

# Switch KVM Através de Visão Computacional

César T. de Albuquerque<sup>1</sup>, Rafael de Santiago<sup>1</sup>, Benjamin G. Moreira<sup>2</sup>

<sup>1</sup> Centro de Ciências Tecnológicas da Terra e do Mar (CTTMar)  
Universidade do Vale do Itajaí (UNIVALI) – Itajaí, SC – Brasil

<sup>2</sup> Centro de Engenharias da Mobilidade  
Universidade Federal de Santa Catarina (UFSC) – Joinville, SC – Brasil

*cesar\_\_albuquerque@hotmail.com, rsantiago@univali.br,  
benjamin.grando@ufsc.br*

**Abstract.** *Software known as KVM switch are responsible for control two or more computers with the same mouse and keyboard. The purpose of this study is to explore face detection and use this to detect where is the attention of the user and changing the use of keyboard and mouse to the computer of attention, proposing an alternative in the form of Human-Computer Interaction. The method used for face detection was the method Viola-Jones using the OpenCV library and the feature was incorporated into a KVM switch called Synergy.*

**Resumo.** *Softwares que promovem o controle de dois ou mais computadores por intermédio de um conjunto de periféricos compostos por mouse e teclado são conhecidos como switch KVM. A proposta deste trabalho visa explorar a detecção da face do usuário e identificar perante qual tela o usuário se encontra, reproduzindo a ação de controlar dois ou mais computadores somente com um mouse e teclado para esse computador alvo da atenção, propondo um acréscimo na forma de Interação Humano-Computador. O método utilizado para a detecção da face foi o método Viola-Jones com apoio da biblioteca OpenCV e o recurso foi incorporado a um software de switch KVM chamado Synergy.*

## 1. Introdução

É por meio da interface que se pode ter um efetivo uso de uma aplicação, pois é através dela que o usuário comunica-se para realizar suas tarefas. Esta pode ser fonte de motivação e até, dependendo de suas características, uma ferramenta para o usuário. No entanto, se mal projetada, acaba tornando-se um ponto decisivo na rejeição de um sistema [Fole 1990 *apud* Oliveira 2004]. Portanto, espera-se que a interface seja sempre a mais fácil de entender e que forneça sequências simples e consistentes para sua utilização, deste modo, descartando qualquer ambiguidade ao usuário. Deve ser menos traumática e ao mesmo tempo mais interativa possível. Não menos importante, a interação deve passar despercebida para que o usuário possa se fixar somente na tarefa que deseja realizar utilizando o sistema [Oliveira 2004]. Atualmente um componente que vem sendo utilizado como elemento de interface com o computador é a webcam.

Segundo [Nixon e Aguado 2002], a imagem capturada pela webcam e processada pelo computador é uma abordagem caracterizada por visão computacional. Alguns trabalhos, como [Trigo 2010], comprovam a afirmação anterior, na qual utiliza-

se a visão computacional (implementada com uma webcam) como alternativa para Interação Humano-Computador (IHC). No caso do trabalho citado, este fornece ações de arrastar pastas, pressionar um botão virtual e controlar um personagem em um jogo, sendo a interação baseada em gestos das mãos utilizando conceitos da Visão Computacional (VC).

VC pode ser definida como um “conjunto de métodos e técnicas através dos quais sistemas computacionais são capazes de interpretar imagens” [Wangenheim e Comunello 2005].

Esse trabalho aplicou VC a uma aplicação de software de Switch KVM, que em sua essência é utilizado para controlar dois ou mais computadores através de um único mouse e teclado. A aplicação utiliza a webcam, geralmente localizada sobre os monitores, para captura da imagem do usuário e com o processamento das imagens através de técnicas de VC, reconhecer em qual computador o usuário pretende fazer uso e passar o controle dos periféricos para este computador. Para fins experimentais do recurso, todo o trabalho foi desenvolvido em um cenário fixo e favorável à aplicação.

No design de sistemas interativos os projetistas farão uso cada vez mais de tecnologias e estas irão além dos sistemas baseados em tela. Designers desenvolverão experiências multimídia usando uma variedade de modalidades como som, imagem e toque e estas serão combinadas de novas maneiras. Existe uma ampla gama de pessoas envolvidas no projeto e desenvolvimento de sistemas interativos, os quais redesenham sistemas para tirar vantagem do desenvolvimento de tecnologias e acrescentar características adicionais aos sistemas [Benyon 2011].

Sendo assim é importante propor uma solução que identifique uma necessidade do usuário a partir de sua interação com o computador, além de mostrar possibilidades novas e diferentes para que designers possam projetar novas interações.

## **2. Visão Computacional**

VC é descrita como uma ciência que estuda como o computador enxerga o meio à sua volta, ou seja, a visão de uma máquina, na qual suas informações são obtidas através de imagens capturadas com periféricos conectados ao computador, como câmeras de vídeo, roupas com sensores e scanners. Com estes é possível que o computador reconheça e manipule os objetos que compõem uma imagem [Ballard e Brown 1982].

A VC é utilizada para resolver problemas específicos na qual essas aplicações são intituladas como sistemas especialistas. Sendo assim não existe um modelo definido como padrão para implementação dessas aplicações, mas estas utilizam do reconhecimento de objetos em imagens capturadas e interpretadas por um sistema especialista [Milano e Honorato 2010].

Utilizar VC requer uma entrada de dados específica e para isto há mecanismos como roupas compostas por inúmeros eletrodos que captam gestos e ações, mas elas escondem a naturalidade do movimento e requerem um processo de calibração para permitir medições precisas [Rehem e Trindade 2009] [Trigo 2010]. Outro componente que vem sendo utilizado como entrada de dados é a imagem de câmeras, que em computadores costuma ser a webcam. As imagens captadas por câmeras precisam ser processadas e transformadas em informações e esse processo depende de métodos

contidos em bibliotecas de processamento gráfico, como o OpenCV (Open Source Computer Vision) e o OpenGL (Open Graphics Library) [Rehem e Trindade 2009].

## 2.1 Face Detection

Pesquisas vêm sendo aplicadas envolvendo imagens de faces humanas utilizando a VC e baseadas em IHC, incluindo *face recognition* (reconhecimento da face), *face tracking* (rastreamento da face) e reconhecimento de expressão facial. Para estes ocorrerem, o primeiro passo é realizar a *face detection* (detecção de face) [Yang e Ahuja 2001].

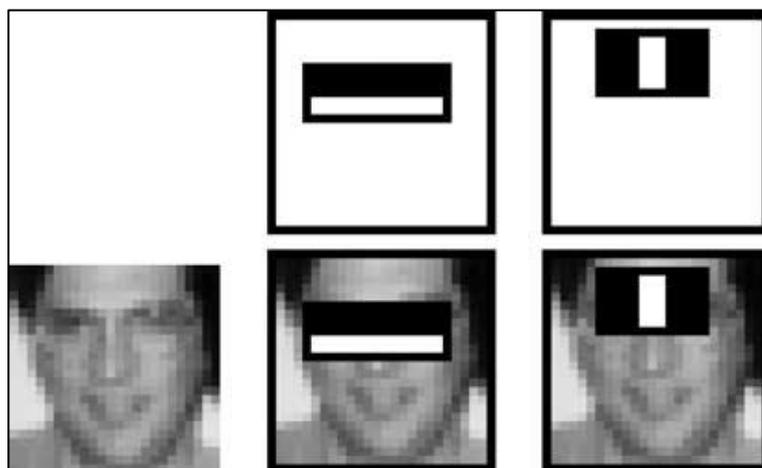
*Face detection* tem como objetivo determinar a posição da imagem de um ou mais rostos [Yang *et al.* 2002]. Segundo Kuehlkamp (2013), não existem rostos idênticos, podendo variar em formato, cor, expressão, textura. Para que a *face detection* ocorra, alguns desafios devem ser superados, onde estes prejudicam a detecção da face.

Apesar dos empecilhos listados por [Scandaroli e Melo 2009], é possível através de uma única imagem ou uma sequência delas, identificar e localizar rostos humanos, independentemente de suas posições, escalas, orientações, poses e condições de iluminação [Yang e Ahuja 2001]. [Yang *et al.* 2002] definem quatro técnicas de detecção de rostos a partir de uma imagem com uma única cor ou intensidade:

1. *Knowledge-based methods* (métodos baseados em conhecimento): estes são baseados em regras para identificar um típico rosto humano. O método leva em consideração as características faciais;
2. *Feature invariant approaches* (abordagem por características invariantes): consiste em procurar por características estruturais mesmo sobre poses ou sobre variações de iluminação;
3. *Template matching methods* (métodos baseados em modelos): vários padrões de face armazenados, descrevendo a face como um todo ou características faciais separadamente. Dentre os padrões armazenados, é feita uma correlação com a imagem de entrada; e
4. *Appearance-based methods* (métodos baseados em aparência): em contraste com os modelos existentes, ocorre um aprendizado a partir de um conjunto de formações de imagem que capturam a variabilidade da aparência facial, como cores, textura e formas presentes na imagem. Os modelos aprendidos são utilizados para detecção das próximas faces, requerendo pouco recurso de processamento, facilitando o cálculo das invariantes rotação e escala. Este método utiliza classificadores como em redes neurais.

Um dos métodos baseados em aparência que aplica o conceito de AdaBoost ou Boosting (impulsionar) é o Viola-Jones, assim comumente chamado, pois foi desenvolvido por Paul Viola e Michael Jones. O AdaBoost (*adaptive boosting*) combina o processamento de imagens com inteligência artificial. O algoritmo de aprendizado, no Viola-Jones, consiste no treinamento de vários classificadores fracos em série, obtendo assim um classificador forte. Cada vez que estes classificadores fracos são treinados, mais discriminantes os classificadores ficam, contribuindo com a mescla dos resultados [Kuehlkamp 2013] [Scandaroli e Melo 2009] [Yang *et al.* 2002].

Conforme [Scandaroli e Melo 2009], as características das regiões retangulares descritas servem de gabaritos para medir a diferença de intensidade destas regiões. No caso da face são utilizadas as características de Haar, onde através do gabarito é medida a diferença de intensidade entre olhos e as bochechas, assim como a diferença de intensidade entre os olhos e a região do nariz. A Figura 1 apresenta estas características.



**Figura 1. Representação geométrica das características de Haar**

**Fonte: Viola e Jones (2003).**

Com o aumento da tecnologia, métodos mais eficazes e amigáveis para IHC estão sendo desenvolvidos, que não dependem de dispositivos tradicionais como teclados, mouses e monitores. Além disso, a relação preço/desempenho cada vez menor da computação juntamente com reduções recentes no custo de aquisição de imagem de vídeo significa que sistemas de VC podem ser implantados em sistemas desktop e em sistemas embarcados. Pesquisas em expansão sobre o processamento de imagens voltadas à face humana é baseada na premissa de que as informações sobre um usuário podem ser extraídas através desta imagem e assim o computador reagir conforme o reconhecimento dos padrões estipulados [Yang e Ahuja 2001].

## **2.2 OpenCV**

A OpenCV (Open Source Computer Vision) é uma biblioteca de código aberto para processamento gráfico desenvolvida inicialmente pela Intel e idealizada com o objetivo de tornar a VC acessível a usuários. A biblioteca possui os executáveis (binários) otimizados para os processadores Intel, que são detectados através de DLL (Dynamic Linked Library). A OpenCV também dispõe de uma biblioteca IPL (Image Processing Library), documentação e alguns códigos exemplos para ajudar no entendimento do funcionamento da biblioteca e implementação da mesma [Laganière 2011].

Desde 1999, ano do início do projeto de desenvolvimento da OpenCV liderado por Gary Bradski, esta biblioteca tem sido adotada como importante ferramenta de desenvolvimento e utilizada por pesquisadores e desenvolvedores de VC. Sua versão 1.0 foi lançada em 2006, evoluindo consideravelmente após o lançamento da OpenCV 2 em 2009 [Laganière 2011].

Na biblioteca OpenCV há 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 [Marengoni e Stringhini 2009].

### 3. KVM

Switch KVM ou somente KVM descreve as iniciais para Keyboard, Vídeo e Mouse que se trata de um adaptador físico ou software para ligar e controlar dois ou mais computadores no mesmo teclado, mouse e monitor. O KVM implicitamente gera maior economia a seus utilizadores, por não necessitar da aquisição de vários conjuntos de periféricos e requerer de apenas um conjunto destes para controlar dois ou mais computadores, além de ser ideal para empresas e usuário domésticos que apresentem falta de espaço em sua área de trabalho física [SHARE MOUSE 2013].

Geralmente softwares que proporcionam KVM como Synergy [SYNERGY 2013] e Share Mouse [SHARE MOUSE 2013], não necessitam de botões ou teclas do teclado para selecionar o computador a ser controlado já que os mesmos devem estar “conectados” de alguma forma, seja ela utilizando o padrão Ethernet, Gigabit, ou conexão sem fio. Em sua totalidade, os softwares que promovem KVM não transmitem conteúdo do monitor através da rede, mas apenas repercutem os movimentos de mouse e teclado. Sendo assim, os computadores devem estar ao alcance e visíveis ao usuário para que seja possível visualizar as ações de mouse e teclado ocorrendo em outro computador. Uma metodologia de funcionamento é baseada quando o usuário ultrapassa o limite do monitor com o cursor do mouse e a partir desta o software se encarrega em enviar ações do mouse e teclado ao outro computador conectado [SHARE MOUSE 2013].

Esse trabalho apresenta uma adição ao método de troca de dispositivo utilizado pelo software Synergy, permitindo que o controle dos periféricos passe para o computador que o usuário está olhando.

O Synergy é um sistema gratuito, de código aberto e desenvolvido na linguagem C++. A versão do Synergy utilizada nesse trabalho foi a versão 1.4.12 Beta, que suporta os principais sistemas operacionais utilizados, entre eles o Windows, Mac OS X, Ubuntu/Debian e Fedora/Red Hat [SYNERGY 2013].

### 4. Projeto

Esse projeto desenvolveu um sistema de *face detection* utilizando a biblioteca de visão computacional OpenCV. A partir de imagens capturadas pela *webcam*, o sistema detecta a face do usuário a fim de distinguir se este está diante da tela do computador. Este *face detection* foi integrado ao sistema Synergy, que se trata de um software de *switch* KVM. O sistema tem como restrição atender somente dois computadores que devem estar interconectados pela rede LAN, e fazer com que estes compartilhem do mesmo mouse e teclado.

O *face detection* foi obtido através do classificador de objetos nomeado de Viola-Jones e já integrado à biblioteca OpenCV com o método chamado *Classificador Haar*. Este mede a diferença de intensidade entre os olhos e as bochechas e posteriormente a diferença de intensidade entre os olhos e a região do nariz entre eles.

Em um primeiro momento a técnica utilizada para descobrir para qual computador o usuário estava olhando foi a detecção de olho (*eye detection*) que depois foi substituída pela *face detection*, conforme justificado a seguir.

Como cenário experimental, o trabalho previu o chaveamento de apenas dois computadores. Cada computador deveria possuir uma webcam para determinar qual dos computadores o usuário está olhando. Com a implementação do *eye detection* primeiramente no computador servidor, verificou-se que o uso de duas webcams não era necessário, pois apenas o servidor detectando os olhos do usuário se mostrou suficiente para identificar que a atenção está direcionada para o mesmo. Caso o usuário mova sua cabeça na horizontal observando o outro monitor o *eye detection* já não conseguiu classificar um rosto humano na imagem. Isto se deve a classificadores utilizados no método aplicado de *haar-like features*. Com este movimento da face do usuário, estes classificadores não reconhecem as características do rosto humano sobre a imagem capturada pela webcam do servidor, acarretando na troca de controle de mouse e teclado para o computador cliente.

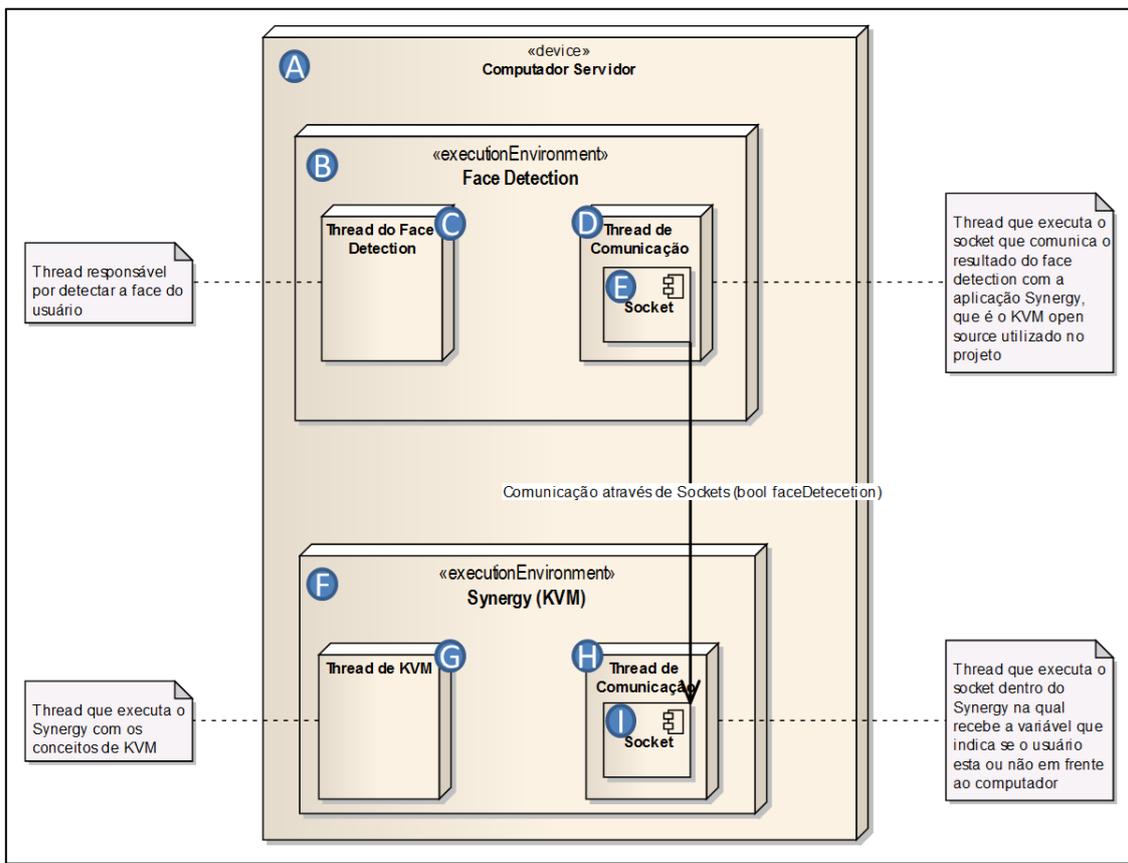
Após o desenvolvimento, percebeu-se que o *eye detection* apresentou um baixo desempenho, defasando em até 5 segundos uma ação realizada pelo usuário. Além deste problema, o algoritmo de *eye detection* propõe primeiramente a busca da face do usuário e posteriormente a busca dos olhos sobre a face obtida no passo antecessor. O dado que o *eye detection* representa acaba sendo redundante, uma vez que já se descobriu o rosto e é passível de assumir que os olhos estão compreendidos naquele mesmo. A partir desta compreensão, para viabilizar o experimento, foi tomada a decisão de buscar apenas a face do usuário (*face detection*).

Para atender as necessidades deste projeto, a arquitetura descrita foi especificada para um dos dois computadores, a que rodará a versão servidor no contexto do Synergy, consequentemente, o cliente continua com o sistema original. A

Figura 3 auxilia na explicação da arquitetura do servidor, onde as letras ajudam a identificar cada recurso utilizado, assim como sua função.

O computador servidor (A) contempla os sistemas KVM Synergy (F) e o *face detection* (B) que se comunicam através de *sockets*, uma vez que os dois foram desenvolvidos separadamente. Os *sockets* (E) e (I) comunicam o *face detection* com o Synergy repassando um parâmetro para identificar se a imagem capturada pela *webcam* percebe as características do rosto humano. Para que o *face detection* não deixe de detectar o rosto do usuário nos momentos de envio do parâmetro para Synergy, se fez necessário a utilização de *threads*.

Os *threads* (D) e (H) fazem com que os *sockets* comuniquem ininterruptamente entre as duas aplicações. Mas ainda desempenhando papel principal, o *face detection* possui um *thread* (C) para processar a imagem capturada pela *webcam* a fim de estabelecer qual dos computadores o usuário está observando. Já o *thread* (G) estabelece a troca de periféricos entre servidor e cliente conforme o parâmetro recebido por a (H). Desta forma, o Synergy utiliza *threads* para comunicar com o *face detection* e continuar a responder quanto aos controles entre mouse e teclado.



**Figura 2. Resultado da arquitetura do servidor de KVM**

Prevendo um cenário onde o monitor do servidor é ligeiramente menor que 14 polegadas, faz com que o movimento realizado pelo usuário para visualizar o computador cliente seja pequeno, podendo fazer que o *face detection* permaneça detectando a face do usuário mesmo quando ele já esteja olhando para o outro monitor. Neste caso é necessário que o usuário aumente a distância entre os computadores para que o movimento da cabeça faça com que a face não seja mais passível de ser visualizada pelo computador servidor.

Idealizando o cenário, a

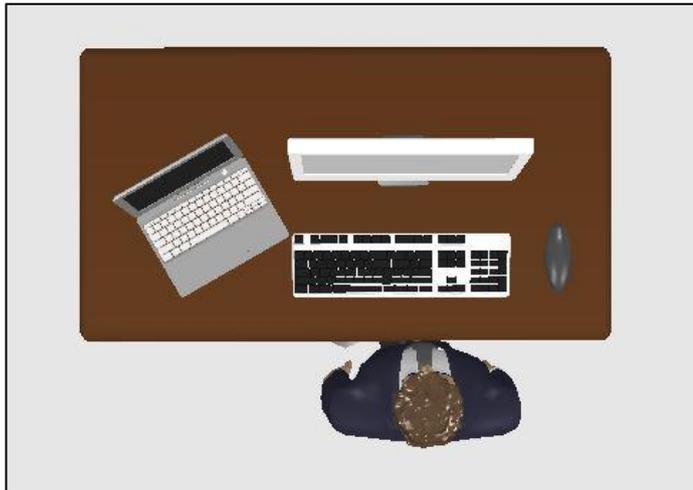
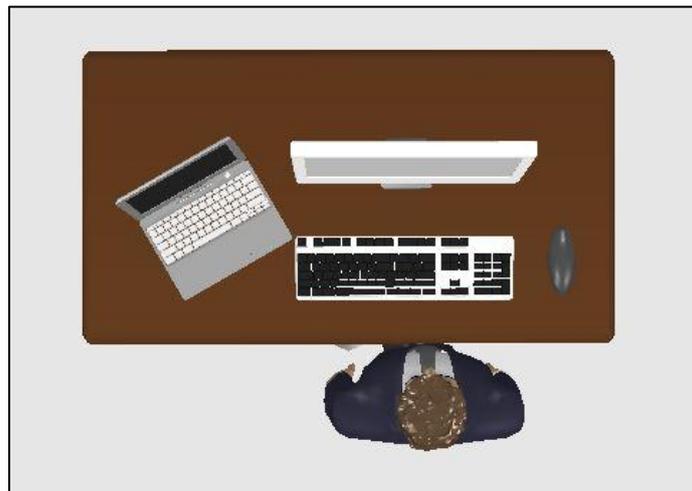


Figura 3 proporciona uma imagem vista de cima em que o usuário está em frente ao computador que possui a versão servidor do software e, conseqüentemente, o mouse e teclado conectados fisicamente ao mesmo. Além disso, a distância entre os dois computadores deve ser medida através do campo de visão da *webcam*, que foi estabelecida em 40 centímetros de distância entre o centro do monitor, onde se espera que a *webcam* esteja, até um dos lados do monitor cliente.



**Figura 3. Cenário ideal para utilização do sistema**

#### **4.1 Testes**

Para testes da ferramenta desenvolvida, esse trabalho entrevistou 13 pessoas não envolvidas no projeto, onde primeiramente foi fornecida uma instrução sobre o uso aos entrevistados. Entrevistas e questionários são utilizados para pesquisas em ciências sociais, de mercado e em IHC. Conforme sugerido por [Preece 2005] uma entrevista estruturada ou questionário são as melhores opções para este intuito, que é obter o *feedback* sobre uma característica particular de design.

Os avaliadores, na sua maioria homens entre 19 e 25 anos, navegaram em sites de futebol em um dos computadores e acessaram notícias no outro. Alguns defenderam a ideia de utilizar a ferramenta para um fim específico e próprio, traduzindo que o

sistema se tornasse base de trabalho para quem necessita utilizar dois computadores constantemente.

Todos os utilizadores conseguiram alternar de controle entre os computadores, mas duas pessoas indicaram que a troca não ocorreu no mesmo instante, o que indica que o desempenho é um elemento primordial a ser trabalhado em aplicações desse tipo.

## 5. Conclusão

Este projeto explorou o uso de diversas estações de trabalho paralelamente. Neste ambiente, cada computador requer de um mouse e teclado. O KVM propõe solucionar este problema, mas geralmente o chaveamento entre os computadores é acionado pelo deslocamento destes periféricos, não sendo identificado nenhum outro KVM em que a “troca” aconteça com a detecção da face do usuário para determinar qual dos computadores deve possuir o foco dos periféricos, sendo essa a motivação para a realização desse trabalho.

Sistemas *open source* como o KVM Synergy, proporcionam que o próprio usuário obtenha o código fonte para personalizar e realizar melhorias no sistema fazendo com que ele se tornasse base para as funcionalidades estabelecidas. Este trabalho utilizou da integração de duas tecnologias: a Visão Computacional e o conceito de *switch* KVM.

Inicialmente, a intenção era utilizar um *eye tracking* em conjunto com o Synergy. No entanto o cenário proposto torna a detecção de face suficiente para o experimento pretendido e o custo computacional necessário para detectar os olhos do usuário é maior.

Ajustes ainda podem ser realizados a fim de melhorar o sistema desenvolvido a partir dos testes observados e apontados durante o experimento. Uma das observações pode ser contornada com detecção da face do usuário por parte do computador cliente, já que atualmente sempre que o usuário observa o teclado para reconhecer qual das teclas pressionar o mesmo deixa seu rosto à mostra fazendo com que foco fique no servidor. Além disso, ter uma webcam em cada cliente permitirá utilizar mais do que os dois computadores propostos.

Este trabalho se ateve ao escopo de atender somente dois computadores, mas é totalmente possível que melhorias proporcionem um número maior de computadores. Achando uma solução consciente é possível que o número clientes conectados ao servidor seja imprescindível.

## Referências Bibliográficas

Ballard, D. H. e Brown, C. M. (1982), Computer Vision, Prentice-Hall.

Benyon, D. (2011), Interação Humano-Computador, Pearson Prentice Hall.

Kuehlkamp, A. (2013), Ferramenta de baixo custo para gaze tracking baseado em imagens, Dissertação do Mestrado de Computação Aplicada da Universidade do Vale do Itajaí, Itajaí.

Laganière, R. (2011), OpenCV 2 computer vision application programming cookbook, Packt Publishing.

- Marengoni, M. e Stringhini, D. (2009), Tutorial: introdução à visão computacional usando OpenCV, Revista de Informática Teórica e Aplicada, volume 16.
- Milano, D. e Honorato, L. B. (2010) “Visão computacional”, [http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010\\_IA\\_FT\\_UNICAMP\\_visaoComputacional.pdf](http://www.ft.unicamp.br/liag/wp/monografias/monografias/2010_IA_FT_UNICAMP_visaoComputacional.pdf), novembro.
- Nixon, M. e Aguado, A. S. (2002), Feature extraction & image processing, Elsevier.
- Oliveira, A. N. (2004), Interação humano computador: modelagem e gerência de interfaces com o usuário, VisualBooks.
- Preece, J., et al. (2005), Design de interação: além da interação homem-computador, Bookman.
- Rehem, A. e Trindade, F. H. V. (2009) “Técnicas de visão computacional para rastreamento de olhar em Vídeos”, [http://almerindo.devin.com.br/index.php?option=com\\_content&view=article&id=78%3Atecnicas-de-computacao-visual-para-rastreamento-de-olhar-em-videos&catid=43%3Atrabalhos-de-alunos&Itemid=86&showall=1](http://almerindo.devin.com.br/index.php?option=com_content&view=article&id=78%3Atecnicas-de-computacao-visual-para-rastreamento-de-olhar-em-videos&catid=43%3Atrabalhos-de-alunos&Itemid=86&showall=1), abril.
- Scandaroli, T. G. e Melo, D. D. (2009), Detecção de usuário e estimação de posição para interface com realidade virtual, Monografia do Bacharelado em Engenharia Elétrica da Universidade de Brasília, Brasília.
- Share Mouse (2013) “Share your mouse and keyboard”, <http://www.keyboard-and-mouse-sharing.com/index.html>, novembro.
- Synergy (2013) “Bolton software”, <http://synergy-foss.org/pt-br/>, novembro.
- Trigo, T. R. (2010), Classificador de gestos das mãos baseado em imagens para aplicação em interface, Dissertação do Mestrado em informática do Instituto Tecnológico de Aeronáutica, São José dos Campos.
- Viola, P. e Jones, M. J. (2003), Robust real-time face detection, Kluwer.
- Wangenheim, A. V. e Comunello, E. (2005) “Visão computacional”, In: Seminário Introdução à Visão computacional, The Cyclops Project, PPGCC-INE-UFSC.
- Yang, M. H., *et al.* (2002) “Detecting faces in images: a survey”, In: IEEE: Transactions on pattern analysis and machine intelligence, volume 24.
- Yang, M. H. e Ahuja, N. (2001), Face Detection and Gesture Recognition for Human-computer Interaction, Kluwer.
- Yilmaz, A., Javed, O. e Shah, M. (2006) “Object tracking: a survey”, In: ACM Computer Survey, volume 38.