

UM COMPONENTE PARA DETECÇÃO DE PISCADA DE OLHOS UTILIZANDO A INTERAÇÃO HOMEM COMPUTADOR

Érika A. Sardinha
Instituto Federal de Goiás
erika.sardinha@outlook.com

Carlos V. S. Ferreira
Instituto Federal de Goiás
carlosvalter@hotmail.com

Poliana S. P. Cassiano
Instituto Federal de Goiás
poliana.piva@gmail.com

Luiz F. B. Loja
Instituto Federal de Goiás
luiz@doutorado.ufu.br

Gabriela E. Meireles
Instituto Federal de Goiás
gabrielameireles18@gmail.com

Renato S. Gomide
Instituto Federal Goiano
renato.s.gomide@gmail.com

Resumo – O objetivo deste trabalho é o desenvolvimento de um componente que detecte a piscada de olhos transformando essa ação em um comando computacional. Este componente foi projetado para ser facilmente adicionado a novas aplicações. Assim, essa ferramenta deve incentivar o desenvolvimento de aplicações que promovam formas de comunicação para pessoas que possuam a Síndrome do Encarceramento. Pois, grande parte das soluções para esse fim são inacessíveis, seja pelo alto valor de custo ou pelas tecnologias adotadas. O componente proposto é uma solução de baixo custo e acessível, pois utiliza *webcams* comuns e um computador pessoal como meio físico de entrada de dados.

Palavras-Chave – tecnologia assistiva, comunicação suplementar e alternativa, piscar de olhos, detecção de face.

AN INTERACTION COMPONENT MAN-COMPUTER TO DETECT BLINKING EYES

Abstract - The objective of this work is to create a component that detects the eyeblink and turning this action in a computer command. The proposed component was designed which can be added to new applications. Therefore, this tool should encourage researchers to develop application that provides ways of communication for people with Locked in Syndrome (LIS). Solutions that provide communication for people with LIS are usually inaccessible and have a high cost. The proposed component is a solution with low-cost and most affordable. Our component proposal uses ordinary webcam and personal computer as a way of data input.

Keywords - Authors shall provide a maximum of 6 keywords (in alphabetical order) in order to identify the major topics of the paper.

I. INTRODUÇÃO

O acesso às tecnologias de informação e comunicação tem impactado significativamente a forma como as pessoas se relacionam com o mundo. Essas mudanças afetam a realização de atividades do dia a dia, as relações humanas, o trabalho e a aprendizagem [6]. Entretanto, grande parte dessas tecnologias são construídas desconsiderando as limitações de pessoas com deficiência.

Até o ano 2000 mais de 10% de toda população mundial sofria de algum tipo de deficiência [13]. Dados do Censo 2010 revelaram que quase 24% da população brasileira – 45,6 milhões de pessoas – têm algum tipo de deficiência [13]. Muitas dessas limitações afetam o aspecto cognitivo dos indivíduos, prejudicando a descoberta e o reconhecimento das informações [1].

Além disso, existem deficiências que prejudicam as capacidades motoras dos indivíduos. Em algumas situações esse tipo de deficiência pode prejudicar gravemente a capacidade motora do indivíduo limitando todos os movimentos corporais voluntários.

A Síndrome do Encarceramento (SE) é uma condição rara em que uma pessoa apresenta paralisia de aproximadamente todos os músculos voluntários do corpo. Apesar das funções cognitivas da pessoa são preservadas [7]. A SE pode ser causada por paralisia cerebral, Esclerose Lateral Amiotrófica (ELA), tetraplegia e doenças degenerativas do sistema nervoso [6].

Assim, os indivíduos que possuem SE tornam-se incapazes de estabelecer uma comunicação ou interação tanto de forma oral como gestual [1]. Portanto, para possibilitar a comunicação dessas pessoas são necessários dispositivos que forneçam meios alternativos de comunicação e interação com o ambiente exterior [6].

A área responsável por desenvolver esses meios alternativos de comunicação é chamada de Tecnologia Assistiva (TA). O conceito de TA surgiu pela busca de um rompimento de barreiras impostas a indivíduos que sofrem danos causados por distúrbios físicos e/ou cognitivos [1].

A TA é uma área do conhecimento de característica interdisciplinar. Essa área engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiência, incapacidades ou mobilidade reduzida. O objetivo principal dessa área de conhecimento é possibilitar a autonomia, independência,



XIV CEEL - ISSN 2178-8308
03 a 07 de Outubro de 2016
Universidade Federal de Uberlândia - UFU
Uberlândia - Minas Gerais - Brasil

qualidade de vida e inclusão social das pessoas com deficiência [1].

A área da TA destinada ao estudo da comunicação assistiva é denominada Comunicação Alternativa e Aumentativa (CAA) [1]. Essa área de pesquisa proporciona um conjunto de ferramentas e estratégias utilizadas para resolver desafios cotidianos de comunicação de pessoas que apresentam algum tipo de comprometimento da linguagem oral, na linguagem gestual e na interação [11].

Os pesquisadores da área de CAA desenvolvem mecanismos que possibilitam as pessoas com SE se comunicarem [1]. Entretanto, esses mecanismos possuem limitações. Algumas dessas limitações são, a baixa velocidade na interação do usuário com o computador e o alto custo para a aquisição dos equipamentos [12].

Essas duas limitações tem uma relação direta com a situação do paciente. Por exemplo, uma pessoa com SE pode executar apenas dois tipos de sinais biológicos. São eles o movimento dos olhos e os sinais relacionados às atividades cerebrais [6]. Esses dois estímulos são limitados exigindo que os dispositivos de CAA sejam bem elaborados e possuam alta tecnologia.

O sinal do movimento dos olhos pode ser adquirido por meio de câmeras de vídeo. A partir da imagem adquirida pela câmera os algoritmos de computação gráfica identificam se o olho está aberto ou fechado. Assim, esses sinais são convertidos em dois estados ativo e inativo [6]. A partir desses estados o computador pode executar comandos específicos usando a interpretação da piscada.

Existem bases de dados de fabricantes de sistemas que utilizam a detecção de movimentos oculares para interação homem-máquina [2]. Porém, essas ferramentas nem sempre são acessíveis em virtude do valor ou devido as tecnologias utilizadas nas soluções. Entretanto, desenvolver aplicações de TA de fácil acesso, assim como prover meios para facilitar o desenvolvimento dessas aplicações é importante para melhorar a qualidade de vida das pessoas com deficiência.

Implementar um mecanismo de comunicação utilizando os olhos não é uma tarefa trivial. Para construir esses dispositivos é necessário conhecimento nas áreas de desenvolvimento de *software* e processamento de imagens. Entretanto, muitos pesquisadores desejam implementar sistemas de CAA sem se preocupar com a interface de entrada de dados. Esses pesquisadores podem desistir de seus projetos, uma vez que o programa de CAA não funcionaria sem essa interface.

A dificuldade de encontrar ferramentas e componentes que facilitem o desenvolvimento de soluções efetivas e de baixo custo para interação homem computador podem impedir a criação de sistemas de CAA acessíveis a um público mais extenso [12].

Embora existam ferramentas gratuitas, de código aberto e que contam com os fóruns de apoio aos desenvolvedores iniciantes, como as bibliotecas OpenCV e JavaCV, ainda existe uma deficiência de material para auxiliar no entendimento e manuseio dessas ferramentas.

Muitas vezes os desenvolvedores querem apenas uma forma de aplicar as funcionalidades oferecidas por tais bibliotecas sem a necessidade de conhecer e entender como

elas funcionam de fato. Principalmente se considerar que a compreensão desses aspectos não é tão simples.

O objetivo deste trabalho é desenvolver um componente do tipo *plug-in* capaz de detectar a piscada do olho. Esse componente usará ferramentas de baixo custo e acessíveis. Além disso, esse mecanismo poderá ser adicionado em qualquer sistema de CAA que utilize o piscar dos olhos como dados de entrada. Assim, com alguns passos simples o componente poderá ser adicionado às mais diversas aplicações que traduzem a piscada de olhos em comandos computacionais.

Este trabalho está estruturado da seguinte forma, a seção II descreve a arquitetura e as tecnologias utilizadas na elaboração do componente proposto. O processamento digital de imagens é mostrado na seção III. O funcionamento do componente é descrito na seção IV. Finalmente, a seção V traz as conclusões deste trabalho.

II. ARQUITETURA E TECNOLOGIAS

Essa seção descreve a arquitetura e tecnologias utilizadas na elaboração do componente proposto. Além disso, são apresentados os padrões de projeto aplicados no desenvolvimento do componente, as técnicas de programação e bibliotecas utilizadas.

O componente foi projetado e implementado na plataforma Windows, utilizando a linguagem de programação Java. Foram utilizadas as bibliotecas OpenCV e JavaCV e a parte gráfica está sendo criada usando a plataforma de software multimídia JavaFX.

A. Linguagem de programação Java

O Java é uma linguagem de programação de alto nível, multiplataforma e orientada a objetos (OO). Por usar o paradigma OO é baseada em elementos do mundo real que são modelados e representados por classes [14].

As classes descrevem um conjunto de características e as ações associadas a elas. Uma vez projetadas, essas classes, podem ser utilizadas para solucionar os mais diversos problemas dentro de um sistema [14].

Um conjunto de classes compõem uma biblioteca de recursos dentro de uma linguagem de programação. A linguagem Java oferece e utiliza um recurso conhecido como *Application Programming Interfaces* (APIs), que são inúmeras bibliotecas prontas na solução de problemas em aplicações Java [14].

B. Interface gráfica com JavaFX

O JavaFX é um *framework* multimídia implementada em Java para o desenvolvimento de aplicações gráficas. As aplicações desenvolvidas em JavaFX podem ser utilizadas em vários dispositivos diferentes.

Esse *framework* utiliza a arquitetura modelo, visão e controle (MVC). Essa funcionalidade permite que o desenvolvedor desenvolva um código mais manutenível e robusto.

C. Bibliotecas OpenCV e JavaCV

A *Open Source Computer Vision Library* (OpenCV) é uma biblioteca de programação livre, disponível gratuitamente e desenvolvida nas linguagens C e C++ pela

Intel. Essa biblioteca implementa mais de 500 funções. Entre suas principais funcionalidades estão ferramentas de interpretação de imagens, análise de movimentos e reconhecimento de padrões [8].

Além disso, a OpenCV é vastamente usada para desenvolver aplicações que utilizam visão computacional [19]. Essa biblioteca fornece suporte para diversas linguagens, incluindo o Java.

A OpenCV é dividida em cinco grupos de funções: Processamento de imagens; Análise estrutural; Análise de movimento e rastreamento de objetos; Reconhecimento de padrões e Calibração de câmera e reconstrução 3D [8].

A JavaCV é uma biblioteca desenvolvida para permitir a comunicação entre OpenCV e projetos desenvolvidos em Java. Assim, essa biblioteca facilita a utilização do OpenCV por meio da linguagem Java.

D. Padrões Singleton e Observer

Os Padrões de projeto são técnicas elaboradas e aplicadas para solucionar problemas comuns em diversas aplicações. Os padrões de projeto *Singleton* e *Observer* foram aplicados para solucionar desafios recorrentes do componente proposto [10].

O padrão *Singleton* define que apenas uma única instância de determinada classe seja criada. Esse padrão garante a existência de um único objeto em todo o projeto e permite que esse objeto possa ser acessado de forma global na aplicação [10].

O padrão *Observer* determina uma dependência um para muitos entre objetos. Assim, quando um objeto muda de estado todos os dependentes são notificados e atualizados automaticamente. Portanto sua função é observar mudanças em um objeto e notificar todos outros objetos interessados nessas mudanças [10].

III. PROCESSAMENTO DIGITAL DE IMAGENS

O Processamento Digital de Imagens é uma técnica para a análise de dados multidimensionais. Essa técnica permite manipular e analisar imagens com objetivo de obter informações e/ou melhorar as características visuais da imagem [9].

No processo de manipulação de imagens o sistema recebe como entrada uma imagem. Esse sistema gera como saída um conjunto de valores numéricos que podem ou não compor uma outra imagem [9].

Os passos fundamentais para o processamento de uma digital imagem são: aquisição da imagem, pré-processamento, segmentação, representação e descrição, reconhecimento e interpretação. Todas essas etapas são ligadas a uma base de conhecimento [18].

A. Imagem Monocromática

Uma imagem digital é formada por uma matriz $N \times M$. Onde N representa o número de linhas e M o número de colunas. Nessa matriz cada elemento, denominado *pixel*, possui uma posição (i, j) . Além disso, esses pixels possuem uma intensidade de cor [9]. O Modelo de Representação de Cores tem a função de padronizar o uso das cores nos dispositivos [4].

O espaço de cor RGB é o padrão mais utilizado nos dispositivos atuais. Esse padrão é composto por três cores primárias: vermelho (R), verde (G) e azul (B). Cada um desses componentes possui um valor de intensidade, numericamente representadas por valores entre 0 e 255 [4].

Além das imagens que utilizam o espaço de cor RGB, existem as imagens monocromáticas. Essas imagens são comumente usadas no processamento de imagens. O processamento necessário para manipular essas imagens é menor, pois utiliza apenas um componente em escala cinza.

Nesse padrão de cores os valores dos pixels variam de 0, equivalente a ausência de cor, até 255, equivalente a maior intensidade de cor. Esses valores representam a intensidade luminosa em cada ponto da imagem. Essa intensidade é obtida por meio da equação (1).

Para efetuar a conversão de uma imagem RGB para monocromática é realizada primeiramente a multiplicação de cada componente por um fator padrão, seguido da soma dos resultados.

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

Onde:

- Y - Intensidade luminosa
- R - Intensidade de vermelho
- G - Intensidade de verde
- B - Intensidade de azul

B. Histograma

Em uma imagem monocromática vários pixels podem apresentar a mesma intensidade de luz, mesmo tom de cinza. O histograma de uma imagem é um conjunto de números indicando o percentual de pixels naquela imagem que apresentam um determinado nível de cinza [7]. O histograma permite obter informações sobre a qualidade da imagem, com relação ao nível de contraste e ao brilho médio.

C. Equalização de Histograma

O histograma mostra estatisticamente a distribuição dos pixels com relação ao contraste e níveis de iluminação da imagem [2]. A equalização de histograma é uma técnica usada para redistribuir os valores dos níveis de cinza dos pixels da imagem. O objetivo dessa distribuição é construir um histograma mais uniforme. [7].

Essa técnica permite aprimorar a qualidade da imagem, manipulando os níveis de contraste e brilho. Essa manipulação possibilita a visualização de detalhes úteis as demais fases do processamento da imagem.

D. Limiarização

Existem várias técnicas de segmentação de imagens. Uma dessas técnicas é a limiarização. O objetivo dessa técnica é facilitar a análise de uma imagem separando os objetos de interesse presentes na imagem [6].

A limiarização utiliza o histograma para segmentar o objeto de interesse de seu fundo na imagem. Para realizar esse processo o algoritmo de limiarização utiliza um valor de limiar cinza. Esse valor classifica os pixels da imagem como branco quando esses pixels possuírem valores acima ou igual ao limiar e preto quando o valor for menor [18].

O resultado gerado nesse processo é uma imagem binária. Uma imagem binária é representada por valores de zero para todos os pixels brancos e valores um para os pixels pretos. Portanto a técnica de limiarização é o processo binarização da imagem [18].

IV. O COMPONENTE

A função componente desenvolvido neste trabalho é reconhecer as piscadas realizadas pelo usuário, permitindo que o desenvolvedor possa definir o tipo de piscada, o tempo de cada uma e associar um comando computacional, efetivando a interação homem-máquina segundo as necessidades do sistema que fizer uso do componente.

A. Eyeblink

O componente criado reconhece a piscada de olhos utilizando a biblioteca Eyeblink, desenvolvida por [4], baseado no estudo proposto por [6]. O trabalho de [4] constitui a primeira etapa na elaboração do componente proposto neste trabalho.

A biblioteca Eyeblink utiliza câmeras de vídeo convencionais, como as que são padrão na maioria dos *notebooks*, para captura a imagem digital que será processada. Essas câmeras fornecem um mecanismo de captura de comandos por meio da interação usuário máquina utilizando o piscar de olhos.

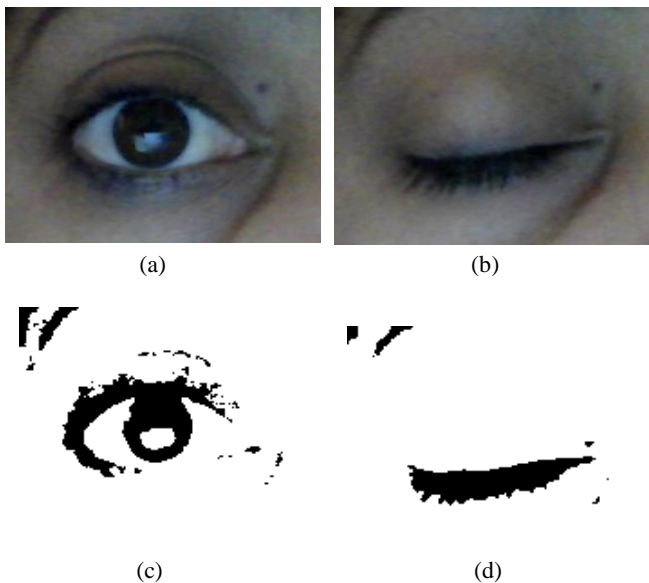


Fig. 1. Máscara do olho do usuário. (a) olho aberto; (b) olho fechado; (c) máscara do olho aberto; (d) máscara do olho fechado

A identificação da piscada é realizada pelo processamento de imagem usando algoritmos de reconhecimento de padrão e as bibliotecas OpenCV e JavaCV. As versões utilizadas dessas bibliotecas foram, respectivamente, a 2.4.10 e a 0.7.

O funcionamento da Eyeblink é realizado em duas etapas. A primeira parte é o processo de calibragem da câmera em relação ao usuário. Em seguida, é executado o processo de captura para detectar a piscada do olho.

O processo de calibragem da câmera acontece sempre que o componente for inicializado dentro da aplicação CAA a qual for incluído, pois a máscara do olho do usuário, como

mostra a Figura 1, é gerada nessa fase e será usada como referência no processo de identificação da piscada.

1) Calibragem da câmera

O processo de calibragem mostrado na Figura 2 inicia com a captura do frame. O frame é uma imagem adquirida pela câmera para realizar a detecção da face. A utilização da OpenCV garante que a imagem somente seja capturada se houver uma face detectada. Se existir mais de um rosto, a face que possuir maiores dimensões é utilizada na próxima etapa do processo.



Fig. 2. Fases do processo de calibragem. (a) captura do frame; (b) transformação em cinza; (c) Equalização do histograma

A Figura 2 mostra a etapa de transformação da imagem RGB em tons de cinza. Essa etapa converte o frame capturado em uma imagem monocromática. Além disso, é realizada a equalização do histograma gerado a partir da imagem monocromática. A segunda etapa é detecção da face.

A face é identificada pela biblioteca Eyeblink por meio do reconhecimento de padrão utilizando o algoritmo proposto por Viola-Jones [16]. A biblioteca OpenCV utiliza esse algoritmo combinado com uma melhoria proposta por Lienhart [17].

Ao final da segunda etapa é iniciado o processo de identificação da região de interesse. Esse processo é ilustrado pela Figura 3. Na Figura 3 nota-se que a região ocular é destacada da face do usuário. Finalmente é aplicada limiarização da imagem.



Fig. 3. Identificação da região de interesse. (a) recorte do rosto; (b) recorte do olho.

A qualidade e a nitidez da imagem binária do recorte do olho dependem do valor do limiar configurado nessa etapa. O valor do limiar pode variar de acordo com a iluminação do local e a cor da pele do usuário.

A Eyeblink permite que esse valor seja ajustado em tempo de execução para que seja obtido um conjunto de imagens binárias, composto por 10 frames obtidos na etapa seguinte a limiar.

Esse conjunto de imagens é armazenado e servirá para a montagem da máscara do olho, como mostra a Figura 4, criada com base nesses frames. Essa máscara é utilizada na fase seguinte que é identificação da piscada.

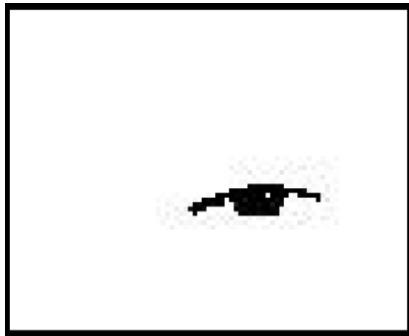


Fig. 4. Máscara do olho

2) Identificação da piscada

A identificação da piscada é o processo que detecta quando o olho está aberto ou fechado e retorna o intervalo de milissegundos da piscada. Entende-se o piscar dos olhos como o ciclo de estados do olho aberto, fechado e aberto novamente.

A fase de identificação é constituída pelas etapas capturar frame, transformar em cinza, equalizar histograma, detectar face, recortar rosto, recortar olho e limiarizar. Assim que essas etapas são concluídas o frame obtido nessa fase é comparado com a máscara gerada na fase de calibragem.

Nessa fase do processo o padrão *observer* é extremamente importante. Sempre que um novo frame do olho é capturado, o observador implementado notifica as demais partes interessadas que existe um novo frame a ser comparado.

A comparação do frame com a máscara inicial gera um percentual de igualdade. Esse percentual determinará se o olho está no estado aberto ou fechado. Após obter o percentual de igualdade é calculado o tempo, em milissegundos, da piscada.

B. Ambiente de funcionamento do componente

O ambiente para funcionamento da Eyeblik depende da correta instalação e configuração das bibliotecas OpenCV e JavaCV. O desenvolvedor que utiliza essas bibliotecas deve configurá-las para utilizar seus recursos em seus projetos.

De acordo com o manual de instalação da OpenCV [19] para o bom desempenho das funcionalidades da biblioteca é necessário sua configuração junto as variáveis de ambiente do sistema operacional e posterior inclusão nos sistemas a serem implementados.

Esse processo de configuração e inclusão é trabalhoso e demanda tempo. Entretanto o componente proposto realiza essas configurações e inclusões, automatizando essa etapa que exigiria maior trabalho e conhecimento por parte desenvolvedor.

C. Interface gráfica

O componente proposto contará com uma interface gráfica que está sendo desenvolvida para ser facilmente incluída da aplicação a qual o componente será associado.

Essa interface viabiliza e auxilia no momento de calibragem da câmera, onde o valor do limiar é definido. Como esse valor sofre influência da luz, por exemplo, a interface permite um *feedback* em tempo real de como será a máscara conforme o valor de limiar é incrementado ou decrementado. A Figura 5 mostra um protótipo da tela de interface.

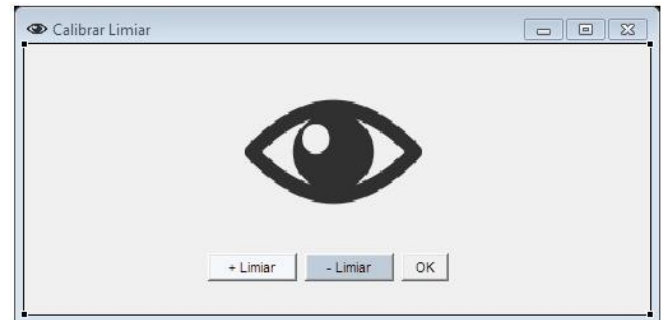


Fig. 5. Tela de calibragem da câmera

V. CONSIDERAÇÕES FINAIS

O presente trabalho atingiu o objetivo proposto que era a criação de um componente que fornecesse um mecanismo de entrada de dados baseada na piscada de olhos para aplicações CAA, sendo uma solução de baixo custo e mais acessível.

A criação desse componente é relevante pois fornece aos desenvolvedores uma ferramenta que permite a criação de suas aplicações CAA sem que seja necessário um aprofundamento em conhecimentos relacionados a processamento de imagens ou na configuração e utilização das bibliotecas usadas no presente estudo.

Em estudos futuros será desenvolvido a evolução do componente o uso das versões mais recentes das bibliotecas OpenCV e JavaCV. Para uma maior flexibilização do componente, será desenvolvido uma versão multiplataforma.

Além disso, será incorporado ao componente o *download* automático das bibliotecas utilizadas, para evitar incompatibilidades entre as versões e serão avaliadas técnicas para a autocalibração do componente.

REFERÊNCIAS

- [1] B. G. Avila, L. M. Pesserino, L. M. R. Tarouco. "Usabilidade em tecnologia assistiva: estudo de caso num sistema de comunicação alternativa para crianças com autismo". *Revista Latinoamericana de Tecnología Educativa*. Vol. 12, no. 2, pp. 115-129. Junho/Dezembro 2013.
- [2] D. C. Cavalieri. *Interface homem-máquina para uma cadeira de rodas baseada no movimento ocular e piscadas de olhos*. Universidade Federal do Espírito Santo, 2007.
- [3] P. Cipresso et al. "The use of P300-based BCIs in amyotrophic lateral sclerosis: from augmentative and alternative communications to cognitive assessment". *Brain Behavior*. Vol. 2, no. 4, pp. 479-498. Julho 2012. Acedido em 09 de Junho de 2016, em <http://www.ncbi.nlm.nih.gov/>.

- [4] C. V. S. Ferreira. *Biblioteca para detecção do piscar de olhos utilizando a linguagem de programação java*. Instituto Federal de Goiás, 2016.
- [5] R. S. Gomide. *Ambiente Virtual para reabilitação de membros superiores utilizando visão computacional*. Universidade Federal de Goiás, 2012.
- [6] R. S. Gomide et al. *Um estudo preliminar do protocolo de navegação de interfaces gráficas baseado na codificação de huffman*. Universidade Federal de Uberlândia, 2015.
- [7] A. Leonel et al. *Uma interface humano-máquina baseada no rastreamento ocular para comunicação escrita de pacientes com síndrome locked-in*. Universidade de Pernambuco, 2009.
- [8] M. Marengoni, D. Stringhini. “Tutorial: Introdução à visão computacional usando OpenCV”. *Revista de informática teórica e aplicada*. Vol. 16, no. 1. pp. 125-160. 2009. Acedido em 10 de Junho de 2016, em <http://seer.ufrgs.br/>
- [9] M. A. M. Silva, A. P. A. Chaves, F. J. A. Aquino. Projeto e desenvolvimento de uma ferramenta educativa para ensino de processamento de imagens baseado na biblioteca OpenCV. XL Congresso Brasileiro de Educação em Engenharia, Belém, 2012.
- [10] E. Gamma et al. *Padrões de Projetos: Soluções reutilizáveis de software orientado a objetos*. Bookman, 2007.
- [11] R. Colker. *Americans with disabilities act: A windfall for defendants*. The Harv. Cr-CIL Rev. 34 (1999) 99
- [12] P. Hohmann, M. R. Cassapian. *Adaptações de baixo custo*. Rev. Ter. Ocup. Univ. São Paulo, v. 22, n. 1, p. 10-18, jan./abr. 2011 .
- [13] C. Sik-Lányi, Á. Molnár-Lányi. *Psychological and pedagogic testing of handicapped children with locomotion disorder using multimedia programs*. Proc. 3rd Intl Conf. Disability, Virtual Reality & Assoc. Tech., Alghero, Italy 2000
- [14] P. Deitel, H. Deitel. *Java Como Programar*. Person, 2010.
- [15] L. B. Honorato, D. Milano. “Visao computacional”. Universidade Estadual de Campinas, p. 1–7, 2010. Acedido em 29 de Junho de 2016 , em <http://www.ft.unicamp.br>.
- [16] P. Viola, M. Jones. rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition (CVPR)*, v. 1, p. I–511–I–518, 2001. Acedido em 15 Março de 2016, em <http://www.cs.cmu.edu>
- [17] R. Lienhart, A. kuranov, V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Proceedings of the 25th DAGM Pattern Recognition Symposium*, p. 297–304, 2003. Acedido em 10 de Junho de 2016, em <http://www.springerlink.com>
- [18] O. M. Filho, H. V. Neto. *Processamento digital de imagens*. Brasport, 1999.
- [19] ITSEEZ FOUNDATION. *OpenCV*. Acedido em 29 de Junho de 2016, em <http://opencv.org>.