:

Esquerda:

$$XMin = P_{esquerdo}.X - 2 \quad (5)$$

$$XMax = P_{esquerdo} + Dist_Horizontal/4 + 2 \quad (6)$$

Direita:

$$XMin = P_{direito}.X - Dist_Horizontal/4 - 2 \quad (7)$$

$$XMax = P_{direito} + 2 \quad (8)$$

Cima:

$$YMin = P_{cima}.Y - 2 \quad (9)$$

$$YMax = P_{cima}.Y + Dist_Vertical/4 + 2 \quad (10)$$

Baixo:

$$YMin = P_{baixo}.Y - Dist_Vertical/4 - 2 \quad (11)$$

$$YMax = P_{baixo}.Y + 2 \quad (12)$$

O centro é representado por todos os valores que não obedecerem aos limites das regiões Baixo, Cima, Esquerda ou Direita, pois o centro é uma região neutra para o sistema, o qual não executará nenhuma função dependente do movimento dos olhos.

Com estes valores delimitados, termina-se o processo de calibragem e o sistema irá comparar a posição central da íris atual do usuário com essas regiões, em que para as regiões Cima e Baixo utilizasse para comparação a coordenada *Y* do ponto central atual e para as regiões Esquerda e Direita utilizasse a coordenada *X* do ponto central atual. Se a coordenada *X* ou *Y* do ponto central da íris atual estiverem dentro do limite de uma dessas regiões, o sistema irá reconhecer que o usuário está olhando para a região em questão e irá fazer a interpretação.



Figura 10. Barra de Ferramentas.
Fonte: Gerado pelo autor.

Com a direção do olhar identificada, algumas funções foram definidas por meio de associação com o uso padrão do computador, e estas funções, juntamente a os movimentos necessários para ativá-las, estão listadas na Tabela 1.

Tabela 1. Lista de funções com seus respectivos movimentos.

Função	Movimento
Realizar o clique do botão esquerdo ou acionar a tecla selecionada no teclado virtual	Fechar olho Esquerdo (2 segundos)
Reiniciar o ponteiro do mouse para canto inferior central	Fechar olho direito (2 segundos)
Ativar ou desativar a rolagem	Fechar os dois olhos (3 segundos)
Movimentar o ponteiro do mouse ou a barra lateral para direção desejada e navegar entre as teclas do teclado virtual.	Olhar para cima, para baixo, para esquerda ou para direita.

Para melhorar a usabilidade, foi desenvolvida uma barra de ferramentas, como mostrado na Figura 10, que estará sempre à vista do usuário, uma vez que está não é minimizada quando acessa outra aplicação.

A função reposicionar o ponteiro do mouse irá colocá-lo próximo a essa barra que apresenta algumas funções úteis para o

usuário, como ativar ou desativar o programa, ativar ou desativar a rolagem da barra lateral, posicionar o ponteiro do mouse em um dos quadrantes da tela, deixando assim o ponteiro mais próximo do local de que se deseja acessar, a partir da separação da tela em quatro regiões, e, por fim, abrir o teclado virtual, mostrado na Figura 11. Nesse teclado a navegação entre as teclas ocorre por meio do movimento dos olhos e a ação de clique dos botões é feita ao fechar os olhos.

4 RESULTADOS

Após a pesquisa e a implementação do sistema de visão computacional com as técnicas empregadas, foi possível que o usuário movimentasse o ponteiro do mouse e fizesse o uso do teclado virtual a partir do movimento dos olhos e pálpebras. O usuário consegue movimentar o cursor do mouse por toda a tela e realizar algumas simples funções, mesmo o sistema não apresentando uma precisão alta no posicionamento do ponteiro do mouse e não possibilitando a mesma agilidade quando comparado à utilização de um dispositivo como o mouse.

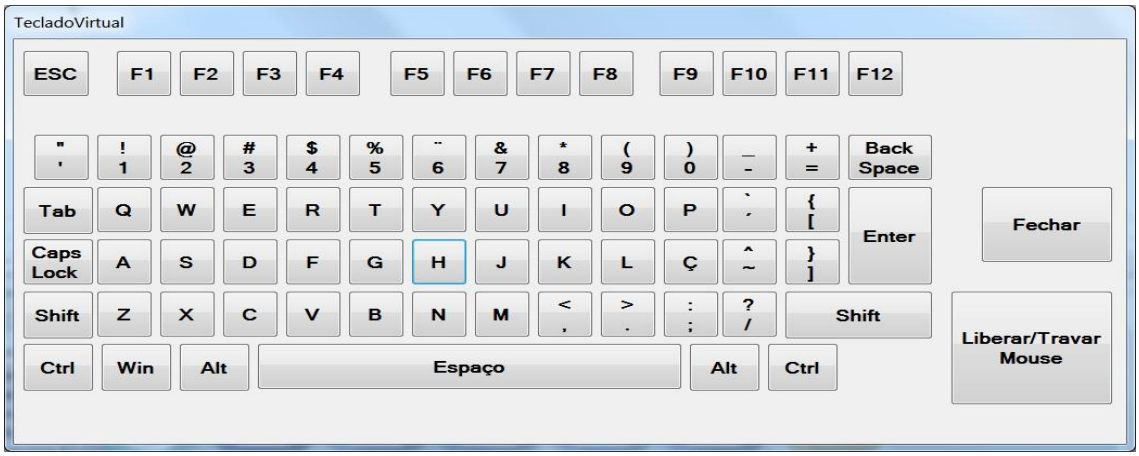

Figura 11. Teclado Virtual Desenvolvido.

Não foi necessária a utilização de marcadores nem objetos intrusivos para que o usuário possa utilizar o sistema. Recomenda-se apenas que o posicionamento da câmera e da tela não seja muito distante dos olhos do usuário e nem muito inclinadas, permitindo uma visualização confortável.

A distância de aproximadamente 70 centímetros entre usuário e câmera foi considerada adequada após a realização de testes pelos próprios autores.

Problemas com alta ou baixa iluminação do ambiente acabaram se tornando obstáculos durante a implementação, assim como o uso por pessoas que possuem óculos, uma vez que o sistema necessita obter imagens de alta qualidade dos olhos do usuário sem que haja

a obstrução da íris causado por objetos e lentes ou perda da definição das bordas da íris devido aos reflexos causados por alta ou baixa iluminação. Para que as imagens pudessem ser obtidas com maior qualidade, optou-se por substituir a webcam embutida do notebook por uma câmera com uma qualidade maior acoplada ao notebook.

O uso do sistema por pessoas sem deficiência pode ser considerado muito cansativo, pois demanda paciência e atenção contínua do usuário, uma vez que este pode realizar movimentos involuntários durante a utilização do sistema e, com isso, realizar alguma função indesejada.

Como a movimentação depende do movimento do olho do usuário, também se torna cansativo, pois é necessário que o usuário fixe o olhar por grande parte de tempo para realizar as movimentações.

Os comandos que o sistema executa de acordo com os movimentos executados, foram criados por meio de associação com o uso de um mouse comum, já que o clicar do botão esquerdo é feito com o piscar do olho esquerdo e as movimentações do ponteiro são feitas de acordo com a direção que usuário está olhando.

A ideia dos quadrantes também surgiu durante a fase de testes, pois é um processo lento movimentar o ponteiro de um extremo da tela até outro. Por isso, possibilitar o posicionamento do ponteiro em um local mais próximo do desejado foi considerado útil.

Com o uso da ferramenta foi possível utilizar um computador, escrever uma mensagem na tela e ler um texto que necessitava de navegação. Mesmo com os problemas encontrados durante a implementação e testes, os resultados obtidos foram considerados bastante satisfatórios.

5 CONSIDERAÇÕES FINAIS

Com a utilização de um sistema baseado em visão computacional, foi possível utilizar um computador por meio da movimentação do cursor do mouse, clique do botão esquerdo, rolagem da barra lateral e a utilização do teclado virtual apenas com a identificação da posição da íris e movimentação das pálpebras. Mesmo não possuindo a mesma precisão que se tem com o mouse tradicional e não utilizando todas as funções que um computador possui, foram considerados satisfatórios os resultados obtidos, levando em consideração que o acesso às funcionalidades do sistema é realizado apenas por um conjunto restrito de movimentos.

As operações ainda são limitadas, mas acredita-se que o projeto apresenta contribuições por não necessitar de equipamentos especiais ou marcadores para identificação do movimento dos olhos,

apenas fez-se o uso de técnicas de visão computacional.

A partir da adaptação dessa solução podem ser desenvolvidas novas aplicações, criando uma interface customizada por meio da qual não seja requerida grande precisão de movimento para o posicionamento do ponteiro do mouse, já que para posicionar o mouse em pequenas regiões apenas por movimento do olho tenha sido considerado um trabalho difícil nos testes realizados pelos autores.

Outra sugestão é a modificação do teclado virtual, acrescentando mensagens predefinidas como meio de facilitar a comunicação do usuário, que ao invés de ter que escrever uma mensagem completa pelo teclado, apenas selecione uma mensagem predefinida, aumentando assim o conforto.

O aprimoramento da solução, tornando possível o reconhecimento da íris mesmo com o uso de óculos, e a também customização para pessoas que possuam estrabismo são ideias que podem complementar o trabalho. Por fim, com as melhorias na solução apresentada e a realização de testes com pessoas com deficiência será possível realizar uma avaliação mais detalhada das contribuições e benefícios alcançados.

REFERÊNCIAS

ARAUJO, G.M. **Algoritmo para reconhecimento de características faciais baseado em filtros de correlação**. 2010. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio de Janeiro, Rio de Janeiro - RJ. Disponível em: <https://www.lps.ufrj.br/profs/eduardo/teses/gabriel-matos-mestrado.pdf>. Acesso em: 05 nov. 2013.

BRKIĆ, K. **An overview of traffic sign detection methods**. 2010. Croatia: Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3, 10000 Zagreb, 2010.

CANNY, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v.8, n.6, p.679-714, 1986. http://dx.doi.org/10.1109/TPAMI.1986.4767851

CASSEMIRO, C.R.; ARCE, C.G. Comunicação visual por computador na esclerose lateral amiotrófica. **Arq Bras Oftalmol.**, v.67, n.2, p.295-300, 2004. http://dx.doi.org/10.1590/S0004-27492004000200020

CHAVES, B.B. **Estudo do algoritmo AdaBoost de aprendizagem de máquina aplicado a sensores e sistemas embarcados**. 2012. 119f. Dissertação (Mestrado em Engenharia Mecânica) – Escola Politécnica da Universidade de São Paulo, São Paulo - SP.

FARIA, D.R. **Reconhecimento de impressões digitais com baixo custo computacional para um sistema de controle de acesso**. 2005. 100f. Dissertação (Mestrado) – Universidade Federal do Paraná, Curitiba - PR.

MANZI, F.A. **Aplicação de visão computacional para extração de características em imagens do olho humano**. 2010. 47 f. Trabalho de Conclusão de Curso (Engenhria da Computação) – Universidade de São Paulo, São Carlos. Disponível em:

<http://www.tcc.sc.usp.br/tce/disponiveis/18/182000/tce-05052010-135906/publico/Manzi_Filipe_Augusto.pdf>. Acesso em: 4 nov. 2013.

MARENGONI, M.; STRINGHINI, S. Tutorial: introdução à visão computacional usando Opencv. **Revista de Informática Teórica e Aplicada**, v.16, n.1, p.125-160, 2009.

PISTORI, H.; PISTORI, J.; COSTA, E.R. Hough-circles: um módulo de detecção de circunferências para o ImageJ. In: WORKSHOP DE SOFTWARE LIVRE, 6. **Anais...** Porto Alegre, 2005. Disponível em: <http://www.gpec.ucdb.br/pistori/publicacoes/pistori_wsl2005.pdf>. Acesso em: 5 nov. 2013.

SCRAMIN, A.P.; MACHADO, W.C.A. Cuidar de pessoas com tetraplegia no ambiente domiciliário: intervenções de enfermagem na dependência de longo prazo. **Esc. Anna Nery**, Rio de Janeiro, v.10, n.3, p.501-508, dez. 2006. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1414-81452006000300020&lng=en&nrm=iso>. Acessado em: 05 set. 2013.

SILVA, J.F.C. et al. Avaliação da qualidade da detecção de bordas em uma sequência de imagens de ruas e rodovias**.** **Revista Brasileira de Cartografia**, v.56, n.2, p.96-103, 2004.

VALE, G.; DAL POZ, A.P. Processo de detecção de bordas de Canny. **Boletim de Ciências Geodésicas,** UFPR: Curitiba, v.8, n.2, p.67-78, 2002.

VIOLA, P.; JONES, M. J. Robust real-time face detection. **International journal of computer vision,** v.57, n.2, p.137-154, 2004. http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb

ZAMPIROLLI, F.A.; CUZZIOL, R.I. Mouse ocular para pseudocoma. In: ENCONTRO DE INICIAÇÃO CIENTÍFICA. **Anais...** Santo André, 2011. Disponível em: <http://www.infonorchile2012.uta.cl/download.php?file=infonor2012_10.pdf>. Acesso em: 5 nov. 2013.