APLICAÇÃO DO RECONHECIMENTO DE GESTOS UTILIZANDO KINECT COMO CONTROLE DE TV

Jonathan Simon Prates, Átila Bohlke Vasconcelos

Faculdade de Informática – Centro Universitário Ritter dos Reis
CEP 90.840-440 – Porto Alegre – RS – Brasil

jonathan.simonprates@gmail.com, atila_vasconcelos@uniritter.edu.br

Abstract. *The objective of this work is to study the user's adaptation with a method of interaction with televisions through gesture recognition, as a substitute for the remote control. With the advent of new gesture recognizing devices with low cost depth sensors and a wide variety of software libraries available, researches like this one are becoming easily possible. This work is about a computer simulated environment, using an application and the Kinect sensor and together they are able to recognize the user behavior and send to the television the most appropriate command according to the gesture. Based on results, we realize that the mode of interaction gestures should not be the only method of interaction available due environment and users limitations.*

Resumo. *O objetivo deste trabalho é estudar a adaptação dos usuários com um modo de interação com televisores através do reconhecimento de gestos, substituindo o controle remoto. Com o advento de dispositivos de reconhecimento de gestos com sensores de profundidade de baixo custo e uma vasta gama de bibliotecas de programação disponíveis, pesquisas como esta estão se tornando possível mais facilmente. Este trabalho trata-se de um ambiente simulado por computador, utilizando uma aplicação e o sensor Kinect, que juntos são capazes de reconhecer o comportamento do usuário e enviar ao televisor o comando mais apropriado de acordo como o gesto reconhecido. Com base nos testes realizados, vemos que o modo de interação por gestos não deve ser o único método de interação disponível devido a limitações de ambiente e dos próprios usuários.*

1. Introdução

As inovações tecnológicas são protagonistas de grandes avanços e mudanças no cotidiano da sociedade [Tapscott e Caston 1995]. Neste cenário, o Reconhecimento de Gestos é um candidato para substituir os atuais métodos de interação com o usuário, como o controle remoto em videogames e televisores. Em relação à TV, provavelmente muitos usuários estão satisfeitos com o atual controle remoto, pois o consideram simples e útil. Assim, as pessoas poderiam achar complicado fazer tarefas rotineiras como passar o "controle" da televisão para outra pessoa através de Reconhecimento de Gestos.

Usar o Reconhecimento de Gestos ao controlar a televisão pode ter um impacto positivo ou negativo. Conhecer a opinião dos usuários utilizando gestos ao invés do controle remoto para comandar a TV foi a motivação ao propor este trabalho. Para isso, foi desenvolvido um *software* para a captura e processamento de dados através de técnicas de Reconhecimento de Gestos utilizando o Kinect. Esta aplicação reconhece o gesto do usuário e envia ao televisor o comando adequado. O presente trabalho está

dividido em cinco partes: esta introdução, referencial teórico, trabalhos relacionados, solução proposta e conclusão.

2. Referencial Teórico

Esta sessão apresenta os principais conceitos sobre usabilidade, interfaces e visão computacional aplicada a este trabalho.

2.1. Usabilidade e tecnologia

A usabilidade é definida pela apresentação das seguintes características: facilidade de aprendizagem, rápida execução de tarefas, baixo número de erros, satisfação e facilidade de lembrar como realizar uma tarefa após algum tempo [Hix e Hartson 1993]. Logo, quanto mais simples de fazer determinada tarefa, mais pessoas podem executá-la.

Por outro lado, a busca para simplificar as tarefas do dia-a-dia em computadores e celulares ou tecnologias como cartões ou controles remotos trazem grande complexidade aos seus desenvolvedores. Isso nos mostra um crescente paradoxo, onde pessoas fazem tarefas mais complexas de forma cada vez mais simples [Manfredo 2011]. O controle remoto de uma televisão pode ser citado como exemplo, pois possui botões para inúmeras funcionalidades em apenas um clique.

Cabe ressaltar que com a popularização dos sistemas interativos, o que se percebe por parte do usuário é uma tendência a optar pelo produto onde se destaque a usabilidade e funcionalidade. Sobretudo, a satisfação com a interface acaba sendo um aspecto decisivo na escolha do usuário diante de produtos semelhantes. Isso faz com que o sucesso competitivo do produto dependa diretamente de uma interface atrativa, ou seja, a que melhor permitir o acesso às funcionalidades do sistema para aquele usuário [Ascencio 1999].

2.2. O usuário e o controle remoto

Seguramente, o surgimento do controle remoto pode ser visto como um grande marco na história da televisão. Esse acontecimento trouxe mais conforto aos telespectadores, além de permitir, com agilidade, a escolha do conteúdo. Porém, mesmo com a evolução deste instrumento, que em certo tempo já possuiu até mesmo um fio, ainda existem problemas no controle remoto quando se relacionam os aspectos funcionalidade e usabilidade. O que acontece é que os controles remotos têm cada vez mais botões, o que proporcionalmente aumenta a complexidade na sua utilização, levando ao não aproveitamento de suas potencialidades definidas em fábrica [Martins Jr. e Pimentel 2011].

O controle remoto, supracitado no item 2.1 como exemplo de tecnologia que objetiva simplificar a vida do usuário, tem, entretanto, aspectos importantes a serem considerados no que diz respeito à usabilidade. Um dos pontos críticos é o excesso de funcionalidades implementadas em controles remotos, e até mesmo a forma (não clara) como elas estão identificadas, que acabam confundindo o usuário. Isso faz com que a grande maioria dos botões não seja regularmente utilizada [Nielsen 2004].

Diante desta gama de opções em um controle remoto, porém com um índice muito baixo de utilização e praticidade ao usuário, temos o que se pode definir como o “paradoxo da escolha”. Isso sugestiona que a maior diversidade de opções, diferentemente do que se supõe, não estaria sendo bem-sucedida, pois exige do indivíduo mais esforço, além de tornar os equívocos mais prováveis, o que leva a sua insatisfação [Schwartz 2007].

2.3. Evolução: tecnológica e interatividade

A tecnologia está a serviço da sociedade, onde o intuito é melhorar o cotidiano das pessoas. No ponto de vista da evolução tecnológica, se uma tecnologia supera a outra, ela é vista como melhor. Mas se pensadas sob a perspectiva social, as novas tecnologias muitas vezes acabam trazendo problemas que outras tecnologias anteriores não tinham ou já tinham resolvido [Montez e Becker 2004].

Os veículos de comunicação como um todo - e aqui se destaca a televisão, acompanham a evolução tecnológica. Neste contexto, a televisão está em constante processo de desenvolvimento tecnológico e, desta forma, busca se adaptar às necessidades da sociedade. E a interatividade surge neste processo evolutivo, embora a sua relação com a televisão ainda não esteja clara [Montez e Becker 2004].

O termo interatividade é recente, tendo várias definições em diversas áreas, mas que, generalizando-se, pode ser definido como "a relação entre dois ou mais agentes resultando num determinado efeito". Ainda é possível destacar a utilização do termo com o propósito de facilitar a comercialização e venda de produtos, no que se chama de "indústria da interatividade" [Barreto 2011].

Em geral, os usuários tendem a escolher produtos que permitam personalização, possibilitando uma maior identificação e aumentando sua interação com este produto [Bonsiepe 1997]. Sobretudo, o principal desafio da TV interativa é mudar os hábitos dos telespectadores passivos para que se tornem participativos [Souto Maior 2002].

2.4. Mudança de paradigma no cotidiano

Pode-se dizer que "a mudança de paradigma é fundamentalmente uma nova maneira de ver alguma coisa. A mudança de paradigma é frequentemente exigida em função de novos desenvolvimentos ocorridos em ciência, tecnologia, arte, ou outras áreas de atuação" [Tapscott e Caston 1995].

De certa forma, o aprendizado de novos gestos pelas pessoas é fácil e intuitivo, afinal, gestos são naturais para os humanos. Assim, é necessário pouco tempo de treinamento para que as pessoas possam usar de forma consistente os novos gestos e assim comunicarem-se com dispositivos ou outras pessoas [Wolf e Morril-Samuels 1987].

Novos conceitos e aplicações estão surgindo a partir de novas interfaces, baseadas em *touchscreen* e Reconhecimento de Gestos. Estas novas interfaces podem ser aplicadas com o objetivo de ajudar o usuário a interagir com os computadores e robôs e, possivelmente, no futuro, fazer parte do cotidiano.

2.5. Reconhecimento de gestos

A área de estudo que utiliza uma gama de técnicas de análise e processamento de imagem com o objetivo de fazer com que o computador entenda o gesto chama-se Reconhecimento de Gestos. Na maioria dos casos estas imagens são capturadas por um dispositivo como uma câmera e enviadas a um computador e podem ser usadas para controlar um dispositivo [Rehm *et al.* 2007]. Este é um assunto complexo dentro da área de Visão Computacional e inclui reconhecimento de padrões e aprendizagem do computador [Togores 2011].

2.6. Visão computacional

A Visão Computacional é entendida como a ciência e tecnologia em que máquinas, computadores e dispositivos enxergam através de imagens capturadas. Nas últimas

décadas, observou-se um crescente desenvolvimento sobre a análise de movimento por Visão Computacional, principalmente quando se trata do movimento do corpo humano [Crowley e Christensen 1993]. As principais tarefas da Visão Computacional são separadas em reconhecimento, análise de movimento, reconstrução de cena e restauração da imagem [Gismero 2012].

Sobre o reconhecimento, detectar objetos é um problema clássico em Visão Computacional. Existem diversas abordagens que se destacam para resolver este problema sem esforço humano, porém, em casos arbitrários como sombras e oclusões nas imagens, ainda podem ocorrer problemas [Rosenfeld 1999].

A segunda grande área da Visão Computacional é a análise de movimento. Ela geralmente é constituída por duas etapas: a detecção e o processamento. A análise do movimento depende da captura dos dados de cada movimento e de cada objeto reconhecido. Muitas tarefas têm relação com uma estimativa do movimento baseada na captura dos pontos gerados por uma sequência de imagem [Pinho *et al.* 2005].

O processamento de imagens é o processo onde, dada uma imagem, a saída é um conjunto numéricos de valores, podendo ou não compor outra imagem. A entrada muitas vezes precisa ainda ser filtrada para remover ruídos oriundos do processo de aquisição.

O objetivo da reconstrução de cena é criar um modelo tridimensional (3D) da cena, dado uma sequência de imagens, podendo construir desde uma cena com somente pontos bidimensionais ou até mesmo com cores e texturas [Rosenfeld 1999]. A restauração de imagens tem como objetivo a remoção de ruídos [Pinho *et al.* 2005].

2.7. Interação humano-computador e o reconhecimento de gestos

A Interação Humano-Computador (IHC), é definida como a ciência que estuda a interação entre pessoas e computadores [Stone 2005]. Observa-se que cada dia mais pessoas, até mesmo crianças, conseguem entender e obter resultados satisfatórios ao interagir com um dispositivo como um *tablet* ou um computador. A evolução dos estudos na área de IHC teve um fator fundamental para o avanço desta tecnologia. Na década de 70, quando os primeiros computadores pessoais foram lançados, somente pessoas com alto nível de conhecimento conseguiam compreender tal interação [Walker 1990].

Uma das formas de Interação Humano-Computador por meio do Reconhecimento de Gestos se dá através da aquisição de imagens. Entretanto, a tarefa de reconhecimento de objetos pode ser muito complexa, tendo em conta a diversidade de ambientes em que os objetos podem ser encontrados [Rosenfeld 1999].

O Kinect, assim como outros dispositivos de controle por gestos, possui vários problemas relacionados à interação com o usuário e suas aplicações. As principais dificuldades são a falta da definição de padrões das interfaces de gestos e a dependência da memória do usuário para executar uma ação, já que as instruções não estão sempre visíveis [Nilsen e Norman 2010].

2.7.1 Avaliação de interface

A avaliação de interface é importante a fim de medir a qualidade do software. GOMS (Goals, Operators, Methods e Selection Rules) é um método de avaliação analítica que tem como objetivo aferir o tempo de desempenho de uma tarefa quando executada por determinado usuário. Este método é utilizado para estimar o tempo em tarefas de IHC. O modelo estima o tempo da tarefa decompondo ela em várias tarefas menores e

atribuindo valores para cada uma destas. O modelo KLM (Keystroke Level Model), uma variação do modelo GOMS, é uma forma de comparar diferentes interfaces que executam a mesma tarefa, levando em conta um usuário que não erre o processo [Kieras 1993]. Para este trabalho, será utilizado o modelo KLM para a comparação.

2.7.2 Métodos de reconhecimentos e a reconstrução de imagens

A reconstrução de formas 3D se baseia nos dados das imagens adquiridas em Visão Computacional. Ao buscar por informações 3D de uma superfície deve-se escolher entre os modos ativo ou passivo. O método passivo consiste em pelo menos um par de câmeras, geralmente baratas, de pontos diferentes e luz ambiente (Figura 1) [Barnard e Fischler 1982]. Já o modo ativo é caracterizado por projetar um padrão de luz no objeto e através de duas câmeras (ou mais), buscar os pontos projetados [Rocker e Kiessling 1975].

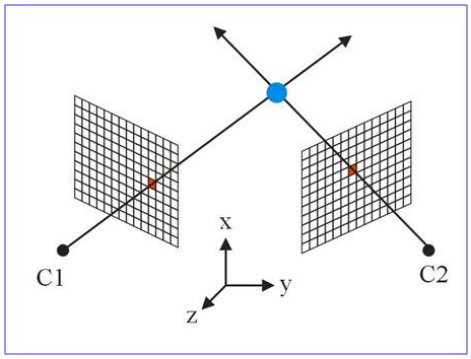


Figura 1. Representação de duas câmeras na visão estéreo [Costa 2007].

Dentre as técnicas de método passivo, a mais amplamente divulgada é a visão estéreo. Na visão estéreo a reconstrução 3D de objetos é feita a partir de um par de imagens obtidas simultaneamente com uma variação no deslocamento. Esta variação na posição gera diferenças entre imagens, conhecido como efeito paralaxe, gerando a percepção 3D e assim a percepção de profundidade, se combinadas. Para a reconstrução tridimensional pode ser utilizada a luminosidade das imagens ou informações de bordas e vértices e unir no par estéreo [Otuyama 1998].

Para representar um objeto na cena, as imagens estereoscópicas precisam se relacionar entre si. O processo para a determinação do relacionamento é conhecido como o problema de correspondência. O problema de correspondência aparece uma vez que um objeto é representado por mais de um contorno ou borda [Meyes 1994]. Ele pode ser consideravelmente minimizado por um método ativo. Um dos métodos ativos mais amplamente usados é baseado na projeção de luz estruturada (Figura 2) [Pires 2007].

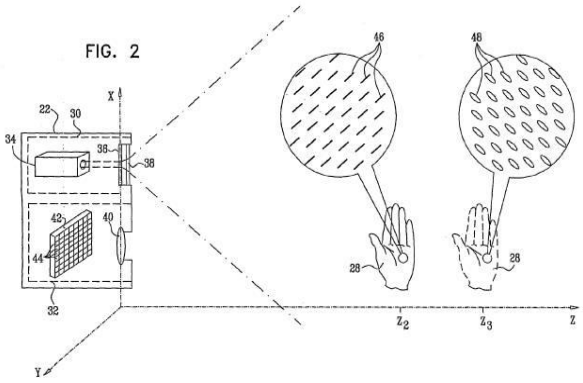


Figura 2. Projeção de luz estruturada [MSAcademic 2011]

Para estimar a profundidade o Kinect usa o espaçamento da luz projetada. A forma 3D se dá devido ao espaçamento de cada ponto, quanto maior, mais longe.

2.7.3 Kinect

O Kinect é um sensor de Reconhecimento de Gestos lançado pela Microsoft no ano de 2010. A sua arquitetura possui uma câmera colorida, uma câmera para detecção de profundidade e um projetor de infravermelho. O Kinect trabalha com o padrão de método ativo de luz estruturada para obter o mapa de profundidade da cena. A luz projetada é infravermelha (Figura 3), ficando invisível aos olhos humanos [Wu e Bainbridge-Smith 2011].

Figura 3. Imagem em infravermelho (IR) do Kinect usada para calcular a profundidade [MSAcademic 2011].

O Kinect é um sensor de Visão Computacional e Reconhecimento de Gestos quem tem capacidade de capturar dados de profundidade e gerar uma cena com detalhamento, em tempo real, e com um custo significativamente menor em comparação com alternativas de *hardware* [Tang 2011].

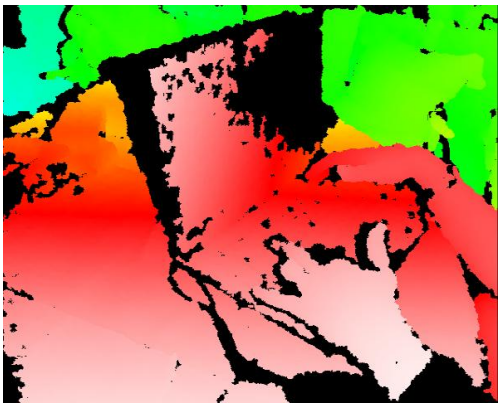


Figura 4. Em escala gradiente, sendo branco perto e azul longe, o mapa de profundidade, calculado através do IR [MSAcademic 2011].

As principais vantagens da utilização do Kinect são a obtenção das informações de forma detalhada e com bastante precisão e a rápida obtenção da informação 3D da cena. Dentre os pontos fracos, está a sensibilidade do Kinect a fontes externas de infravermelho, além de não ser aconselhável para ambientes externos, pois seu mapa de profundidade não alcança mais do que 7 metros [Wu e Bainbridge-Smith 2011].

2.7.4 PrimeSense OpenNI e Processing: Utilizando o Kinect

Processing é uma linguagem de programação e IDE (*Integrated Development Environment*) de código aberto para as pessoas que desejam criar animações. A linguagem foi desenvolvida em Java e inicialmente projetada para ensinar fundamentos de computação gráfica para estudantes. Hoje existem diversos trabalhos divulgados desenvolvidos nesta ferramenta [Processing 2012]. É fácil e simples a integração do código com o Kinect através do Simple-OpenNI.

O SimpleOpenNI é uma biblioteca de código aberto. Esta biblioteca é um *wrapper* para o OpenNI e Primesense NITE, que fornecem um conjunto de APIs (*Application programming interface*) para ser implementados no código e que obtém informações de sensores como o Kinect [Simple-OpenNI 2012].

3. Trabalhos relacionados

Com o avanço dos estudos na área de Visão Computacional chega o advento de sensores e câmeras de Reconhecimento de Gestos relativamente baratos. Um dos dispositivos mais populares usados para fazer pesquisa com Reconhecimento de Gestos é o Kinect, da Microsoft [Tang 2011].

Um exemplo é o trabalho de Silveira (2011) que defende a utilização do Reconhecimento de Gestos para a navegação em documentos utilizando o Kinect. A abordagem do autor apresenta testes com usuários através de uma aplicação capaz de reconhecer o gesto e movimentar *slides*, substituindo os métodos atuais de entrada como, por exemplo, o teclado. Silveira apresenta alguns pontos importantes nos testes que podem ser considerados no presente trabalho. Para o autor, foi inviável o uso da técnica quando houve mais de uma pessoa em frente ao sensor devido à falta de desempenho. Outro ponto citado por Silveira é a perda da capacidade do Reconhecimento de Gestos do Kinect quando o ambiente estava muito escuro. Segundo Silveira, 100% dos usuários que foram submetidos à avaliação acreditam que o sistema poderia ser utilizado substituindo os métodos convencionais.

Já Tang (2011) define um novo método para o reconhecimento da mão. O autor explora a viabilidade de Reconhecimento de Gestos em escalas menores em relação ao

padrão de rastreamento de corpo inteiro. Tang apresenta em seu trabalho uma técnica baseada em algoritmos de inteligência artificial que envolve o reconhecimento de movimentos das mãos juntamente com a análise do rastreamento do corpo do usuário a fim de aprimorar a identificação do gesto. Em seu projeto o autor cita a precisão de aproximadamente 100% para o reconhecimento de gestos como "pegar" (mão fechada) e uma média de 90% de precisão para "soltar" (mão aberta). Freeman e Weissman (1995) sugerem em seu trabalho um controle de uma televisão baseado em um único gesto. Os autores citam que o principal problema do controle de máquinas por reconhecimento de gestos da mão se dá devido à existência de diversos comandos complicados na televisão. Para atender esses comandos seria necessário o reconhecimento de gestos complexos, o que não é bom para o usuário. Outro problema citado pelos autores é o difícil reconhecimento de uma mão em um ambiente complexo. Com base nestes trabalhos, e utilizando uma abordagem semelhante, foi testado um método de interação com usuários utilizando o sensor Kinect.

4. Solução proposta

A solução proposta neste projeto foi organizada em quatro etapas. A primeira etapa foi a criação do cenário da experiência. Na segunda fase, foi desenvolvida uma aplicação para a simulação de um controle remoto com funções básicas, reconhecidas através do sensor Kinect. Nas duas últimas fases foram efetuados testes com um grupo de usuários e a uma avaliação de sua experiência, obtendo os prós e contras.

4.1. Criação do ambiente para simulação

Para a simulação do ambiente proposto foi utilizado a versão mais recente do *Software Development Kit* (SDK) do Kinect para a linguagem *Processing* [1]. O SDK utilizado foi o *PrimeSense OpenNI* [2]. O *software* foi executado em um computador com sistema operacional *Microsoft Windows 7 64 bits*, porém os *drivers* e aplicações de *32 bits*, devido a problemas conhecidos.

Para a captura da imagem foi utilizado uma placa de captura de vídeo USB – *Universal Serial Bus*. A placa foi conectada ao cabo RCA (conhecido também com *A/V* ou *Audio-Video* - nome derivado de *Radio Corporation of America*) e este por sua vez, foi ligado ao receptor de um aparelho de televisão digital. O computador que executou o *software* - produzido neste trabalho - foi conectado a uma televisão através de um cabo HDMI (*High-Definition Multimidia Interface*) retornando a imagem processada pela aplicação. Desta maneira foi possível interferir quadro a quadro e redesenhar a interface com a imagem original da televisão ao fundo.

Para a captação dos gestos, foi utilizado um sensor Kinect, posicionado em frente ao usuário e conectado através de um cabo USB diretamente no computador, responsável por enviar informações sobre os gestos de cada usuário. O SDK OpenNI, juntamente com o NITE, fornece um detalhamento de esqueletos através de APIs para detecção de gestos. Utilizando um dispositivo USB-UIRT - *Universal Infrared Receiver/Transmitter USB* a aplicação envia um código infravermelho pré-mapeado de acordo com a fabricante para simular a ação capturada pelo usuário.

---

[1] Processing é uma linguagem de programação de código aberto e ambiente de desenvolvimento integrado (IDE), construído para as artes eletrônicas e comunidades de design visual.

[2] O OpenNI Framework traz as *Application Program Interfaces* (APIs) necessárias para uso conjunto com dispositivos de Interação Natural (em inglês, Natural Interaction – NI).

4.2. Desenvolvimento da aplicação

Para construção da aplicação, realizou-se uma pesquisa nos trabalhos relacionados, referências em artigos e fontes de informação na internet. Embora ainda exista certa dificuldade de encontrar exemplos de código em livros, existe muito trabalho em volta deste dispositivo e pode ser encontrado na internet.

Para o desenvolvimento do trabalho, foi utilizado o Microsoft Kinect para *Xbox* 360 que possui uma grande limitação em relação ao último modelo lançado pela Microsoft, Kinect 1.5 para *Windows*. A nova versão do dispositivo possui rastreamento do esqueleto em pé ou sentado, reconhecimento facial e uma câmera com resolução superior para reconhecimento dos detalhes da cena.

4.2.1 Rastreamento do esqueleto

O rastreamento do esqueleto permite ao Kinect reconhecer pessoas de uma determinada cena. Com base nos dados capturados pela câmera de infravermelho, o Kinect pode reconhecer o corpo do usuário dentro do campo de visão do sensor. O mapa de cada ponto do esqueleto, mãos e reconhecimento de gestos básicos (*Push*, *Swipe*, *Wave*, *Circle*) é feito pelo *middleware* NITE (*Natural Interaction Engine*) da PrimeSense (Figura 5).

A biblioteca NITE trabalha sobre o *framework* da OpenNI (*Open Natural Interface*). A OpenNI é uma instituição sem fins lucrativos que trabalha no desenvolvimento entre dispositivos de interface natural e bibliotecas de interface natural de usuário (NUI) e interface orgânica de usuário (OUI).

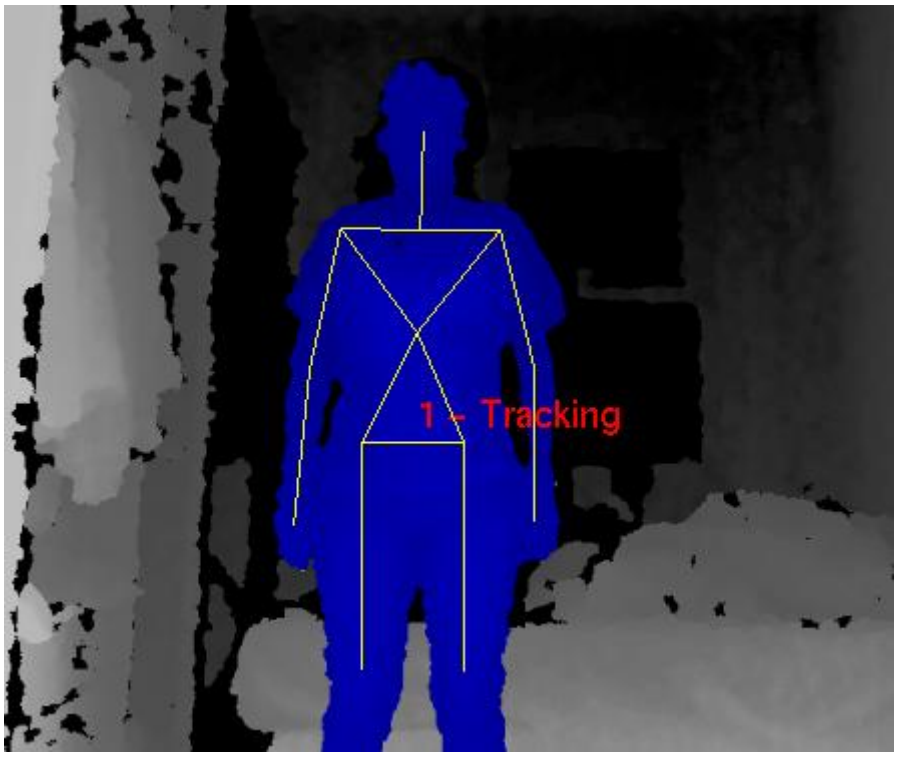


Figura 5. Rastreamento do usuário na cena.

O rastreamento é baseado em pontos do esqueleto (Figura 6). Cada ponto é formado pelas coordenadas X, Y, Z, representados no código pela classe *PVector* na linguagem *Processing* (ANEXO 1). Cada instância de *PVector* recebe o respectivo

valor através do *SimpleOpenNI*. Cada vez que a cena é redesenha, os vetores recebem a nova posição do usuário (ANEXO 2). O OpenNI precisa calibrar um usuário antes de fornecer as coordenadas do esqueleto. Ao detectar o usuário, é atribuído um ID para este que fica disponível até o usuário deixar a cena.

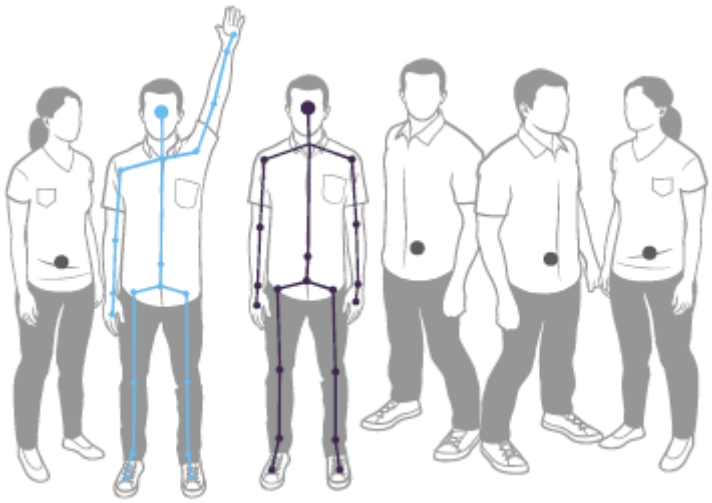


Figura 6. Pontos do esqueleto [Microsoft 2012].

4.2.2 Inicializando o teclado virtual

O NITE utiliza o fluxo de estado de sessão para determinar se o usuário esta controlando a aplicação. O gerenciamento é feito pela classe *SessionManager* e os possíveis valores são *"Not in Session"*, *"In Session"* e *"Quick Refocus"*. Durante o estado *"Not in Session",* o NITE irá tentar identifica uma pose de foco para iniciar a sessão. O estado "*In Session*" mantém o usuário interagindo com a aplicação. Por ultimo, o *"Quick Refocus"* é responsável por manter a sessão do usuário ativa por alguns segundos caso ele sai de cena, intencionalmente ou não [PrimeSense, 2010].

Nesta aplicação, o teclado virtual é exibido durante o estado *"In Session"*. Este teclado é definido pela classe *Trackpad*. As definições de controles e a detecção dos movimentos sobre o teclado são obtidos através da classe *XnVSelectableSlider2D* do NITE (Figura 7).

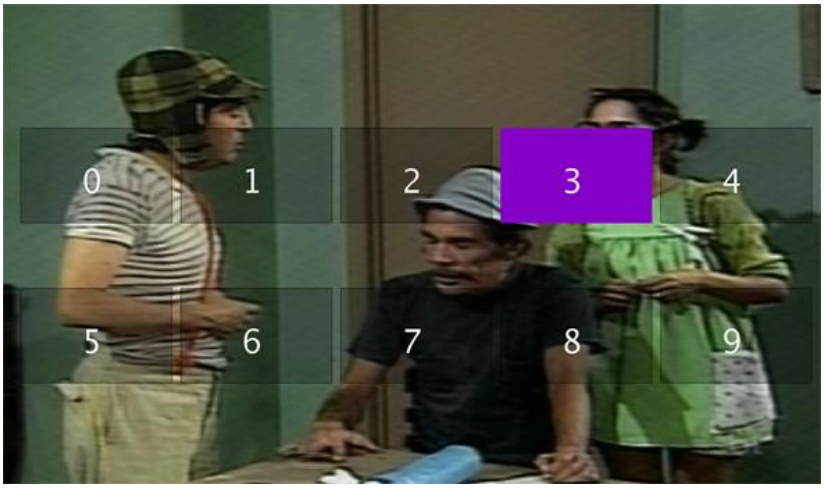


Figura 7. Teclado virtual

Para mudar o estado de *"Not In Session"* para *"In Session"*, é necessário fazer o gesto de foco *Wave* (abanar) nesta aplicação.

4.2.3. Reconhecimento de gestos

O NITE oferece um conjunto limitado de gestos que serão reconhecidos através de uma sequência de pontos. Esta tarefa é complexa e existem várias maneiras de reconhecer um gesto através de padrões. Neste trabalho foi utilizado o algoritmo conhecido como *Dynamic Time Warping* (DTW).

O DTW baseia-se em programação dinâmica, uma técnica computacional para o aprimoramento de resolução de problemas algorítmicos. O algoritmo faz comparação de padrões baseado em sua forma e não leva em consideração o tamanho e tempo das amostras [Keerthy 2012]. Dado um conjunto de padrões de referência, conhecidos como *templates*, o DTW é capaz de encontrar qual *template* se ajusta com um determinado padrão de forma eficiente. Para o cálculo, é feita uma matriz com os dados do *template* na horizontal e os pontos do padrão procurado na vertical.

$$\text{Template:} \quad \underline{r}(1), \underline{r}(2), \ldots, \underline{r}(I)$$

$$\text{Test pattern:} \quad \underline{t}(1), \underline{t}(2), \ldots, \underline{t}(J)$$

O cálculo da similaridade pelo algoritmo DTW é definindo através da seguinte equação:

$$DTW(i,j) = \gamma(y_i, z_j) + min \begin{cases} DTW(i, j-1) \\ DTW(i-1, j) \\ DTW(i-1, j-1) \end{cases}$$

sendo $\gamma\ (y_i\ ,\ z_j)$ uma função que calcula a distância entre os pontos (normalmente, distância Euclidiana). O principal benefício do uso do DTW é a simplicidade em sua implementação, pois não necessita de modelos matemáticos complexos para executar a tarefa [Souza *et al.* 2009]. Para a detecção do gesto foi levado em consideração a variação do movimento em uma determinada direção. Para cada direção foi atribuído um valor inteiro sendo direita: 1, esquerda: 2, cima: 3 e baixo 4. Para as diagonais soma-se o valor das direções, por exemplo, diagonal superior esquerda, soma-se o valor de cima e esquerda, totalizando 5. Com estes valores é criado um vetor com os últimos movimentos dos usuários e este é comparado aos modelos da aplicação (ANEXO 3).

4.2.3. Trocar de canal

Atualmente, é predominante em controles remotos o envio de sinal via protocolo infravermelho (IR). A luz infravermelha é responsável por enviar sinais entre controle remoto e o dispositivo, como uma televisão por exemplo. O controle remoto é capaz de mandar pulsos de luz que correspondem a códigos binários. Além do código específico do botão que será utilizado (como por exemplo, abaixar o volume) é enviado também mais informações como uma sequência binária que representa iniciar comando, o código de abaixar volume, o endereço/código do dispositivo e um comando de parar (quando o botão é solto). Esta luz infravermelha fica fora da faixa de frequência em que o olho humano consegue enxergar [Wikipédia 2012].

Neste trabalho, foram utilizados alguns códigos infravermelhos obtidos do controle remoto original. A classe *IRData* possui os códigos que são enviados ao dispositivo USB para cada interação com o teclado virtual (ANEXO 4).

4.3. Experimentos - Avaliação preditiva

Utilizando o modelo GOMS-KLM, foi executada uma comparação com controle remoto comum. A tarefa consiste em mudar o canal atual para o canal 12 (Tabela 1). Embora a experiência seja nova, os valores de K foram definidos como sendo um usuário familiarizado com a aplicação. Percebe-se que a execução do modelo de interação baseado em gestos é mais lenta se comparado ao clássico controle remoto.

Tabela 1. Avaliação da tarefa (modelo KLM)

Controle Remoto		**Gestos**	
Prepara-se mentalmente	M *(1,35s)*	Prepara-se mentalmente	M *(1,35s)*
Posicionar o controle	H *(1,10s)*	Levantar a mão	H *(1,10s)*
Prepara-se mentalmente	M *(1,35s)*	Prepara-se mentalmente	M *(1,35s)*
Digitar "1"	K *(0,28s)*	Abanar	D *(6x0,5)*
Digitar "2"	K *(0,28s)*	Digitar "1"	K *(0,28s)*
		Digitar "2"	K *(0,28s)*
Tempo estimado	**3,01s**	**Tempo estimado**	**6,20s**
** Usuário familiarizado com o teclado.*		** Usuário familiarizado com o teclado.*	

4.4. Experimentos - Aplicação de testes com usuário e avaliação

Nesta etapa foi reunido um grupo de dezoito pessoas de variadas características. Para todas as pessoas foram dadas instruções básicas de comportamento da aplicação e solicitado que respondessem um questionário ao final da experiência. A aprendizagem para a utilização da aplicação é demorada. Percebe-se nitidamente uma grande dificuldade dos usuários com idade mais avançada, até mesmo com movimentos simples, como abanar por exemplo. Em seus feedbacks, a maioria dos usuários citou que a experiência foi diferente e interessante, porém 41% não comprariam uma TV hoje que possuísse reconhecimento de gestos (sem considerar outras variáveis, como preço, etc).

4.5. Experimentos - Teste de precisão com controles de volume

Nesta etapa, para os mesmos usuários, foi solicitado que fizessem algumas tarefas, pausadamente, três vezes. O objetivo foi testar a precisão do reconhecimento dos gestos reconhecidos pelo *framework* NITE e pelo DTW.

Para começar, tarefa dada aos usuários foi "mudar o canal para 12". O reconhecimento é feito pelo *framework* NITE, através dos gestos *wave* e *push*. Dentre as tarefas aplicadas, esta obteve a maior taxa de erros pois apenas 29,41% das tentativas houve êxito. Neste teste foi fácil identificar onde os erros ocorrem: praticamente todos os usuários levavam seu braço levemente para esquerda ou direita durante o gesto de *push*. Isso leva a mudança do foco para o botão mais perto antes de concluir o gesto, fazendo com que o usuário cometa um erro.

A segunda tarefa em questão foi "mudar para o canal 5" onde 80,39% das tentativas obtiveram êxito. O problema visto no primeiro teste foi observado neste caso também, porém, devido a localização do botão 5 (canto), os erros diminuíram.

A terceira tarefa foi "aumentar o volume". Nesta parte, os gestos são reconhecidos baseados em *templates* utilizando DTW. O modelo requer que o usuário leve o braço levemente para cima e após a direita. Durante os testes, 74,51% das tentativas apresentaram sucesso. Por último, a tarefa "diminuir o volume" obteve

64,71% de êxito. A tarefa requer que o usuário leve o braço levemente para cima e após para esquerda.

5. Conclusão

Este trabalho avaliou a interação dos usuários com um televisor através do reconhecimento de gestos. A técnica utilizada foi avaliada por pessoas em uma sessão de testes.

Observou-se uma grande dificuldade por parte dos usuários em substituir o controle remoto normal por este modelo. Um dos pontos mais críticos foi em relação ao "empurrar" (*push*) do botão do teclado virtual fornecido pelo NITE, apontando para necessidade de melhoria na precisão do reconhecimento deste gesto, a fim de proporcionar uma melhor experiência para os usuários. Uma possível alternativa seria que, ao iniciar o gesto *push*, o *framework* inclua uma margem de erro da variação do eixo X da mão do usuário.

A interação por gestos neste trabalho permite observar que aumentando a exatidão do reconhecimento de gestos, os usuários ficam insatisfeitos, pois requer que realizem os gestos com muita precisão. Porém, diminuindo a precisão, é muito fácil de observar falsos positivos, como ocorreu algumas vezes durante a troca de canal em que o sistema entendeu como “aumentar volume”.

Como objetivo secundário, foi avaliada a precisão do reconhecimento de gestos utilizando o algoritmo conhecido *Dynamic Time Warping*, onde duas sequências de dados foram comparadas, independentemente do tempo, calculando a sua semelhança. O reconhecimento de gestos com essa técnica superou as expectativas, pois se trata de um modelo matemático simples se comparado a técnicas utilizando *machine learning* ou redes neurais.

Obtendo informações do questionário (ANEXO 5) de *feedback* preenchido pelos usuários após o teste, 94,12% dos usuários ficaram cansados durante ou depois do experimento. Neste mesmo questionário foi exibida uma foto com seis botões de um controle remoto comum e logo perguntado qual a funcionalidade de cada botão. A maioria dos usuários sabia a função de apenas um destes botões. Corroborando com Nielsen, 76,47% dos usuários pensam que os nomes dos botões são insuficientes para explicar sua funcionalidade (ANEXO 6).

Apesar de ser uma experiência interessante como descrito pela maioria dos participantes da pesquisa, a utilização deste modelo deve ser opcional, permitindo outros modos de interação para atender variações do ambiente, limitações físicas e motoras dos usuários e outros possíveis cenários não mapeados neste trabalho. O código desenvolvido neste trabalho encontra-se disponível para *download* em https://github.com/jonathansp/KinectRemoteControl.

6. Referências

Ascencio, A. F. G. (1999) “Método Heurístico para Projetar e Analisar Interfaces Hipermídia Inteligentes, http://bit.ly/SJX6M5, abril de 2012.

Barreto, D. de M. (2011) “TV Digital Interativa: uma nova forma de assistir à TV”, ECCOM, v. 2, n. 3, p. 16-23, jan/jun., http://www.fatea.br/seer/index.php/eccom/article/viewFile/422/275, abril de 2012.

Barnard, S. T. e Fischler, M. A. (1982) “Computational stereo”, Computing Surveys.

Bonsiepe, G. (1997) “Design: do material ao digital”, Florianópolis, FIESC/IEL.

Costa, R. (2007) “Reconstrução 3D a partir de fotos estéreo”, http://www.tecgraf.puc-rio.br/~mgattass/ra/trb07/RicardoCosta/trab3.html, maio 2012.

Crowley, J. L. e Christensen, H. I. (1993) “Vision as Process”, Springer Verlag Basic Research Series.

Donald A. Norman , Jakob Nielsen, Gestural interfaces: a step backward in usability, interactions, outubro de 2010 .

Gismero, G. (2012) “Visión por Computador: Estado del arte”, Inco - Fing, Udelar, Montevideo, Uruguay.

Gope, D. C. (2011) “Hand Gesture Interaction with Human-Computer”, Dhaka University of Engineering and Technology, Gazipur.

Hix, D. e Hartson, H.R (1993) “Developing User Interfaces”, Wiley, United States of America.

Keerthy, N. Kumar (2012), “Virtual Kung fu Sifu with Kinect” Master's Projects. [Online] Disponível em: http://scholarworks.sjsu.edu/etd_projects/252. 2012.

Kieras, D. (1993), “Using the Keystroke-Level Model to Estimate Execution Times”. University of Michigan, [Online] Disponível em: http://www.pitt.edu/~cmlewis/KSM.pdf.

Manfredo, M. T. (2011) “A complexa busca pela simplicidade”, http://www.comciencia.br/comciencia/?section=8&edicao=70&id=875&tipo=0, maio de 2012.

Martins Jr., A. C. e Pimentel, M. da G. C. (2011) “Interação usuário - TV digital interativa: Contribuições Via Controle Remoto”, Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

Meyes, D. (1994) “Reconstruction of Surfaces From Planar Contours”, Doctor of Philosophy University of Washington.

Microsoft (2012), “Kinect for Windows Human Interface Guidelines”, Disponível em: http://msdn.microsoft.com/en-us/library/jj663791.aspx

Montez, C. e Becker, V. (2004) “TV Digital Interativa: Conceitos e Tecnologias”, In: WebMidia e LA-Web 2004 – JointConference, Ribeirão Preto, SP, Outubro.

MSAcademic (2011), http://msacademic.rs/Blog.aspx?id=167, julho 2012.

Nielsen, J. (2004) “Remote Control Anarchy”, http://www.useit.com/alertbox/20040607.html, abril de 2012.

Otuyama, J. (1998) “Visão estéreo”, Universidade Federal de Santa Catarina, http://www.inf.ufsc.br/~visao/1998/otuyama/1.html, junho de 2012.

Pinho, R. R., Tavares, J. M. R. S. e Correia, M. F. P. (2005) “Análise de Movimento Humano por Visão Computacional: Uma Síntese”, Porto, Portugal, Instituto de Engenharia Biomédica.

Pires, D. da S. (2007) “Rastreamento de componente conexo em vídeo 3D para obtenção estruturada tridimensional”, Universidade de São Paulo, Instituto de Matemática e Estatística.

PrimeSense (2010), “Prime Sensor NITE 1.3 Controls Programmer's Guide", PrimeSense Inc. [Online]. Disponível em: http://pr.cs.cornell.edu/humanactivities/data/NITE.pdf

Processing (2012), http://www.processing.org, agosto de 2012.

Rehm, M., Bee, N. e André, E. (2007) “Wave Like an Egyptian - Accelerometer Based Gesture Recognition for Culture Specific Interactions”, British Computer Society.

Rocker, F.; Kiessling, A. (1975) “Methods for analyzing three dimensional scenes”, Proceedings of International Joint Conference on Artificial Intelligence.

Rosefeld, A. (1999) “Image Analysis and Computer Vision”, Center For Automation Research, Computer Vision Laboratory, University Of Maryland, College Park, Md 20742-3275.

Schwartz, B. (2007) “O paradoxo da escolha: por que mais é menos”, São Paulo, A Girafa Editora.

Silveira, M. A. da (2011) “Técnica de Navegação em Documentos Utilizando o Microsoft Kinect”, Departamento de Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.

Simple-Openni (2012), http://code.google.com/p/simple-openni/, abril de 2012.

Stone, D. (2005) “User Interface Design And Evaluation”, Morgan Kaufmann Series in Interactive Technologies.

Souto Maior, M. (2002) “A TV interativa e seus caminhos”, Dissertação Final de Mestrado Profissional (Mestrado em Computação) – Instituto de Computação, Universidade Estadual de Campinas, Campinas.

Souza, C. F. S. de; Pantoja, C. E. P.; Souza, F. C. M. de. "Verificação de assinaturas off-line utilizando o algoritmo dynamic time warping". Universidade Federal de Pernambuco – UFPE. 2009.

Tang, M. (2011) “Recognizing Hand Gestures with Microsoft’s Kinect”, Department Of Electrical Engineering, Stanford University, Palo Alto, USA.

Tapscott, D. e Caston, A. (1995) “A. Mudança de paradigma: a nova promessa da tecnologia da informação”, Trad. Pedro Catunda, São Paulo, Makron Books.

Togores, T. A. (2011) “Vitruvius: Um Reconhecedor de Gestos para o Kinect”, Departamento de Instituto de Matemática e Estatística, USP, São Paulo.

W. T. Freeman and C. D. Weissman (1994), "Television control by hand gestures", Mitsubishi Electric Research Laboratories, Tech. Rep. TR94-24. [Online]. Disponível em: http://www.merl.com/papers/TR94-24/

Walker, J. (1990) “Through the looking glass”, The art of human-computer interface design, Massachusetts, Addison-Wesley Publishing.

Wikipedia (2012), “Likert Scale”, Wikipedia, The Free Encyclopedia, 20 Aug 2012 at 20:45 UTC, http://en.wikipedia.org/wiki/Likert_scale, agosto de 2012.

Wikipedia (2012), “Remote Control”, Wikipedia, The Free Encyclopedia, 16 Nov 2012 at 15:24 UTC, http://en.wikipedia.org/wiki/Remote_control, novembro de 2012.

Wolf, C. G e Morrel-Samuels P. (1987) “The use of hand-drawn gestures for text editing”, In: International Journal of Man-Machine Studies, volume 27, pages 91-102.

Wu, H. e Bainbrisge-Smith, A. (2011) “Advantages of using a Kinect Camera in various applications”, Electrical e computer Engineering, University of Canterbury, New Zealand.

7. Anexos

ANEXO 1 – Representação dos pontos do esqueleto na linguagem *Processing*.

```
PVector jointNeck3D = new PVector();
PVector jointLeftShoulder3D = new PVector();
PVector jointLeftElbow3D = new PVector();
PVector jointLeftHand3D = new PVector();

(...)
```

ANEXO 2 – Atribuição dos pontos do esqueleto obtidos pelo NITE.

```
context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL_NECK,jointNeck3D);
context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL_LEFT_SHOULDER,jointLeftShoulder3D);
context.getJointPositionSkeleton(userId,SimpleOpenNI.SKEL_LEFT_ELBOW,jointLeftElbow3D);

(...)
```

ANEXO 3 – Coleção dos pontos dos modelos para comparação utilizando o DTW.

```
public Recognizer()
{
  maxThreshold = 0.01F;
  gesturesBufferSize = 15;
  gestures = new HashMap<String, Float[]>();
  moves = new SimpleQueue<Float>();
  gestures.put("Quadrado", new Float[] {
    1.0F, 1.0F, 1.0F, 1.0F, 3.0F, 3.0F, 3.0F, 3.0F, 2.0F, 2.0F, 2.0F, 2.0F, 4.0F, 4.0F, 4.0F
  }
  );
  gestures.put("+Vol", new Float[] {
    4.0F, 4.0F, 4.0F, 4.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F, 1.0F
  }
  );
  gestures.put("-Vol", new Float[] {
    4.0F, 4.0F, 4.0F, 4.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F, 2.0F
  }
  );
}
```

ANEXO 4 – *Hashtable* com a coleção dos códigos que serão enviados via infravermelho.

```
private void setupDevices()
{
     Hashtable<String, String> _control = new  Hashtable<String, String>();
    _control.put("0","0000 006C 0022 0002 0157 00AC 0016 0040 0016 0015 0016 0015 0016
    _control.put("1","0000 006C 0022 0002 0157 00AC 0016 0040 0016 0015 0016 0015 0016
    _control.put("2","0000 006C 0022 0002 0156 00AC 0016 0040 0016 0015 0016 0015 0016
    _control.put("3","0000 006D 0022 0002 0155 00AB 0016 0040 0016 0015 0016 0015 0016

    (...)
```

ANEXO 5 – Questionário de usabilidade

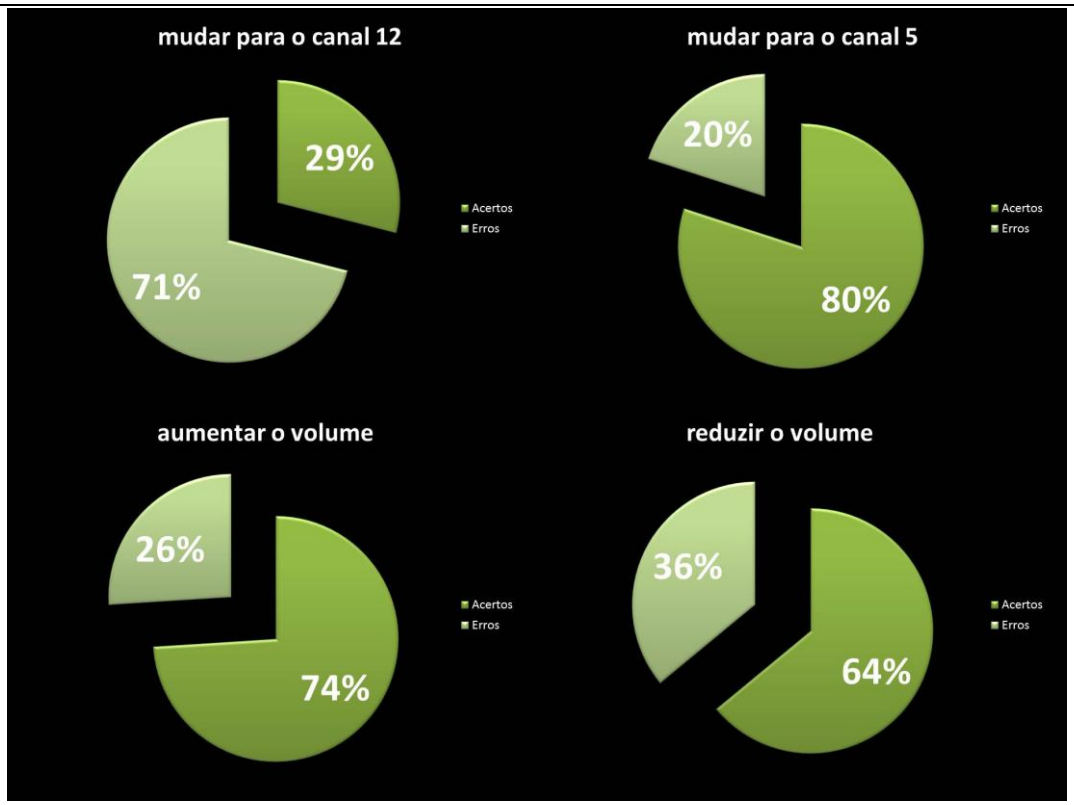
Questionário de usabilidade

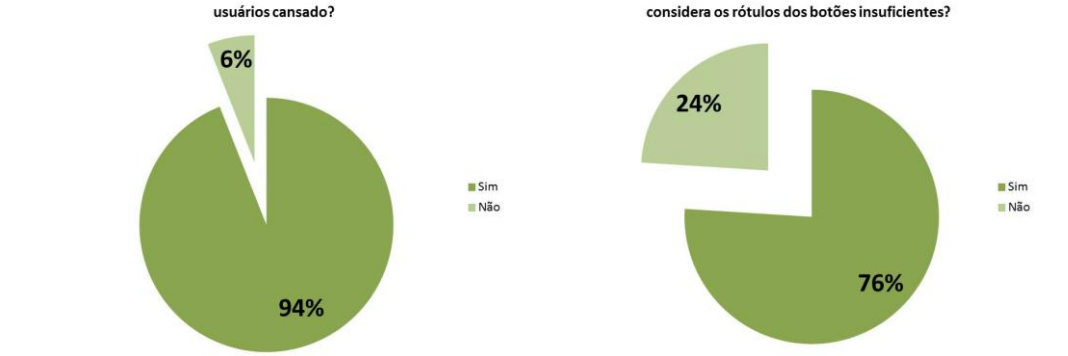
Trabalho de conclusão – Sistemas de informação 2012/2

Obrigado por participar do teste. Para responder, analise cada uma das afirmações que melhor representa a sua opinião sobre os testes realizados. É fundamental que ao responder cada questão você retrate sua realidade; não pense no que seria a resposta ideal. Se tiver dúvida, solicite fazer o teste novamente.

	Discordo totalmente	Discordo	Neutro	Concordo	Concordo totalmente
1. Na minha casa possuo muitos aparelhos com controle remoto.	1	2	3	4	5
2. Confunde-me possuir muitos controles remotos (DVD, TV, SOM, TV A CABO).	1	2	3	4	5
3. Prefiro ter um controle universal a usar cada um dos controles separadamente.	1	2	3	4	5
4. Interação de gestos com eletrodomésticos pode ser complexa de uma forma desnecessária.	1	2	3	4	5
5. Penso que as pessoas aprenderiam a utilizar de forma rápida o reconhecimento de gestos.	1	2	3	4	5
6. Se hoje minha televisão suportasse interação por gestos, deixaria de utilizar o controle remoto aos poucos.	1	2	3	4	5
7. No futuro, o controle remoto de televisores será substituído por métodos de interação por gestos.	1	2	3	4	5
8. Hoje, não me interessa comprar um produto que utilize interações por gestos.	1	2	3	4	5
9. Fiquei cansado ao utilizar o sistema.	1	2	3	4	5
10. De forma geral, achei interessante a experiência de interagir com um eletrodoméstico por gestos.	1	2	3	4	5

ANEXO 6 – Gráficos dos resultados





8. Agradecimentos

Agradeço a Deus por me guiar nesta trajetória. Aos meus pais por me dar o apoio sempre quando preciso.

Ao meu orientador, professor Átila Bohlke Vasconcelos, por me ter aceitado como seu orientando.

Ao professor Dr. Wilson Pires Gavião Neto, por me ajudar em assuntos tão difíceis com tanta clareza.

A minha namorada, professora Simone Perotto, por me ajudar e me dar forças quando precisei e dedicar seu tempo me ajudando com correções.

Ao meu ex-colega e amigo Maxwell Dayvson, por mostrar que sempre é possível dar um passo a frente e por me ensinar muitas coisas.

Ao meu colega e amigo Rodolfo Machado, pela confiança e colaboração indispensável para o meu trabalho.

Os meus sinceros agradecimentos a todos vocês!